



HAL
open science

Space-Time accurate anisotropic adaptation and stabilized finite element methods for the resolution of unsteady CFD problems

Ghina El Jannoun

► **To cite this version:**

Ghina El Jannoun. Space-Time accurate anisotropic adaptation and stabilized finite element methods for the resolution of unsteady CFD problems. Fluid mechanics [physics.class-ph]. Ecole Nationale Supérieure des Mines de Paris, 2014. English. NNT : 2014ENMP0077 . tel-01146245

HAL Id: tel-01146245

<https://pastel.hal.science/tel-01146245>

Submitted on 28 Apr 2015

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

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

École doctorale 364 : Sciences Fondamentales et Appliquées

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'école Nationale Supérieure des Mines de Paris

Spécialité doctorale "Mécanique Numérique"

présentée et soutenue publiquement par

Ghina JANNOUN

le 22 Septembre 2014

Space-Time accurate anisotropic adaptation and stabilized finite element methods for the resolution of unsteady CFD problems

~~~~~

**Adaptation anisotrope précise en espace-temps et méthodes d'éléments finis stabilisées pour la résolution de problèmes de mécanique des fluides instationnaires**

Directeurs de thèse : **Elie Hachem, Thierry Coupez**

#### Jury

|                             |                                                   |            |
|-----------------------------|---------------------------------------------------|------------|
| <b>M. David Darmofal,</b>   | Professeur, Massachusetts Institute of Technology | Rapporteur |
| <b>M. Tarek Zohdi,</b>      | Professeur, University of California at Berkeley  | Rapporteur |
| <b>M. Johan Hoffman,</b>    | Professeur, KTH Royal Institute of Technology     | Examineur  |
| <b>Ms. Simona Perotto,</b>  | Professeur Associé, Politecnico di Milano         | Examineur  |
| <b>M. Nabil Nassif,</b>     | Professeur, American University of Beirut         | Examineur  |
| <b>M. Elie Hachem,</b>      | Professeur Associé, Mines ParisTech               | Examineur  |
| <b>M. Thierry Coupez,</b>   | Professeur, École Centrale de Nantes              | Examineur  |
| <b>M. Christian Dumont,</b> | Docteur, Aubert & Duval                           | Invité     |

**MINES ParisTech**

**Centre de Mise Forme des Matériaux (CEMEF)**  
UMR CNRS 7635, F-06904 Sophia Antipolis, France



*It is but the beginning ...*



# ACKNOWLEDGEMENTS

ليروت من قلبي سلام لبيروت ...

When all is done, it is a joy to look back over the path and remember all the friends and family who helped me cross my way through this long and fruitful journey. First of all, I thank all the amazing people who enormously helped me and who I forgot to mention here, as you know I have a fish memory :(

I would like to express my heartfelt gratitude to my advisors, Thierry Coupez and Elie Hachem. Thierry, I gratefully acknowledge your help and support. Your keen scientific insights, bright ideas and constructive remarks inspired and enriched my growth as a scientist. I learned a great deal from you. You keep on telling me you're the devil, well if that was true then peace and prosperity will be on earth. My gratitude goes as well to Elie who believed in me and encouraged me from the very early stages of this work. I will always remember the midnight phone calls, the valuable suggestions, your enthusiasm and positive mindset. Thank you for helping me be an independent thinker, to look beyond the ordinary, and develop as a researcher. That was a true privilege.

I would also like to thank my thesis committee members: Prs. Darmofal, Zohdi, Hoffman, and Perotto, who provided thoughtful directions and constructive feedback. I hope you enjoyed reading my manuscript and that it was as "heavy" as it weighted. Special regards go to Pr. Nabil Nassif who established the backbones of my development from an early age as a student, starting from the basic undergraduate courses all the way through the most challenging graduate tasks. My regards go as well to Pr. Touma who nourished my intellectual maturity. I was luckily surrounded by wonderful lab leaders and mates: Elisabeth, Patrick, Françoise, Marie-Françoise, Geneviève, Florence, Louisa, Hugues, Marc, Nathalie and P-O, you have provided a rich and fertile environment and a warm and inviting place to work. I also acknowledge the Agence Nationale de Recherche (ANR), and the members of the REALisTIC project, I see their fingerprints all over the best parts of my work.

I have truly been blessed with friends all over this long journey. I would especially like to thank Jeremy Veysset for the great times and memories we have built together, for sharing various thoughts and giving me valuable opinions. You were always next to me not just as a colleague but as a brother. Jeremz, I hope our friendship will last forever.

Sauci, I was fortunate to have you by my side no matter the distance that separates us. Loulou, my dearest friend, thank you for the moral support, for the joy you planted in me during the few moments we spent together these years.

Stephy and Kiko, 3absoun, thank you for making my journey a funny one. I will always remember the lunch meetings, the barbecues, the shopping times, and the weekly walks around Antibes' shore. Joe, the godfather, thank you for all the advice, the ones I apply and the ones I fail to. Ali & Noura, you have been like a surrogate family for me, bearing the brunt of frustrations, and sharing the joy of success. Noura, I have a spoiler for you, 'Brody is NOT alive'. My fellow Londoners, Carine, my jogging mate, and Simon, the Marathon man and my biking tutor, it is time you come back to Antibes. Gottis, I sincerely thank you for keeping the sense of humour when I lost mine.

Bilal, Dania, Faf, Khouloud, Noura, Roro, and Suz thank you for being next to me in the ups and downs regardless of how far we were, I really appreciate it.

To Nancy I would like to express my sincere gratitude for providing me with words of comfort all through this long journey. We have struggled together during these years but we did it!!! Maher & Anais, les mariés, thank you for being next to me when I most needed it. Batta will always love you.

Laure & Bastien, it was a pleasure to hang around with you guys. A big thanks for the instruc-

tive lessons in ski. Laure, merci pour ton amitié, merci pour les fous-rire, merci de m'avoir soutenu quand je me plaignais. Fred, thank you for the delicious brownies and cookies, they were worth the kilos I gained during the last year. Achraf, no words can value how kind you were to me. Thanks! Alex and Romain, thanks for sharing with me lots of funny and pleasant moments. Jean-François, I am truly indebted to you for guiding me through my first steps in CIMLIB.

Rebecca and Fadi, thanks for the pleasant times, the coffee breaks and the roaring laughters, your presence perpetually refreshed my days. I would also like to acknowledge the king of ski, Charbel; it was fun and inspiring to be around with you man. Pampa, as I always say, 'btefhamle raseh'.

To my dear comrades, Abbas, Apples, Anne, Antoine, Benjamin, Carole, Cynthia, Danai, Dimitri, Dorian, Elias, Francesco, Gabriel, Geoffrey, Jacob, Jose, Koffi, Luis, Massiel, Mehdi, Mohamad Ibrahim, Modesar, Myriam, Pam, Paul, Paula, Quentin, Romain, Sabrina, Stephanie, Xavier, Ziad, Zhao and 'ta2 7anak nation' thanks for the fun and support. I will certainly miss hanging out with you.

Masris, thanks for the humor, the motivation and moral support.

Words fail me to express my appreciation to Moze whose support, dedication and persistent confidence in me, has taken the load off my shoulder. ThankS for the proof-readings, with big S. Thanks for believing in me, for giving me the force to go on and for bearing with my mood swings after long days of hard work. Thanks for always being there, no matter what, and for always reminding me that being hopeless is never an option. Peanut, I would have never done it without you. I wish you and Quinta da Regaleira a joyful life and successful children.

GrandMa I don't have words to thank you for your prayers; I love you so much. Auntie Rafa, I missed you so much. I am sorry I couldn't be there to say goodbye. I will always remember the old memories, the moments of joy and laughter. Peace be upon your soul. Family, tantes et oncles, cousins et cousines, zamo Samir et tante Hana, grand merci pour le support et les prières... bisés...

Parents, if I am to be anything in this world, it is thanks to you. My brother, Sa33oud, you have been a protective shelter and a driving force through the best and worst of times. If I were to choose a brother in life, I would have chosen you. I am grateful for the moral support and encouragement. I am grateful for the lovely moments we spent together. Thank you for being next to me at the very end of my thesis writing, you have been the fuel that kept me going on. Don't sell my bed, I will be back one day.

My Father, Barhoum, the person who showed me the joy of intellectual pursuit ever since I was a child; I am greatly indebted to you for everything you have done for me. I know it was too hard for you to let me leave home and be self-dependent; how wouldn't it be hard and you were always next to me looking for my own good and holding my hand through the difficulties. Dad, thank you for trusting and believing in your daughter. You have always said "Ghannouj batal" well this is absolutely thanks to you. I will always be your little girl, I love you and I will never disappoint you.

My Mother, Ghadghoud, the one who sincerely raised me with her caring and gentle love, thanks for your unwavering faith, trust and mostly patience. Thanks for being my closest friend, my guide through the hard times. Thanks for the tears you've shed on me; thanks for being my mom.

... وأعشق عمري لأنني إذا متّ، أنجّل من دمع أُمي ...

... يا ست الحبايب ...

# CONTENTS

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| LIST OF FIGURES                                                 | xii |
| INTRODUCTION                                                    | 1   |
| 1 UNSTEADY COMPUTATIONAL FLUID DYNAMICS                         | 17  |
| 1.1 INTRODUCTION                                                | 19  |
| 1.2 CONVECTION-DIFFUSION-REACTION PROBLEM                       | 19  |
| 1.2.1 Governing equation                                        | 21  |
| 1.2.2 Standard Galerkin formulation                             | 22  |
| 1.2.3 Time integration scheme                                   | 23  |
| 1.2.4 Streamline Upwind Petrov-Galerkin (SUPG) method           | 24  |
| 1.2.5 Existence, uniqueness, stability and order of convergence | 25  |
| 1.2.6 Choice of the stabilizing parameter                       | 27  |
| 1.2.7 Element characteristic length                             | 28  |
| 1.2.8 Shock Capturing Petrov-Galerkin                           | 28  |
| 1.2.9 Transient heat transfer by conduction                     | 30  |
| 1.2.10 Enriched space approach                                  | 32  |
| 1.2.11 Time interpolated enriched (EM-I) method                 | 35  |
| 1.3 THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS                  | 35  |
| 1.3.1 Governing equations                                       | 37  |
| 1.3.2 Standard Galerkin formulation                             | 38  |
| 1.3.3 Variational Multiscale (VMS) stabilization method         | 40  |
| 1.3.4 Choice of the stabilizing parameter                       | 45  |
| 1.4 TURBULENCE MODELLING                                        | 45  |
| 1.4.1 Direct Numerical Simulation (DNS)                         | 46  |
| 1.4.2 Reynolds Averaged Navier-Stokes (RANS)                    | 47  |
| 1.5 CONJUGATE HEAT TRANSFER                                     | 52  |
| 1.6 NUMERICAL VALIDATION                                        | 55  |
| 1.6.1 Turbulent flow behind a prismatic cylinder (2D)           | 55  |
| 1.6.2 Natural convection (2D)                                   | 58  |
| 1.6.3 Heat treatment of workpieces inside an industrial furnace | 62  |
| 1.7 CONCLUSION                                                  | 67  |
| 1.8 RÉSUMÉ FRANÇAIS                                             | 69  |

|          |                                                                                   |            |
|----------|-----------------------------------------------------------------------------------|------------|
| <b>2</b> | <b>ANISOTROPIC MESH ADAPTATION</b>                                                | <b>71</b>  |
| 2.1      | INTRODUCTION . . . . .                                                            | 73         |
| 2.2      | MESH GENERATION . . . . .                                                         | 73         |
| 2.2.1    | The three classical mesh generation algorithms . . . . .                          | 74         |
| 2.2.2    | The topological optimization mesh generation algorithm . . . . .                  | 75         |
| 2.3      | AMR: ADAPTIVE MESH REFINEMENT . . . . .                                           | 80         |
| 2.3.1    | Techniques of mesh adaptation . . . . .                                           | 82         |
| 2.4      | METRIC-BASED ANISOTROPIC MESH ADAPTATION . . . . .                                | 83         |
| 2.4.1    | State of the art . . . . .                                                        | 83         |
| 2.4.2    | Preliminary definitions and theoretical framework . . . . .                       | 83         |
| 2.4.3    | Metric based mesh adaptation . . . . .                                            | 85         |
| 2.4.4    | Metric construction at the nodes' level . . . . .                                 | 87         |
| 2.5      | ERROR ESTIMATION FOR ANISOTROPIC MESH ADAPTATION . . . . .                        | 95         |
| 2.5.1    | Overview on error estimation techniques . . . . .                                 | 95         |
| 2.5.2    | Global optimization problem . . . . .                                             | 96         |
| 2.5.3    | Gradient and strong continuity along the edges . . . . .                          | 97         |
| 2.5.4    | Least square gradient recovery . . . . .                                          | 102        |
| 2.5.5    | A posteriori error analysis . . . . .                                             | 103        |
| 2.5.6    | Edge-based error estimation . . . . .                                             | 106        |
| 2.5.7    | Optimal control on the $L_p$ norm of the interpolation error . . . . .            | 109        |
| 2.5.8    | Optimal metric construction . . . . .                                             | 114        |
| 2.5.9    | Privileged length distribution tensor . . . . .                                   | 122        |
| 2.6      | ANISOTROPIC MESH ADAPTATION THROUGH LOCAL TOPOLOGICAL OPTI-<br>MIZATION . . . . . | 124        |
| 2.7      | FIELDS' INTERPOLATION BETWEEN MESHES . . . . .                                    | 125        |
| 2.8      | NUMERICAL EXPERIMENTS . . . . .                                                   | 128        |
| 2.8.1    | Convergence tests on a quadratic function . . . . .                               | 128        |
| 2.8.2    | Numerical validations on functions with steep gradients . . . . .                 | 130        |
| 2.8.3    | Numerical validations on functions with multiscale variations . . . . .           | 138        |
| 2.9      | CONCLUSION . . . . .                                                              | 140        |
| 2.10     | RÉSUMÉ FRANÇAIS . . . . .                                                         | 141        |
| <b>3</b> | <b>COUPLING THE ANISOTROPIC MESH ADAPTATION WITH CFD PROBLEMS</b>                 | <b>143</b> |
| 3.1      | A PRIORI ERROR ANALYSIS . . . . .                                                 | 145        |
| 3.1.1    | convection-diffusion-reaction equation . . . . .                                  | 145        |
| 3.1.2    | Incompressible Navier-Stokes equations . . . . .                                  | 147        |
| 3.2      | TUNING WITH STABILIZED FINITE ELEMENT METHODS . . . . .                           | 148        |
| 3.2.1    | Illustrative example . . . . .                                                    | 151        |
| 3.2.2    | Validation of the characteristic lengths choice . . . . .                         | 152        |
| 3.2.3    | Navier-Stokes equations with dominant viscous terms . . . . .                     | 152        |
| 3.2.4    | Convection-Diffusion-Reaction equation with dominant viscous terms . . . . .      | 158        |
| 3.3      | APPLICATIONS TO CONVECTION-DIFFUSION PROBLEMS . . . . .                           | 160        |

|          |                                                                                              |            |
|----------|----------------------------------------------------------------------------------------------|------------|
| 3.3.1    | Numerical experiments on the steady convection-diffusion problem . . .                       | 161        |
| 3.4      | APPLICATIONS TO INCOMPRESSIBLE FLOW PROBLEMS . . . . .                                       | 169        |
| 3.4.1    | Driven flow cavity problem (2D) . . . . .                                                    | 170        |
| 3.4.2    | Driven flow cavity problem (3D) . . . . .                                                    | 176        |
| 3.4.3    | Application to a 3D coupled heat transfer and fluid flows problem . . .                      | 180        |
| 3.5      | CONCLUSION . . . . .                                                                         | 183        |
| 3.6      | RÉSUMÉ FRANÇAIS . . . . .                                                                    | 183        |
| <b>4</b> | <b>TIME ACCURATE ANISOTROPIC ADAPTATION FOR UNSTEADY SIMULATIONS</b>                         | <b>185</b> |
| 4.1      | STATE OF THE ART . . . . .                                                                   | 187        |
| 4.1.1    | Stability constraints on explicit advancing . . . . .                                        | 187        |
| 4.1.2    | Local time-stepping . . . . .                                                                | 188        |
| 4.1.3    | Overview on existing time-stepping algorithms . . . . .                                      | 189        |
| 4.1.4    | Generating anisotropic space-time meshes . . . . .                                           | 191        |
| 4.2      | TIME ADAPTATION PROCEDURE . . . . .                                                          | 194        |
| 4.2.1    | Edge-based temporal error estimation . . . . .                                               | 195        |
| 4.2.2    | General space-time adaptive algorithm . . . . .                                              | 198        |
| 4.3      | NUMERICAL VALIDATION . . . . .                                                               | 199        |
| 4.3.1    | Convergence analysis on an analytical test case . . . . .                                    | 199        |
| 4.3.2    | A two-dimensional analytical test case with sinusoidal evolution in time                     | 203        |
| 4.3.3    | Applications on 2D fluid flow problems . . . . .                                             | 203        |
| 4.3.4    | Flow around a circular cylinder (2-D) . . . . .                                              | 211        |
| 4.3.5    | Application to the unsteady convection-diffusion problem . . . . .                           | 213        |
| 4.3.6    | Convection-diffusion in a plane shear flow . . . . .                                         | 214        |
| 4.3.7    | Internal and boundary layers . . . . .                                                       | 215        |
| 4.3.8    | Application to coupled heat transfer and fluid flows . . . . .                               | 217        |
| 4.3.9    | Application to 3D heat transfer and turbulent flow inside an industrial<br>furnace . . . . . | 222        |
| 4.4      | CONCLUSION . . . . .                                                                         | 226        |
| 4.5      | RÉSUMÉ FRANÇAIS . . . . .                                                                    | 227        |
| <b>5</b> | <b>IMMERSED VOLUME METHOD AND MULTI-DOMAIN ADAPTATION</b>                                    | <b>229</b> |
| 5.1      | STATE OF THE ART . . . . .                                                                   | 231        |
| 5.2      | IMMERSED VOLUME METHOD . . . . .                                                             | 234        |
| 5.2.1    | First component: the levelset function . . . . .                                             | 234        |
| 5.2.2    | Second component: anisotropic mesh adaptation . . . . .                                      | 238        |
| 5.2.3    | Third component: mixing laws . . . . .                                                       | 243        |
| 5.3      | MESH ADAPTATION BASED ON SEVERAL FIELDS . . . . .                                            | 246        |
| 5.3.1    | Metric intersection . . . . .                                                                | 246        |
| 5.3.2    | Multi-components metric construction . . . . .                                               | 247        |
| 5.4      | NUMERICAL VALIDATION . . . . .                                                               | 251        |
| 5.4.1    | Flow behind a circular cylinder . . . . .                                                    | 251        |

|          |                                                                                    |            |
|----------|------------------------------------------------------------------------------------|------------|
| 5.4.2    | Turbulent flow behind a F1 racing car at 300km/h . . . . .                         | 254        |
| 5.4.3    | Forced turbulent convection . . . . .                                              | 256        |
| 5.4.4    | Simulation of a rotating helicopter propeller . . . . .                            | 260        |
| 5.5      | CONCLUSION . . . . .                                                               | 262        |
| 5.6      | RÉSUMÉ FRANÇAIS . . . . .                                                          | 263        |
| <b>6</b> | <b>SPACE-TIME SLAB ADAPTIVE MESHING</b>                                            | <b>265</b> |
| 6.1      | STATE OF THE ART . . . . .                                                         | 267        |
| 6.2      | SINGLE TIME-STEP SPACE-TIME ADAPTIVE REMESHING . . . . .                           | 269        |
| 6.3      | PARADOXICAL MESHING: FULL ADAPTIVITY ALGORITHM . . . . .                           | 270        |
| 6.3.1    | Validity of the generated mesh and frequency of remeshing . . . . .                | 271        |
| 6.3.2    | A predictor-corrector approach . . . . .                                           | 272        |
| 6.3.3    | Generating time-slabs for adaptation . . . . .                                     | 273        |
| 6.3.4    | Solution sampling . . . . .                                                        | 274        |
| 6.3.5    | Edge-based error estimation . . . . .                                              | 275        |
| 6.3.6    | Metric construction for a slab of time . . . . .                                   | 275        |
| 6.3.7    | Temporal stretching factors . . . . .                                              | 276        |
| 6.3.8    | Space and time remeshing . . . . .                                                 | 277        |
| 6.3.9    | Convergence of the space-time adaptive algorithm . . . . .                         | 278        |
| 6.4      | PARADOXICAL MESHING ALGORITHM . . . . .                                            | 279        |
| 6.5      | 1D TEMPORAL MESHER . . . . .                                                       | 280        |
| 6.5.1    | Application of the Time mesher . . . . .                                           | 290        |
| 6.6      | NUMERICAL EXAMPLES . . . . .                                                       | 291        |
| 6.6.1    | A two-dimensional rotating Circle . . . . .                                        | 291        |
| 6.6.2    | A two-dimensional analytical test case . . . . .                                   | 294        |
| 6.6.3    | A two-dimensional analytical test case with sinusoidal evolution in time . . . . . | 295        |
| 6.6.4    | Fluid flow for different Reynolds numbers . . . . .                                | 297        |
| 6.6.5    | Flow past a wind turbine . . . . .                                                 | 298        |
| 6.6.6    | Conjugate heat transfer inside an industrial furnace . . . . .                     | 308        |
| 6.7      | CONCLUSION . . . . .                                                               | 310        |
| 6.8      | RÉSUMÉ FRANÇAIS . . . . .                                                          | 311        |
| <b>7</b> | <b>INDUSTRIAL APPLICATIONS</b>                                                     | <b>319</b> |
| 7.1      | INTRODUCTION . . . . .                                                             | 321        |
| 7.2      | 2D QUENCHING OF A HEATED INGOT BY FORCED CONVECTION . . . . .                      | 323        |
| 7.3      | AIR COOLING OF A HAT SHAPED INGOT BY NATURAL CONVECTION . . . . .                  | 329        |
| 7.4      | 3D HEATING OF AN INDUSTRIAL FURNACE . . . . .                                      | 335        |
| 7.5      | 3D COOLING INSIDE A QUENCHING CHAMBER . . . . .                                    | 346        |
| 7.6      | RÉSUMÉ FRANÇAIS . . . . .                                                          | 355        |
|          | <b>CONCLUSION AND OUTLOOK</b>                                                      | <b>357</b> |
|          | <b>BIBLIOGRAPHY</b>                                                                | <b>363</b> |

# LIST OF FIGURES

|      |                                                                                                                                                            |    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 1    | Ice on the inlet face of a fuel-oil heat exchanger similar to that of BA38 Boeing 777. . . . .                                                             | 1  |
| 2    | Zero-isovalue of the support grid's levelset function and plane cuts on the generated adapted mesh, adopted from [Hachem 09]. . . . .                      | 4  |
| 3    | Heat treatment inside an industrial furnace. . . . .                                                                                                       | 6  |
| 4    | Thermal distribution inside an industrial furnace at 7mins of the simulation time, adopted from [Hachem 09]. . . . .                                       | 6  |
| 1.1  | Galerkin vs SUPG weighting function on linear elements adopted from [Brooks 82]. . . . .                                                                   | 25 |
| 1.2  | The auxiliary vector $\mathbf{v}_c$ : projection of the advection direction onto the solution's gradient $\nabla u$ adopted from [Hachem 09]. . . . .      | 29 |
| 1.3  | $P1^+ / P1$ elements. . . . .                                                                                                                              | 32 |
| 1.4  | VMS approach solution decomposition adopted from [Hughes 98] . . . .                                                                                       | 40 |
| 1.5  | Schematic of turbulent flow scales (left) and the three modelling approaches (right), adopted from [Ferziger 96, Hachem 09] . . . . .                      | 46 |
| 1.6  | Flow past a prismatic cylinder: sketch of the geometrical domain ( <i>left</i> ) and the corresponding gradual mesh construction ( <i>right</i> ). . . . . | 55 |
| 1.7  | Flow past a prismatic cylinder: profile of the turbulent eddy viscosity $\tilde{\nu}$ ( <i>left</i> ) and velocity streamlines ( <i>right</i> ). . . . .   | 57 |
| 1.8  | Flow past a prismatic cylinder: drag (solid) and lift (dotted) coefficients time evolution. . . . .                                                        | 58 |
| 1.9  | Natural convection 2D: sketch of the computational domain. . . . .                                                                                         | 59 |
| 1.10 | Natural convection 2D: velocity streamlines at time $t = 20s$ for $Ra=10^6$ , $10^7$ and $10^8$ . . . . .                                                  | 60 |
| 1.11 | Natural convection 2D: temperature isotherms at time $t = 20s$ for $Ra=10^6$ , $10^7$ and $10^8$ . . . . .                                                 | 61 |
| 1.12 | Computational domain after anisotropic mesh adaptation. . . . .                                                                                            | 62 |
| 1.13 | A top view of the furnace (left) and the immersion of an ingot (right) . .                                                                                 | 63 |
| 1.14 | Different angle views of the furnace. . . . .                                                                                                              | 63 |
| 1.15 | Isotherms inside the furnace at two different simulation times. . . . .                                                                                    | 64 |
| 1.16 | Velocity field and streamlines distribution inside the furnace and around the ingots. . . . .                                                              | 65 |
| 1.17 | Velocity vectors on different horizontal cut-planes inside the furnace. . .                                                                                | 66 |

|      |                                                                                                                                                                                                                                                               |     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 1.18 | Velocity vectors on different vertical cut-planes inside the furnace. . . . .                                                                                                                                                                                 | 67  |
| 1.19 | Temperature profile evolution over time captured at the center of the immersed ingots. . . . .                                                                                                                                                                | 68  |
| 1.20 | Temperature profile evolution over time captured at the surface of the immersed ingots. . . . .                                                                                                                                                               | 68  |
| 2.1  | Virtual elements connected to a virtual node outside the domain. . . . .                                                                                                                                                                                      | 77  |
| 2.2  | Example on the local optimization process by the ‘starring’ operator, adopted from [Coupez 00]. . . . .                                                                                                                                                       | 79  |
| 2.3  | Unit ball and linear mapping between the metric space and the canonical space in two and three dimensional spaces. . . . .                                                                                                                                    | 85  |
| 2.4  | Length $\mathbf{X}^{ij}$ of the edge joining nodes $X^i$ and $X^j$ . . . . .                                                                                                                                                                                  | 98  |
| 2.5  | Stretching or shrinking of a spatial edge $\mathbf{X}^{ij}$ as a result of a scaling $s_{ij}$ . . . . .                                                                                                                                                       | 108 |
| 2.6  | Arbitrary point $P$ inside an element connected to the node $X^i$ . . . . .                                                                                                                                                                                   | 111 |
| 2.7  | Angle between the solution’s gradient and the edge $\mathbf{X}^{ij}$ . . . . .                                                                                                                                                                                | 123 |
| 2.8  | $P_1$ interpolation from an old mesh to a new one, adopted from [Boussetta 06]. . . . .                                                                                                                                                                       | 126 |
| 2.9  | Quadratic function: validation of the gradient reconstruction and the edge-based error estimator. . . . .                                                                                                                                                     | 129 |
| 2.10 | Quadratic function: Mesh convergence for the $L_1$ , $L_2$ and $L_3$ norms of the interpolation error . . . . .                                                                                                                                               | 130 |
| 2.11 | Mountains and valleys: Iterative anisotropic refinement obtained within 4 consecutive iteration of the proposed mesh adaptation algorithm. . . . .                                                                                                            | 132 |
| 2.12 | Mountains and valleys: Iterative anisotropic refinement obtained within 8 consecutive iterations in the reference [Borouchaki 01]. . . . .                                                                                                                    | 133 |
| 2.13 | Mountains and valleys: first iteration ( <i>left</i> ) and last one ( <i>right</i> ) in the mesh adaptation process. . . . .                                                                                                                                  | 133 |
| 2.14 | Anisotropic mesh adaptation around the logo of the <i>Ecole des Mines</i> and zoomed snapshots near sharp angles. . . . .                                                                                                                                     | 133 |
| 2.15 | Surface plot (left), and isovalues (right) of the steep radial function described by equation (2.124) on a mesh made up of 7,000 nodes. . . . .                                                                                                               | 134 |
| 2.16 | Steep radial function: anisotropic meshes ( $\mathcal{H}^I$ , $\mathcal{H}^{IP}$ , $\mathcal{H}^E$ , and $\mathcal{H}^{EP}$ respectively from top to bottom) obtained with around 7,000(left) and 20,000(right) nodes after 7 successive adaptations. . . . . | 135 |
| 2.17 | Steep radial function: convergence analysis on the $L_1$ and $L_2$ norms of the interpolation error of the analytical function defined by (2.124). . . . .                                                                                                    | 137 |
| 2.18 | Multiscale function: surface plot (top-left), function isovalues (top-right) and a cross section along the $x = y$ axis (bottom). . . . .                                                                                                                     | 139 |
| 2.19 | Multiscale function: Convergence analysis on the $L_1$ norm of the interpolation error. . . . .                                                                                                                                                               | 139 |

|      |                                                                                                                                                                                                                                                           |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 2.20 | Multiscale function: meshes obtained with 9,000 nodes using the averaged (left) and the privileged (right) length distribution tensor for metric construction. . . . .                                                                                    | 140 |
| 3.1  | Characteristic length for isotropic and anisotropic elements based on classical formulas. . . . .                                                                                                                                                         | 149 |
| 3.2  | Element's characteristic length in the streamline direction. . . . .                                                                                                                                                                                      | 151 |
| 3.3  | Anisotropic mesh obtained for the Poiseuille problem. . . . .                                                                                                                                                                                             | 151 |
| 3.4  | Velocity profiles obtained with the classical characteristic length ( <i>top-left</i> ) and the modified one ( <i>bottom-left</i> ) and 1D vertical cut along the outlet boundary ( <i>right</i> ). . . . .                                               | 152 |
| 3.5  | Numerical solution for $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$ ( <i>from left to right</i> ). . . . .                                                                                                                                                    | 163 |
| 3.6  | Anisotropic meshes for $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$ ( <i>top to bottom</i> ) and zooms on the boundary layer. . . . .                                                                                                                         | 164 |
| 3.7  | $L_\infty$ , $L^2$ and $H^1$ norms of the error versus the number of elements in the mesh for $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$ ( <i>top-left, top-right, and bottom</i> ). . . . .                                                                | 165 |
| 3.8  | Numerical solution (left) and convergence history in the $L^\infty$ , $L^2$ and $H^1$ (right) for $\kappa = 0.005$ and a zoom on the interior layer. . . . .                                                                                              | 165 |
| 3.9  | Anisotropic mesh obtained for $\kappa = 0.005$ . . . . .                                                                                                                                                                                                  | 166 |
| 3.10 | Numerical solution for $\kappa = 10^{-6}$ with its corresponding anisotropically adapted mesh. . . . .                                                                                                                                                    | 166 |
| 3.11 | Numerical solution for $\kappa = 10^{-6}$ , its corresponding anisotropically adapted mesh with different zooming levels near the left boundary. . . . .                                                                                                  | 167 |
| 3.12 | Numerical solution for $\kappa = 10^{-3}$ (right), its corresponding anisotropically adapted mesh with close-ups at the lower boundary to the left( <i>middle-top</i> ) and right( <i>middle-bottom</i> ) of the line $x = \frac{1}{2}$ . . . . .         | 168 |
| 3.13 | Numerical solution for $\kappa = 10^{-3}$ and its corresponding anisotropically adapted mesh. . . . .                                                                                                                                                     | 169 |
| 3.14 | Numerical solution for $\kappa = 10^{-3}$ and its corresponding anisotropically adapted mesh. . . . .                                                                                                                                                     | 169 |
| 3.15 | Driven cavity problem: 2D computational domain and the different cross sections. . . . .                                                                                                                                                                  | 171 |
| 3.16 | Driven cavity problem: anisotropic meshes at Reynolds 1000, 5000 and 10,000. . . . .                                                                                                                                                                      | 171 |
| 3.17 | Driven cavity problem: zooms on the mesh near the right wall. . . . .                                                                                                                                                                                     | 172 |
| 3.18 | Driven cavity problem: comparison of the first component of the velocity field $V_x$ in the mid-plane $x = 0.5$ for $Re = 1,000$ (left), for $Re = 5,000$ (middle) and for $Re = 10,000$ (right). . . . .                                                 | 173 |
| 3.19 | Driven cavity problem: comparison of velocity profiles for $Re = 1,000$ (top), for $Re = 5,000$ (middle) and for $Re = 10,000$ (bottom). Left: Velocity profiles for $V_x$ along $x = 0.1$ . Right: Velocity profiles for $V_x$ along $x = 0.9$ . . . . . | 175 |

|      |                                                                                                                                                                                                                                      |     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 3.20 | Driven cavity problem: snapshots of the anisotropic meshes for Reynolds 20,00 and 100,000 . . . . .                                                                                                                                  | 175 |
| 3.21 | Streamlines snapshots at Reynolds 1000, 3200 and 5000 . . . . .                                                                                                                                                                      | 176 |
| 3.22 | Comparison of the velocity field's first component in the mid-plane $x = 0.5$ for $Re = 1,000$ (left) and for $Re = 3,200$ (right). . . . .                                                                                          | 177 |
| 3.23 | Zooms on the mesh details inside the 3D cavity . . . . .                                                                                                                                                                             | 177 |
| 3.24 | Geometry for the flow behind a NACA body test case. . . . .                                                                                                                                                                          | 178 |
| 3.25 | Mesh around the NACA body with a detail of the anisotropy. . . . .                                                                                                                                                                   | 178 |
| 3.26 | Adaptation driven by the velocity field ( <i>left</i> ) and a combination of velocity and turbulent viscosity ( <i>right</i> ). . . . .                                                                                              | 179 |
| 3.27 | Plots of initial and stationary state meshes. . . . .                                                                                                                                                                                | 179 |
| 3.28 | Plots for Test # 4 at the final time. . . . .                                                                                                                                                                                        | 180 |
| 3.29 | Streamlines and isotherms inside the furnaces at different time instances. . . . .                                                                                                                                                   | 182 |
| 4.1  | Sketch of the edges connected to node $X^i$ in a 4D mesh. . . . .                                                                                                                                                                    | 192 |
| 4.2  | Temporal discretization at the spatial node $X^i$ . . . . .                                                                                                                                                                          | 195 |
| 4.3  | Anisotropic mesh obtained at different time instances (top) and closeups around the radial steep gradient region (bottom). . . . .                                                                                                   | 201 |
| 4.4  | Temporal convergence of the interpolation error with respect to the mesh complexity. . . . .                                                                                                                                         | 201 |
| 4.5  | Variations of the number of time-steps with respect to the imposed number of nodes in the mesh. . . . .                                                                                                                              | 202 |
| 4.6  | Time-step evolution corresponding to adapting the mesh on the $L_1$ , $L_2$ and $L_\infty$ norms. . . . .                                                                                                                            | 202 |
| 4.7  | History of the time-steps (s) obtained with the developed algorithm on problem (4.24) . . . . .                                                                                                                                      | 204 |
| 4.8  | Anisotropic meshes at Reynolds 1,000, 5,000, 10,000 and 20,000 . . . . .                                                                                                                                                             | 205 |
| 4.9  | Comparison of velocity profiles in the mid-planes for $Re = 1,000$ (top) and $Re = 5,000$ (bottom). Left: Velocity profiles for $U_x$ along $x = 0.5$ . Right: Velocity profiles for $U_y$ along $y = 0.5$ . . . . .                 | 206 |
| 4.10 | Comparison of time-averaged velocity profiles in the mid-planes for $Re = 10,000$ (top) and $Re = 20,000$ (bottom). Left: Velocity profiles for $U_x$ along $x = 0.5$ . Right: Velocity profiles for $U_y$ along $y = 0.5$ . . . . . | 207 |
| 4.11 | Comparison of the time-step sizes' variations (s) for $Re = 1,000$ , $Re = 5,000$ , $Re = 10,000$ and $Re = 20,000$ . . . . .                                                                                                        | 209 |
| 4.12 | Comparison of velocity profiles in the mid-planes for $Re = 1,000$ computed using different number of nodes. . . . .                                                                                                                 | 210 |
| 4.13 | Time-steps evolution (s) after introducing an outlet to the cavity at time $t = 4,000$ seconds. . . . .                                                                                                                              | 211 |
| 4.14 | Geometry of the problem adopted from [Schäfer 96]. . . . .                                                                                                                                                                           | 212 |
| 4.15 | Flow past a cylinder test case: evolution of the velocity magnitude (top) and of the mesh (bottom) at different time instances. . . . .                                                                                              | 212 |

|      |                                                                                                                                                                                                                                                                 |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.16 | Flow behind a circular cylinder: time-steps (s) history. . . . .                                                                                                                                                                                                | 213 |
| 4.17 | Flow around a circular cylinder (2D): pressure difference $P_{\text{diff}} = P(0.15, 0.2) - P(0.25, 0.2)$ ( <i>left</i> ) and drag coefficient ( <i>right</i> ). . . . .                                                                                        | 214 |
| 4.18 | The obtained isovalues and the adapted meshes at two different time-step. . . . .                                                                                                                                                                               | 215 |
| 4.19 | Total number of time-steps with respect to the tolerance. . . . .                                                                                                                                                                                               | 216 |
| 4.20 | Numerical solutions at different time-steps with their corresponding anisotropically adapted meshes. . . . .                                                                                                                                                    | 217 |
| 4.21 | Evolution of the time-step size over time. . . . .                                                                                                                                                                                                              | 218 |
| 4.22 | Total number of time-steps with respect to the tolerance. . . . .                                                                                                                                                                                               | 218 |
| 4.23 | Natural convection 2D: temperature isotherms at time $t = 20\text{s}$ for $\text{Ra}=10^6$ and the corresponding adapted mesh. . . . .                                                                                                                          | 219 |
| 4.24 | Temperature fields and velocity streamlines with the corresponding adapted mesh for $\mu = 0.01$ ( <i>top</i> ) and $\mu = 0.0005$ ( <i>bottom</i> ). . . . .                                                                                                   | 221 |
| 4.25 | Time-step evolution (s) for the different possible combinations for adaptation. . . . .                                                                                                                                                                         | 222 |
| 4.26 | Isotherms inside the furnace (Top) and corresponding adapted meshes (bottom) at three different time-steps. . . . .                                                                                                                                             | 223 |
| 4.27 | Time-step evolution (s) for the simulation of an hour of the heating process inside an industrial furnace. . . . .                                                                                                                                              | 225 |
| 4.28 | Time-step evolution (s) for the simulation of an hour of the heating process inside an industrial furnace using the $k - \varepsilon$ turbulence model. An hour of fluid flow ( <i>left</i> ) and zoom on the first 100s ( <i>right</i> ) are depicted. . . . . | 225 |
| 4.29 | Velocity magnitude on a horizontal plane cut along the $z$ -axis at the burner's level obtained at time $t = 50\text{s}$ when adapting the mesh every 5 time-increments ( <i>left</i> ) and every 20 time-increments ( <i>right</i> ). . . . .                  | 226 |
| 5.1  | Schematic for immersing objects inside computational domains. . . . .                                                                                                                                                                                           | 235 |
| 5.2  | Schematic representation of the levelset function for multi-domain problems. . . . .                                                                                                                                                                            | 236 |
| 5.3  | Schematic representation of a node $X^i$ 's projection onto an element on the surface mesh. . . . .                                                                                                                                                             | 236 |
| 5.4  | Zero-isovalues of the levelset function of a missile object in the fitted domain ( <i>left</i> ) and the computational domain ( <i>right</i> ). . . . .                                                                                                         | 237 |
| 5.5  | Anisotropic meshes adapted on the immersed hat shaped solid obtained using the layer based approach ( <i>left</i> ) and the edge-based metric construction developed in this work ( <i>right</i> ). . . . .                                                     | 240 |
| 5.6  | Levelset ( <i>left</i> ) and filtered levelset ( <i>right</i> ) functions corresponding to an immersed missile object. . . . .                                                                                                                                  | 241 |
| 5.7  | Immersed missile object with its corresponding anisotropic mesh. . . . .                                                                                                                                                                                        | 242 |
| 5.8  | Immersed F1 car with its corresponding anisotropic mesh. . . . .                                                                                                                                                                                                | 242 |

|      |                                                                                                                                                                                                                             |     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.9  | Density distribution in a multi-domain simulation on adapted meshes of 4,000 nodes based on a gradual refinement (left) and on an anisotropic refinement (right).                                                           | 245 |
| 5.10 | Anisotropic mesh adaptation around different shapes obtained after 10 adaptive iterations using the metric intersection (left), the multi-components metric construction based on the $L_2$ norm of the edge error (right). | 248 |
| 5.11 | Normalized velocity field (left) the velocity vector (right).                                                                                                                                                               | 249 |
| 5.12 | Anisotropic mesh obtained when adapting on the velocity norm (left) and on its norm and direction (right).                                                                                                                  | 249 |
| 5.13 | Flow behind a circular cylinder: comparison of the velocity profiles (top) obtained using the body fitted and the IVM and their corresponding meshes (bottom) at time $t = 6s$ .                                            | 252 |
| 5.14 | Flow behind a circular cylinder: time history of the drag (top) and lift (bottom) coefficients.                                                                                                                             | 252 |
| 5.15 | Flow behind a F1 racing car: Velocity streamlines, pressure field and their corresponding dynamically adapted meshes at different time instances.                                                                           | 255 |
| 5.16 | Frontal view on the flow behind the F1 racing car at different time instances.                                                                                                                                              | 255 |
| 5.17 | Anisotropic meshes describing the characteristics of the flow behind the F1 racing car.                                                                                                                                     | 255 |
| 5.18 | Forced convection: sketch of the geometry and initial conditions.                                                                                                                                                           | 257 |
| 5.19 | Forced convection: temperature distribution, velocity profiles and the corresponding meshes over time.                                                                                                                      | 258 |
| 5.20 | Forced convection: thermal heating of immersed ingots.                                                                                                                                                                      | 259 |
| 5.21 | Forced convection: time-steps evolution over time (left) and a zoom on the first 200s (right).                                                                                                                              | 260 |
| 5.22 | 2D mesh of the horizontal cut along the blade and the zero-isovalue of the blade's levelset (in solid red line).                                                                                                            | 261 |
| 5.23 | Three time-steps after remeshing: mesh and zero-isovalue of the blade's levelset.                                                                                                                                           | 261 |
| 5.24 | Different snapshots on the mesh capture of the blade's trajectory over time and a zoom on the blade's mesh.                                                                                                                 | 261 |
| 5.25 | History of the time-step sizes for the first 0.05s of the propeller's rotation.                                                                                                                                             | 261 |
| 6.1  | Time slabs $[T^{k-1}, T^k]$ and their corresponding initial meshes.                                                                                                                                                         | 274 |
| 6.2  | Updated time slabs $[T^{k-1}, T^k]$ and their corresponding optimal meshes.                                                                                                                                                 | 278 |
| 6.3  | Example on the failure of the trivial time advancing.                                                                                                                                                                       | 282 |
| 6.4  | Scenario 1: the optimal time-slab size is smaller than the initial one. Consequently, new slabs will be generated in the mesh.                                                                                              | 283 |
| 6.5  | Scenario 2: the optimal time-slab size is equal to the initial one.                                                                                                                                                         | 283 |

|      |                                                                                                                                                                                                                                                    |     |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.6  | Scenario 3: The time-slab $[T^{k-1}, T^k]$ will be stretched. . . . .                                                                                                                                                                              | 284 |
| 6.7  | Scenario 3 (a): The time-slab $[T^{k-1}, T^k]$ will be stretched with $T^{k+1}$ and the new endpoint of the predicted optimal slab $[T^k, T^k + \widetilde{\delta t}_k]$ lying outside $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ . . . . . | 284 |
| 6.8  | Scenario 3 (b): The time-slab $[T^{k-1}, T^k]$ will be stretched with $T^k$ and the new endpoint of the predicted optimal slab $[T^k, T^k + \widetilde{\delta t}_k]$ lying inside $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ . . . . .      | 285 |
| 6.9  | Scenario 3 (c): The time-slab $[T^{k-1}, T^k]$ will be stretched with $T^k, T^{k+1}, T^{k+2}$ lying inside $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ . . . . .                                                                             | 286 |
| 6.10 | Scenario 3 (d): The time-slab $[T^{k-1}, T^k]$ will be stretched with $T^k, T^{k+1}, T^{k+2}$ and $T^{k+2} + \widetilde{\delta t}_{k+2}$ lying inside $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ . . . . .                                  | 286 |
| 6.11 | Scenario 4: the generated optimal mesh $\widetilde{\mathcal{H}}_{k-1}$ up to the end of the simulation. . . . .                                                                                                                                    | 287 |
| 6.12 | Example of application of the time mesher. . . . .                                                                                                                                                                                                 | 290 |
| 6.13 | Paradoxical meshing algorithm applied to a rotating circle for the interval $[t^n, t^{n+1}]$ . . . . .                                                                                                                                             | 292 |
| 6.14 | Time-steps generated by the single time-step adaptation (left) and the paradoxical (right) meshing algorithms. . . . .                                                                                                                             | 292 |
| 6.15 | Instabilities appearing when adapting the mesh using the single time-step meshing algorithm applied to a rotating circle with a fixed time-step $\Delta t = 0.01$ . . . . .                                                                        | 292 |
| 6.16 | Time sub-intervals satisfying the requirement for second order convergence in the anisotropic case. . . . .                                                                                                                                        | 294 |
| 6.17 | Different snapshots reflecting the rotating sphere at the beginning, the middle and the end of the predicted slabs of time. . . . .                                                                                                                | 301 |
| 6.18 | Anisotropic mesh adapted over a slab of time to contain the analytical function defined by (6.20). . . . .                                                                                                                                         | 302 |
| 6.19 | History of the time-steps corresponding to the single time-step and the paradoxical meshing techniques applied to (6.20). . . . .                                                                                                                  | 302 |
| 6.20 | History of the time-steps corresponding to the single time-step and the paradoxical meshing techniques applied to (6.21). . . . .                                                                                                                  | 302 |
| 6.21 | Plot of the mesh and a zoom on the main velocity vortex over the time slab $[1.93, 2.75]$ corresponding to 20 time-increments. . . . .                                                                                                             | 303 |
| 6.22 | Plot of the mesh and the main velocity vortex over the time slab $[6.8027, 7.11646]$ corresponding to 20 time-increments. . . . .                                                                                                                  | 303 |
| 6.23 | Final mesh corresponding to the stable flow inside the cavity (left). Zooms on the mesh close to the right wall (right). . . . .                                                                                                                   | 304 |
| 6.24 | Velocity magnitude over a slab of time with the optimal anisotropic mesh obtained at $Re=100,000$ . . . . .                                                                                                                                        | 304 |
| 6.25 | Comparison of velocity profiles in the mid-planes for $Re = 1,000$ (top), $Re = 5,000$ (middle) and for $Re = 10,000$ (bottom). Left: Velocity profiles for $U_x$ along $x = 0.5$ . Right: Velocity profiles for $U_y$ along $y = 0.5$ . . . . .   | 305 |

|      |                                                                                                                                                                                                                  |     |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 6.26 | Variation of the time-slab sizes generated by the single time-step (solid line) and the paradoxical (dashed line) meshing techniques for the driven cavity problem at Reynolds numbers 5,000 and 20,000. . . . . | 305 |
| 6.27 | Q-criterion contours for the value 0.5 at different time instances. . . . .                                                                                                                                      | 306 |
| 6.28 | Pressure field and velocity streamlines at three successive times. . . . .                                                                                                                                       | 306 |
| 6.29 | Velocity magnitude on the blades and the generated anisotropically adapted meshes. . . . .                                                                                                                       | 307 |
| 6.30 | Initial configuration and mesh for the heat treatment inside an industrial furnace. . . . .                                                                                                                      | 308 |
| 6.31 | Thermal distribution inside the volume and on the surface of the immersed solids. . . . .                                                                                                                        | 313 |
| 6.32 | Thermal distribution inside the volume and in the core of the immersed solids. . . . .                                                                                                                           | 314 |
| 6.33 | Velocity streamlines at different time instances. . . . .                                                                                                                                                        | 315 |
| 6.34 | Velocity field at the beginning and during the course of the simulation. . . . .                                                                                                                                 | 315 |
| 6.35 | Anisotropic mesh generated for the time-slab [26.05,29.65]. . . . .                                                                                                                                              | 316 |
| 6.36 | Temperature distribution at selected nodes inside the volume ( <i>top</i> ) and at the heart of the central thick workpiece ( <i>bottom</i> ). . . . .                                                           | 316 |
| 6.37 | History of the time-step sizes generated by the paradoxical meshing approach for a conjugate heat transfer application. . . . .                                                                                  | 317 |
| 7.1  | Heat treatment of a large hollow steel shell. . . . .                                                                                                                                                            | 321 |
| 7.2  | Heat losses during a furnace heating process taken from [Was 14]. . . . .                                                                                                                                        | 322 |
| 7.3  | General diagram for heat treatment simulation. . . . .                                                                                                                                                           | 324 |
| 7.4  | Computational domain, initial mesh and temperature ( <i>left</i> ) and thermocouples' positions ( <i>right</i> ). . . . .                                                                                        | 325 |
| 7.5  | Temperature profiles and corresponding meshes at times $t = 53s, 175s, 323s, 600s$ (from top to bottom and left to right). . . . .                                                                               | 327 |
| 7.6  | History of temperature at different sensors inside the workpiece. . . . .                                                                                                                                        | 328 |
| 7.7  | Velocity field ( <i>top</i> ) and position of the horizontal cuts ( <i>bottom</i> ). . . . .                                                                                                                     | 328 |
| 7.8  | History of time-step evolution for the forced convection quenching of a heated workpiece. . . . .                                                                                                                | 329 |
| 7.9  | Comparisons of the velocity field's component $v_y$ along the different horizontal cuts at the final time 600s. . . . .                                                                                          | 330 |
| 7.10 | Computational domain, initial mesh and the iso-zero value of the immersed solid. . . . .                                                                                                                         | 331 |
| 7.11 | Thermal evolution inside the enclosure at different time instances with the corresponding anisotropic meshes. . . . .                                                                                            | 333 |
| 7.12 | Temperature distribution inside the hat shaped ingot at times $t = 40s, t = 600s$ and $t = 1600s$ . . . . .                                                                                                      | 334 |
| 7.13 | Thermocouples' positioning inside the hat shaped ingot. . . . .                                                                                                                                                  | 334 |
| 7.14 | Temperature evolution at different locations inside the hat shaped ingot. . . . .                                                                                                                                | 334 |

|      |                                                                                                                                            |     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 7.15 | Temperature evolution at different locations inside the hat shaped workpiece, with and without thermal radiation effects. . . . .          | 335 |
| 7.16 | Generated time-steps for the cooling of a hat shaped solid. . . . .                                                                        | 336 |
| 7.17 | Geometry and initial mesh of the 3D furnace. . . . .                                                                                       | 336 |
| 7.18 | Temperature distribution inside the furnace volume at different time instances. . . . .                                                    | 338 |
| 7.19 | Velocity streamlines inside the furnace volume at different time instances. . . . .                                                        | 339 |
| 7.20 | Anisotropic meshes generated at different time instances. . . . .                                                                          | 340 |
| 7.21 | Sensors' positions inside the cylindrical workpiece. . . . .                                                                               | 341 |
| 7.22 | Temperature distribution in the core of the workpiece. . . . .                                                                             | 342 |
| 7.23 | Temperature distribution at the surface of the workpiece. . . . .                                                                          | 343 |
| 7.24 | Temperature distribution in the furnace volume. . . . .                                                                                    | 343 |
| 7.25 | Thermal evolution on the surface and at the heart of the workpiece. . . . .                                                                | 345 |
| 7.26 | Evolution of the time-step sizes. . . . .                                                                                                  | 345 |
| 7.27 | Initial configuration and mesh of the quenching chamber for the pre-cooling phase. . . . .                                                 | 346 |
| 7.28 | Velocity streamlines during the pre-cooling phase inside the quenching chamber at different time instances. . . . .                        | 347 |
| 7.29 | A top view on the velocity field inside the quenching chamber during the pre-cooling phase at two different time instances. . . . .        | 348 |
| 7.30 | Snapshot on the anisotropic mesh associated with the quenching chamber at time 1100s. . . . .                                              | 348 |
| 7.31 | Velocity streamlines during the cooling phase inside the quenching chamber at different time instances. . . . .                            | 349 |
| 7.32 | Velocity field inside the quenching chamber during the cooling phase at two different time instances. . . . .                              | 350 |
| 7.33 | Temperature distribution inside the quenching chamber during the cooling phase at two different time instances. . . . .                    | 351 |
| 7.34 | Snapshot on the anisotropic mesh associated with the quenching chamber after immersing the hollow solid. . . . .                           | 351 |
| 7.35 | Temperature variation at thermocouples C <sub>1</sub> and C <sub>3</sub> implanted on the surface of the workpiece. . . . .                | 352 |
| 7.36 | Temperature variation at thermocouples C <sub>2</sub> , C <sub>4</sub> and C <sub>5</sub> implanted at the heart of the workpiece. . . . . | 353 |
| 7.37 | Comparison of the thermal distribution at the different thermocouples implanted inside the workpiece. . . . .                              | 354 |



# INTRODUCTION

*“You give but little when you give of your possessions.  
It is when you give of yourself that you truly give.”  
Gibran Khalil Gibran, *The Prophet*.*

On January 17<sup>th</sup> 2008, British Airways flight 38, a Boeing 777 heading from Beijing to London Heathrow airport crash landed on the grass, 330 meters short from the runway on which it was supposed to land. Luckily, no casualties were reported, only one passenger suffered from serious injuries.

Pilots’ accounts and primary investigations showed that the engines did not respond to an increase in throttle requested first by the autopilot and then by the co-pilot who was in the commands at the time of the crash. This resulted in an unwanted decrease in the airplane’s speed at a very critical phase, right before landing. A quick and intelligent reaction from the aircrew allowed the plane to glide to a point where it could land with the safest outcome possible, saving the lives of many passengers and people living in the perimeter of the airport.

Investigations [AAIB 10] by the Air Accidents Investigation Branch (AAIB) into the reasons of this engine failure lasted for about two years. It went over many possibilities which all proved not to cause such a dysfunctionality. The report concluded that, during the course of the flight, the airplane went across very cold environments when flying through the air spaces of both Russia and Sweden; amounts of water, naturally present in the fuel, transformed into ice. When a big demand in fuel was formulated during the landing, some of the remaining ice was moved around the fuel system and clogged the Fuel Oil Heat Exchanger (FOHE) as one can see in figure 1. This diminished the fuel flow through the system causing the throttle to be below the commanded level. A report [Rosenker 09] from the American National Transportation Safety Board required a redesign of the FOHE to render it more tolerant to ice accumulation.



Figure 1 – Ice on the inlet face of a fuel-oil heat exchanger similar to that of BA38 Boeing 777.

Such an accident and terrifying situation could have been avoided if this ice-accumulation scenario had been studied during the design phase of this metal product.

The design of such a piece that intervenes in a sensitive part of an airplane's functioning needs careful and thorough study of the operational possibilities in diverse environments. The FOHE is required for example to have a perfect throughput whatever the fuel flow is and in whatever temperature conditions. Therefore, early studies of a piece's design should include simulation of the interaction between the fuel and solid in all possible what-if scenarios.

The numerical modelling and investigation of this physical situation can be classified as a fluid-structure interaction (FSI) problem. Furthermore, in order to predict the behavior of the workpieces for different weather conditions and during the real time context, long time and large scale simulations shall be conducted. Today, even with the most powerful computational resources, a full simulation is very challenging and highly demanding in terms of computational time. Accurate numerical predictions are unaffordable if no space and time adaptive techniques are used.

FSI problems belong to the variety of multi-physical phenomena. Typical examples can be encountered in a wide range of applications including: aerodynamics like an airflow analysis around aircrafts or racing cars [Hoffman 09, Hoffman 14], biomechanics like blood flow inside arterial membranes [Takizawa 12], turbo-machinery like the study of turbine cooling performance [Sidwell 05], heat treatment inside industrial furnaces [Hachem 10b], etc.

They involve highly coupled systems where fluid and solid parts possessing different material and thermo-dynamic properties interact and influence the mechanical behavior of one-another. These phenomena are intrinsically dynamic and may exhibit heterogeneous scales undertaking a 'cascade' of energy dissipation from the largest to the smallest scale.

The understanding and modelling of these problems have considerably attracted researchers over the last few decades especially with the increasing growth in computer powers and the continuous development of numerical tools. Several mathematical models and numerical methods have been developed in the literature to describe and simulate the underlying phenomena. The accuracy and efficiency of these simulations highly depend on the complexity, the scale and the duration of the considered problem.

Two main immersing techniques can be adopted to ensure the geometric compatibility and the continuity of conditions across the solid/fluid interfaces: the partitioned coupling approach, and the monolithic approach. In the partitioned technique, each of the fluid and solid subdomains is solved separately, then the governing equations are integrated interactively in time and the interface conditions are imposed asynchronously as boundary conditions for the subproblems. The difficulty arises when communicating the data at the interface level. The coupling between the subdomains is ensured using either weakly or strongly coupled methods [Ferziger 02]. In the former case, as its name suggests, the coupling between the subdomains' solutions is very weak and hence the boundary conditions need not to necessarily coincide. At each time increment, a single solution is computed on either sides of the interface [Park 83, Farhat 06]. However, using this scheme, numerical instabilities might arise

depending on the density ratio and the complexity of the geometry hence deteriorating the accuracy of the global solution [Nobile 01, Le Tallec 01, Causin 05]. The strong coupling approach, consists in iteratively resolving the subproblems in a fully coupled manner until the solution stabilizes and hence the interface conditions are well respected [Gerbeau 03, Bathe 04, Fernández 05]. Clearly, this scheme necessitates an important computational cost.

In the context of thermal fluid structure interaction problems using a partitioned method, a heat transfer coefficient between the subdomains should be known and prescribed as boundary condition to ensure an appropriate energy transfer across the interface. Several experimental tests need to be conducted in order to determine this coefficient in terms of the problem's geometry and thermo-mechanical properties. Consequently, for every change in the geometry and/or the physical properties, the experimental tests shall be repeated. This parametric investigation is time consuming and expensive and can thus rapidly become unfeasible. A common industrial practice is to study simplified model problems then to extrapolate the results and draw conjectures on the large scale behavior. However, these procedures can hardly yield optimal configurations and may still be very costly.

On the other hand, monolithic approaches for fluid structure interaction problems involve solving simultaneously a single set of equations for both the fluid and solid parts of the computational domain. The mutual influence of the subdomains will then be naturally accounted for without resorting to coupling models nor experimental data at the interface level, thus enhancing the stability of the computations. The interface tracking method [Lakehal 02] is a monolithic approach where the multi-physical domains' interfaces are tracked by levelsets, which are signed distance functions. The immersed volume method (IVM) [Bernacki 08, Valette 09, Bruchon 09, Hachem 10a, Hachem 13] was defined in that sense and can be applied on a wide range of multi-material applications. The key feature of this approach is to retain the advantages of a monolithic formulation and couple it with an anisotropic mesh well adapted along the fluid/solid interfaces to provide a high resolution at that level. In the monolithic framework, a levelset is used to delimit the different sub-domains. The latter are treated as a single fluid with different material properties, and hence the solution is being evolved with a single set of equations on the global domain. The use of a well adapted mesh along the interfaces ensures an accurate distribution of the thermo-mechanical parameters over the computational domain. This mesh is constructed prior to the simulation and is maintained fixed all over the computational time. It is anisotropic and well adapted along the fluid/solid interfaces and isotropic with a relatively small background mesh size in the rest of the domain. Figure 2 presents an example of the anisotropic mesh used by the immersed volume method that is well adapted to the levelset function of a support grid positioned inside the computational domain. In order to provide a well respected shape in terms of curvatures, sharp angles and thin layers, an anisotropic refinement is generated in the vicinity of the interface and an isotropic mesh size inside the rest of the enclosure, the generated mesh is made up

from 155,015 nodes and 896,539 tetrahedral elements. When dealing with large computational domains and long time simulations, the mesh resolution would be more complex inducing a considerable computational cost.

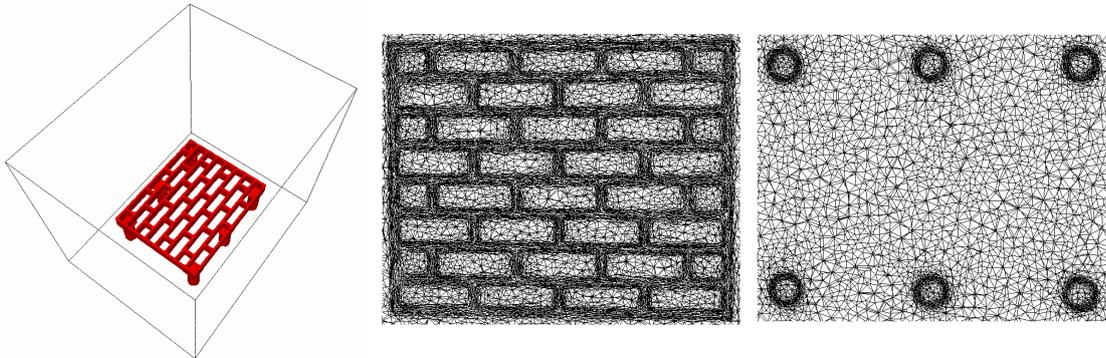


Figure 2 – Zero-iso-value of the support grid’s levelset function and plane cuts on the generated adapted mesh, adopted from [Hachem 09].

## INDUSTRIAL CONTEXT AND NUMERICAL TOOLS

This thesis is part of the **REALisTIC** (laRgE sCAle investigation and Long TIME Computation) research project founded by an industrial consortium in the view of studying and optimizing the heat treatment processes inside industrial furnaces and quenching chambers. As its name suggests, the REALisTIC project aims at providing numerical simulations that mimic the realistic industrial applications. A second interpretation of this acronym can be inferred by decomposing it into “real is tic” by means that real physical phenomena can be reproduced in one click and in a reduced computational time.

The heat treatment cycle is a manufacturing process involving a series of controlled heating, quenching and tempering operations whereby the physical and metallurgical properties of a workpiece material are changed. A good quality production is the result of a well controlled calibration of the heating sequences in accordance with the thermo-physical properties of the metallic component and the heat transfer inside the enclosures. Therefore, from an industrial viewpoint, in order to optimize the industrial compartment throughput and workpieces’ microstructure, it is highly desirable to devise a thermally and energy efficient heating schedule. The research development will also permit the investigation of possible insights to lower the energy consumption and reduce the pollutant emissions and thus improve the environmental health. The study can be carried out using experimental or numerical procedures. Given the thermal history of a metallic ingot and temperature evolution inside the domain, the final characteristics and properties of the product can be determined. However, performing experimental analysis and testing “*what-if*” scenarios necessitates a lot of raw materials and the privatization of machinery for a long time. Therefore, such a procedure can rapidly become unfeasible. On the other hand, given the important growth in com-

puter resources and the important progress of numerical developments, the modelling of long time heat transfer and fluid flows inside large scale complex domains is now possible.

The numerical simulation consists in solving the turbulent flows within the computational domain, combustion and heat transfer at the burner's level and the conduction, convection and thermal radiation at the load's level. The sought method should be capable of providing accurate and efficient computations. Despite the considerable advances in computational fluid dynamics (CFD) and the increasing computer power, different challenges still need to be addressed for providing accurate and efficient simulations. The problem can be cast as a thermal fluid-structure interaction one involving the simultaneous resolution of turbulent flows and conjugate heat transfers between the solid and the fluid subdomains. Due to the deficiency of analytical solution to this coupled system of equations, all hope for resolution culminates in the numerical analysis. To model this multi-material problem, we resort in this thesis work to the immersed volume method whereby the fluid and solid subdomains will be considered as a single fluid with variable material properties and thus one set of equations is considered for the global computational domain. Stabilized finite element approaches will be adopted for the numerical treatment of the physical problems.

## PROBLEMATIC

In the context of the immersed volume method, a fixed mesh anisotropically adapted at the fluid-structure's interfaces and isotropic with a fixed relatively small mesh size is maintained all over the simulation time. When applied on large-scale phenomena like heat treatment inside industrial furnaces, such a generation process leads to a high mesh resolution and requires an important computational cost. Indeed, it was pointed out in [Hachem 09] that using the IVM to simulate 7 minutes of heat transfers and fluid flow inside an industrial furnace (see figure 4), with an anisotropically adapted mesh along the interface of a skid made up from 155,015 nodes and a time-step size 0.001s, required 5 days on 32 2.4 Ghz Opteron cores in parallel (linked by an Infiniband network) although a turbulence model was used to account for the turbulence effect on the general flow and hence to accelerate the computations.

Therefore, accurate and fast algorithms are necessary for the simulation within a reasonable time of the full heating sequence, which is in the order of hours.

## AMBITION OF THIS WORK AND STATE OF THE ART ON EXISTING METHODS IN THE LITERATURE

The accurate and efficient prediction of conjugate heat transfer and fluid flow phenomena of long duration and inside complex large scale domains is certainly a challenging task. Several possible routes can be adopted for achieving this goal: parallel comput-



Figure 3 – Heat treatment inside an industrial furnace.

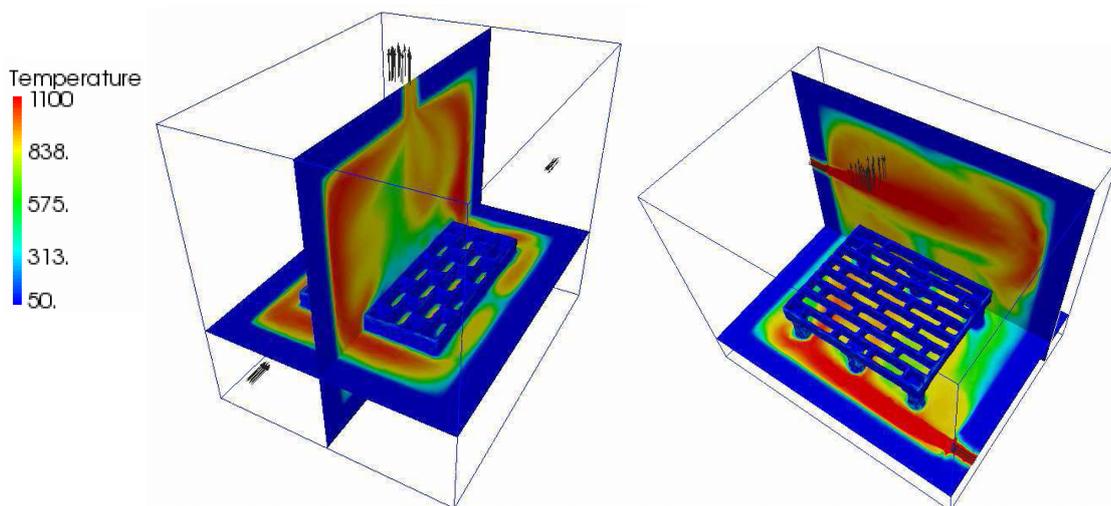


Figure 4 – Thermal distribution inside an industrial furnace at 7mins of the simulation time, adopted from [Hachem 09].

ing, reduced order methods, mesh adaptation, and time adaptation. In this work, a new approach for space and time adaptation will be developed. The method will be suitable for a wide range of problems involving scale heterogeneity and will be subjected to a constraint accounting for the available computer resources. The mesh will be dynamically and automatically adapted to provide an accurate representation of the solution's variations within the least computational cost. The core of this work primarily consists in deriving the methodological aspect and providing a good mathematical ground for the adaptive approaches. The actual application of the algorithms will in turn validate their performance, accuracy and efficiency.

The idea of mesh adaptation was first suggested in the 1970s [Allgower 78] in the view of improving the quality and yielding faster computations of numerical solutions. This subject has obtained more attention in the following years [Lo 90, Craig 87, Adjrid 86, Berger 84] and error estimators found their way inside this enlightened path in order to generate isotropically adapted meshes [Arney 90, Berger 84, Bieterman 86, Chew 89]. Nevertheless, in the absence of the control on the mesh resolution, isotropic mesh adaptation tends to drastically increase the number of elements in the mesh in order to well capture anisotropic features, that is physical phenomena whose variations in one direction are more significant than in the other directions. In these situations, anisotropic meshes are the best candidate as they respond well to the directional aspect of a solution's variations. These types of meshes were first introduced in the late 1980s [Peraire 87, Selmin 92, Löhner 89, Zienkiewicz 94] in the context of the moving front technique. By that time, the applications to 3D were not very successful and showed limited stretching capabilities. The idea of metric based anisotropic adaptation targeting the generation of elements with high aspect ratio started in the 1990s [Mavriplis 90, George 91, Castro-Díaz 97]. An improvement on this method to avoid the generation of obtuse elements was to associate it with local re-meshing strategies [Coupez 91, Castañón 96, Laemmer 97, Gruau 05, Dobrzynski 07, Tremel 07, Loseille 12]. Four major error estimation classes can be distinguished in the literature: the hessian based relying on the solution's hessian information to evaluate the linear interpolation error [Tam 00, Pain 01, Bottasso 04, Gruau 05, Frey 05, Boussetta 06], the a posteriori estimates approximating the discretization error using a theoretical analysis [Kunert 00, Formaggia 03, Picasso 03, Formaggia 04, Micheletti 06, Picasso 06, Hecht 14], the a priori error estimates [Formaggia 01, Huang 05] and the goal oriented estimates that provide mathematical framework for assessing the quality of some functionals [Venditti 03, Micheletti 08a, Alauzet 10b, Peter 12, Yano 12].

The first step in the derivation of an anisotropically adapted mesh involves error estimations which serve as indicators of the zones of the solution's high gradients. From this analysis, a metric tensor is constructed holding information about the optimal mesh sizes and the directions of their prescriptions. In the third step, this tensor is provided to a remesher that will in turn generate the appropriate well adapted mesh. This method for mesh adaptation has proven to be efficient and accurate in responding to the solution's behavior.

On the other hand, since for most physical phenomena it is hard to determine a priori the appropriate time-step size that would respect a good level of efficiency and provide accurate computations, dynamic time adaptation algorithms are highly recommended. The first class of time-stepping techniques involves local time-stepping control whereby different buffer zones evolve at different time-step sizes dictated by local stability conditions. Usually these methods are employed on structured meshes [Tam 03, Lörcher 07, Lörcher 08]. Another possibility consists in limiting the global induced temporal error by a user prescribed tolerance [Sloan 99, Kavetski 02, Chen 04b, Berrone 09]. Heuristic time-stepping algorithms relate the modification of the time-step sizes to the number of iterations needed to solve the nonlinear system at the previous time increment [Celia 92, Rathfelder 94, Paniconi 91]. The selection of time-step sizes in accordance with adjoint weighted residual method [Fidkowski 11, Kast 13] or based on fictive residual methods [Georgiev 09] were also proposed in the literature. However, due to the computational cost associated with the resolution of the an adjoint problem or the resolution of the problem using a Crank-Nicholson and an implicit time discretization, these approaches will not be adopted in this work. Moreover, a fixed point adaptive staggering method was proposed in [Zohdi 07] where the time-steps are automatically adapted using the spectral radius of the considered system of equations to control the convergence of the numerical solutions within a predefined number of iterations. The idea of fixed point iterative time-stepping selection seems very interesting from an accuracy standpoint and will be a source of inspiration in this work for the derivation of a space-time full adaptation method.

Space and time adaptive techniques have also been the subject of research investigations. A space and time residual based approach was developed in [Micheletti 08c] where the error analysis is conducted on space-time slabs and according to the generated temporal error, the time-step size is updated. Furthermore, a space-time method for metric tensor construction is proposed in [Belme 11] whereby the generated anisotropic spatial mesh will provide a dual control on the spatial error and the temporal one. Yet the time-step size is controlled by a CFL condition which in turn is related to the minimum mesh size hence decelerating the computations.

## PRESENT APPROACH

In this thesis, we are interested in providing numerical tools for the resolution of long time conjugate heat transfer and turbulent fluid flows inside large scale computational domains. In order to fulfill this challenge, we pursue the following strategy: we start by exploiting the stabilized finite elements methods that will be used for the resolution process. The application of these numerical tools on fixed meshes with fixed time-step sizes highlights the need for adaptive algorithms to enhance the accuracy and efficiency of the computations. The aim next is to control through anisotropic mesh adaptation and time adaptive algorithms the global space and time interpolation errors.

Starting with the space adaptation, the objective can be cast as an optimization

problem whereby one seeks the metric tensor, a symmetric positive definite tensor, that minimizes the global  $L^p$  norm of the interpolation error of a selected scalar or vector or a combination of several fields of interest. The multi-component metric construction will take into account the multiscale features of the numerical solutions. When provided to a remesher, this tensor enables the optimization of the elements' mesh sizes and their orientations in response to the anisotropy of the physical phenomena. Moreover, edge-based error estimations will be used as indicators of the domain regions where the variables of interest exhibit sharp variations. The construction procedure takes into account a prescribed fixed number of nodes and generates the optimal mesh with this resolution.

When dealing with unsteady Computational Fluid Dynamics (CFD) applications, the time-dimension is an additional backbone component for adaptation. For that reason, the theory will be extended into a space-time formulation. The error equidistribution principle [Babuška 78], stating that minimizing the error over the global domain is equivalent to equidistributing it over the edges of the mesh, permits the splitting of the space-time optimization problem into two folds: on the one hand, an anisotropic mesh adaptation controlling the spatial error will be solved and on the other hand, a time adaptive procedure is followed to control the temporal error. The coupling between them is enforced through the use of a homogenous equidistributed error. This adaptation process will be referred to as the classical adaptation.

The appropriate tuning of the stabilized finite elements methods will also be addressed. It intends to modify the definition of the element's characteristic length in order to handle highly stretched elements.

In the FSI framework, the proposed anisotropic mesh adaptation will not only provide an accurate representation of the fluid/solid interfaces but also dynamically and automatically follow the development of the flow and the heat transfer over time, thus optimizing the repartition of the available nodes over the domain. Indeed, the generated mesh will not be maintained the same over the simulation time, for a given number of nodes, the elements' repartition, direction and stretching will be optimized. As a result of this adaptation, the elements will be concentrated in zones where the solution exhibits a sharp gradient and they will be highly stretched and very coarse in regions of low variations.

Moving toward the ultimate goal of this thesis, we will extend the theory on the classical space-time adaptation to handle slabs of time in a full adaptation approach. The method will be called 'paradoxical meshing', as it will generate meshes that are well predicted for slabs of time to contain and well capture the advancing unknown solution. The method couples the concept of space and time fixed point adaptation [Zohdi 07, Alauzet 10b] with the proposed edge-based error estimation and nodal metric construction developed in this work. In order to optimally redistribute the temporal nodes in accordance with the predicted optimal time-slab sizes, a new 1D temporal mesher will be developed. This novel approach for full adaptivity can be considered as a breakthrough in the resolution of physical problems with moving geometries. When

applied in this context, the method will enhance further the accuracy and efficiency of the numerical solution. Indeed, the constructed mesh will account for the time-evolving geometry and the developing flow characteristics.

## WORKING ENVIRONMENT

This thesis work was developed at the Center for Material Forming, “Centre de Mise En Forme des matériaux” (CEMEF) created in 1974 and located in the Sophia Antipolis technology park, in the south of France.

CEMEF develops global approaches in material forming and computational mechanics (finite element methods, meshing and remeshing, fluid-structure interaction, CFD applications,...). It fosters high synergy between the different fields of study within it. The scientific advancement brought forth by this thesis work was implemented in the CIM-LIB finite element library developed at CEMEF. More precisely, CIMLIB is an abbreviation for “CIM” the advanced Computing in Material forming and “LIB” the scientific library developed by the research team of T. Coupez [Coupez 91, Digonnet 03]. This finite element library is developed in C++, an object oriented programming language, and gathers the developments of the team (Ph.D. students, research assistants and professors). This implementation framework has been highly parallelized [Digonnet 03] and fully automated, thus facilitating the integration of new developments.

## LAYOUT OF THIS THESIS

We intend at the end of this introduction to provide a brief overview that would procure an idea on the different components addressed in the respective chapters of this work.

In chapter 1, we introduce the governing equations for the conjugate heat transfer and fluid flows phenomena. We start by considering the convection-diffusion-reaction problem, then point out the numerical instabilities arising when applying the standard Galerkin numerical formulation in convection dominated regimes and define the adopted time discretization schemes. To deal with the numerical oscillations, appearing in convection dominated problems and at the level of sharp solution’s gradient, the Streamline Upwind Petrov-Galerkin (SUPG) and the Shock Capturing Petrov-Galerkin (SCPG) stabilizations are used. Heat transfers by conduction will also be discussed and stabilized using the time interpolated enriched (EM-I) method. To model the fluid flows inside computational domains, the incompressible Navier-Stokes equation will be considered, highlighting the incompatibility restriction on the spaces of the finite elements used for the velocity and pressure fields. To cope with this problem, the Variational Multiscale approach will be adopted. Then the turbulence models that can be employed at high Reynolds numbers to provide a mean flow representation will be addressed. In order to take into account the effects of heat transfer by thermal radiation, the radiative heat transfer equation is solved. Numerical validations and motivational

examples will be provided next highlighting the need for adaptation techniques to enhance the accuracy and efficiency of computations.

Chapter 2 presents the core derivation of the anisotropic mesh adaptation technique. Starting with a descriptive outline on the mesh generation procedure developed in our laboratory, we pinpoint its capacity to receive as input a metric tensor and generate the corresponding mesh. Then the proposed error estimators along the edges of the mesh will be discussed and the control on the  $L^p$  norm of the interpolation error will be verified. This is followed by the derivation of optimal metric construction and anisotropic mesh adaptation. The chapter concludes with numerical validations on the accuracy and efficiency of the developed method on analytical scalar functions.

An a priori error analysis on the flow equation and the convection-diffusion-reaction problem will be provided in chapter 3 to show that the control on the interpolation error yields a control on the approximation error. Then the tuning of the stabilized finite element methods discussed in chapter 1 to take into account highly stretched elements will be emphasized. We assess afterward the anisotropic mesh adaptation technique on CFD problems.

The theory of adaptation is extended in chapter 4 to include a time-adaptive technique. The coupling between the two adaptations will be acquired through the use of an equi-distributed error. The approach is then validated on several numerical examples, reflecting the convergence, accuracy and efficiency of the method. The developed space and time adaptive approaches will be combined under the name of classical adaptation.

Chapter 5 will present an overview on the Immersed Volume Method whereby a single fluid and solid domains will be considered as a single fluid with variable material properties. This immersing is done using the levelset function and the materials are distributed using mixing laws. These two components of the IVM will then be coupled with the newly developed anisotropic adaptation technique to provide dynamic and automatic mesh updates. The multi-component feature of the developed metric construction approach will then be presented. Numerical results on turbulent fluid flows and conjugate heat transfer problems will be considered to reflect the potential of the IVM and the classical adaptation technique to handle fluid-structure interaction problems with good levels of accuracy and efficiency. In the last example, we apply this combination to simulate moving geometries and point out the decelerating aspect of the classical adaptation approach when applied to this kind of problems. Consequently, we highlight the need for an adaptation that deals with slabs of time.

The classical approach will be extended, in chapter 6, to ensure full adaptation whereby optimal meshes will be constructed for optimal slabs of time. The algorithm for paradoxical meshing will be presented and a 1D temporal mesher will be developed. Detailed information about the implementation of this mesher will be provided. This chapter concludes with numerical validations on analytical functions, a 2D CFD application, a fluid-structure interaction problem involving moving geometries, and a heat treatment problem inside an industrial furnace.

Chapter 7 gathers numerical results obtained for some industrial applications. Comparisons with experimental data and results obtained using fixed meshes and time-step sizes will be provided to reflect the advantages of the proposed adaptation tools.

Finally, in the conclusion, we summarize the achievements of this work and present some possible short and long term extensions to it.

## SCIENTIFIC PUBLICATIONS, COMMUNICATIONS AND RECOGNITIONS

### Publications

- E. Hachem, G. Jannoun, J. Veysset, M. Henri, R. Pierrot, I. Poitraul, E. Massoni & T. Coupez. *Modeling of heat transfer and turbulent flows inside industrial furnaces*. Simulation Modelling Practice and Theory, vol. 30, no. 0, pages 35 – 53, 2013
- T. Coupez, G. Jannoun, N. Nassif, H.C. Nguyen, H. Digonnet & E. Hachem. *Adaptive time-step with anisotropic meshing for incompressible flows*. Journal of Computational Physics, vol. 241, no. 0, pages 195 – 211, 2013
- E. Hachem, G. Jannoun, J. Veysset & T. Coupez. *On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation*. Applied Mathematics and Computation, vol. 232, no. 0, pages 581 – 594, 2014
- G. Jannoun, E. Hachem, J. Veysset & T. Coupez. *Adaptive time-step with anisotropic meshing for unsteady convection-dominated problems*. Applied Mathematical Modelling, vol. Accepted, 2014
- F. Cremonesi, G. Jannoun, L. Billon, J. F. Geneste, Y. H. Grunevald & E. Hachem. *Application of anisotropic edge-based unsteady mesh adaptation to aerodynamics*. International Journal for Numerical Methods in Fluids, vol. submitted, 2014

### Proceedings and oral communications

- G. Jannoun, E. Hachem, J. Veysset & T. Coupez. *Immersed volume method with anisotropic mesh adaptation and time-stepping control for fluid structure interaction and heat transfer applications*. Oral communication at the 11th World Congress on Computational Mechanics WCCM XI, July 2014
- G. Jannoun. *Time accurate anisotropic edge-based mesh adaptation for unsteady flows*. Finalist at the 25th Annual Robert J. Melosh Medal Competition for the Best Student Paper on Finite Element Analysis, April 2014
- G. Jannoun, E. Hachem & T. Coupez. *Stabilized finite element method for incompressible flows using new space and time anisotropic adaptive technique*. In 17th International Conference on Finite Elements in Flow Problems, May 2013

- J. Veysset, G. Jannoun, E. Hachem & T. Coupez. *Immersed NURBS for CFD applications*. accepted in SEMA SIMAI Springer Series, 2014
- G. Jannoun, J. Veysset, E. Hachem & T. Coupez. *Adaptive Time-step with Anisotropic Meshing for Incompressible Flows*. In 41st National Congress on Numerical Analysis, May 2012
- T. Coupez, G. Jannoun, J. Veysset & E. Hachem. *Edge-based anisotropic mesh adaptation for CFD applications*. In Proceedings of the 21st International Meshing Roundtable, pages 567 – 583. Springer, 2013
- G. Jannoun, J. Veysset, E. Hachem, J-F Zaragoci & T. Coupez. *Fully space-time metric based anisotropic mesh adaptation for unsteady problems*. In Adaptive Modeling and Simulation 2013-Proceedings of the 6th International Conference on Adaptive Modeling and Simulation, ADMOS 2013, 2013
- J-F Zaragoci, E. Hachem, G. Jannoun, J. Veysset & T. Coupez. *Méthode d'immersion de pièces mobiles dans des fours industriels*. In 11e Colloque National en Calcul des Structures, CSMA, 2013

## Recognition

The work of this thesis has been recognized as a finalist on the *Robert J. Melosh medal Competition for the Best Student Paper on Finite Element Analysis, held at Duke University, USA*.

## RÉSUMÉ FRANÇAIS

### Contexte général

Les problèmes d'interaction fluide-structure (IFS) sont observés dans plusieurs phénomènes physiques et constituent un sujet d'étude de première importance aussi bien dans le domaine de recherche en mécanique numérique que dans le domaine industriel. Ces problèmes impliquent la résolution de systèmes fortement couplés dans lesquelles les parties fluides et solides, possédant des propriétés physiques et thermo-dynamiques différentes, interagissent et influent le comportement mécanique de l'un l'autre. De plus, les phénomènes physiques concernés sont intrinsèquement dynamiques et peuvent présenter des échelles de variations très hétérogènes.

La compréhension et la modélisation de ces problèmes ont considérablement attiré les chercheurs au cours des dernières décennies surtout avec la croissance des puissances de calcul informatique et le développement continu des outils numériques. Plusieurs modèles mathématiques et méthodes numériques ont été développés dans la littérature pour décrire et simuler les phénomènes sous-jacents. La précision et l'efficacité de ces simulations dépendent fortement de la complexité, l'ampleur et la durée du problème considéré.

## Contexte du projet REALisTIC

Cette thèse se présente dans le cadre du projet REALisTIC, un projet de recherche ANR avec la collaboration de plusieurs partenaires: académique (CEMEF), industriels (Areva, Arcelor-Mittal, Industeel, Aubert & Duval, Snecma-Safran, Transvalor) et développement de logiciels (Sconsultants). L'objectif général du projet est l'optimisation et la simulation numérique de procédés industriels tels que les traitements thermiques de pièces métalliques (chauffage, trempe, refroidissement à l'air libre, ...). Malgré l'amélioration continue des méthodes numériques d'une part et des puissances de calcul informatique d'autre part, des difficultés persistent quant à la simulation de procédés réels dues à la grandeur des installations (dizaines de mètres), la complexité des phénomènes physiques traités et la longue durée du processus (douzaine d'heure).

En effet, un compromis était toujours nécessaire entre précision et efficacité des calculs. Le défi était alors de développer des outils numériques capables d'optimiser les temps de calcul tout en générant des modélisations précises.

## Objectif de la thèse

Le but de cette thèse est de développer des outils numériques qui permettent la simulation de phénomènes physiques complexes avec un bon niveau de précision et un temps de calcul raisonnable.

Dans ce travail, on adopte une méthode d'immersion de volume pour simuler numériquement des problèmes d'interaction fluide-structure. Cette méthode était premièrement couplée avec des maillages fixes bien adaptés aux interfaces fluides/structures. Elle a montré un bon potentiel à simuler avec précision des problèmes multiphysiques complexes. Cependant, le maillage et les pas de temps nécessaires pour de telles précisions engendraient des temps de calculs importants limitant ainsi l'application de la méthode dans un cadre d'applications réelles complexes et industrielles.

Le principal enjeu de cette thèse est de développer des méthodes d'adaptations anisotropes en espace et en temps qui visent à améliorer les précisions de calcul d'une part et à réduire le temps de calcul d'une autre part. L'objectif final étant de simuler des applications 3D réelles et complexes dans de grands espaces et pour de longues durées telles que les procédés industriels de traitement thermiques, on revisite dans le chapitre 1 les solveurs éléments stabilisés qui vont être utilisés dans les simulations numériques. Ensuite, on introduit dans le chapitre 2 la méthode d'adaptation anisotrope du maillage développée. Dans le but d'appliquer cette méthode sur des problèmes CFD, les solveurs éléments finis stabilisés adoptés seront modifiés dans le chapitre 3 pour prendre en compte des maillages fortement anisotropes. Une extension de l'adaptation anisotrope vers une adaptation temporelle sera proposée et sa capacité à réduire le temps de calcul sera mise à l'épreuve dans le chapitre 4. Le couplage de la méthode d'immersion de volume avec l'adaptation anisotrope multi-composantes sera présenté

dans le chapitre 5. Ensuite une nouvelle méthode d'adaptation en espace-temps sera proposée dans le chapitre 6. Elle vise à générer des maillages et des pas de temps qui durent des slabs de temps améliorant ainsi la précision des calculs et réduisant le coût des calculs surtout pour des applications impliquant des objets mobiles. Finalement, la performance, la précision et l'efficacité des outils numériques développés seront mis à l'épreuve dans le chapitre 7 dans le contexte de simulations de procédés industriels et des comparaisons avec des données expérimentales. On clôture la thèse avec de nouvelles propositions d'investigations numériques.



# UNSTEADY COMPUTATIONAL FLUID DYNAMICS



*"We cannot solve our problems with  
the same level of thinking that created them."*  
Albert Einstein.

## CONTENTS

---

|        |                                                           |    |
|--------|-----------------------------------------------------------|----|
| 1.1    | INTRODUCTION                                              | 19 |
| 1.2    | CONVECTION-DIFFUSION-REACTION PROBLEM                     | 19 |
| 1.2.1  | Governing equation                                        | 21 |
| 1.2.2  | Standard Galerkin formulation                             | 22 |
| 1.2.3  | Time integration scheme                                   | 23 |
| 1.2.4  | Streamline Upwind Petrov-Galerkin (SUPG) method           | 24 |
| 1.2.5  | Existence, uniqueness, stability and order of convergence | 25 |
| 1.2.6  | Choice of the stabilizing parameter                       | 27 |
| 1.2.7  | Element characteristic length                             | 28 |
| 1.2.8  | Shock Capturing Petrov-Galerkin                           | 28 |
| 1.2.9  | Transient heat transfer by conduction                     | 30 |
| 1.2.10 | Enriched space approach                                   | 32 |
| 1.2.11 | Time interpolated enriched (EM-I) method                  | 35 |
| 1.3    | THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS                | 35 |
| 1.3.1  | Governing equations                                       | 37 |
| 1.3.2  | Standard Galerkin formulation                             | 38 |
| 1.3.3  | Variational Multiscale (VMS) stabilization method         | 40 |
| 1.3.4  | Choice of the stabilizing parameter                       | 45 |
| 1.4    | TURBULENCE MODELLING                                      | 45 |
| 1.4.1  | Direct Numerical Simulation (DNS)                         | 46 |
| 1.4.2  | Reynolds Averaged Navier-Stokes (RANS)                    | 47 |
| 1.5    | CONJUGATE HEAT TRANSFER                                   | 52 |
| 1.6    | NUMERICAL VALIDATION                                      | 55 |
| 1.6.1  | Turbulent flow behind a prismatic cylinder (2D)           | 55 |
| 1.6.2  | Natural convection (2D)                                   | 58 |

|       |                                                                     |    |
|-------|---------------------------------------------------------------------|----|
| 1.6.3 | Heat treatment of workpieces inside an industrial furnace . . . . . | 62 |
| 1.7   | CONCLUSION . . . . .                                                | 67 |
| 1.8   | RÉSUMÉ FRANÇAIS . . . . .                                           | 69 |

---

**T**HIS chapter aims at introducing the numerical tools used for the modelling and resolution of the conjugate heat transfer and fluid flow problems. It will provide an overview on the adopted stabilized finite element methods involved in order to yield accurate and oscillation free numerical solution. Turbulence models will also be considered to treat the chaotic and fluctuating behavior of the flow at high Reynolds number. Finally numerical validations will assess the performance of the stabilization tools on two and three dimensional problems.

## 1.1 INTRODUCTION

Accurate predictions of conjugate heat transfer and fluid flow phenomena have been the subject of intensive studies from numerical and industrial perspectives in a wide variety of applications including but not limited to: design of heat exchangers, energy conservation and nuclear studies, weather forecast, and automotive and machinery for manufacturing industries. A great effort was employed at both experimental and numerical levels to understand and simulate conjugate heat transfer and turbulent flows for multi-component systems. A common industrial practice, for experimental investigations, is to study simplified model problems then to extrapolate the results and draw conjectures. However, these procedures can hardly yield an optimized setup and may still be very costly. Hence resorting to numerical simulations is expected to provide a better modelling and save both time and economical resources.

The present chapter focuses on the modelling and resolution of the heat transfer and fluid flows applications. The computations consist in simultaneously solving the Navier-Stokes and energy equations. Moreover, in order to take into account the radiation effects, the radiative transfer equation has to be solved. Turbulence models may be used at high Reynolds numbers to provide a mean flow representation. The equations will be discretized using stabilized finite element methods. We note that the implementation and validation of finite element solvers were subject to previous works in the CIMLIB finite element library developed at the CEMEF laboratory [Digonnet 03]. For that purpose, we omit the detailed derivation of the stabilization techniques and refer to [Hachem 09] for a complete study.

After briefly reviewing the set of equations and stabilization methods involved in our study, we present some numerical validations and motivational examples showing the need for adaptation techniques to accelerate the computations while maintaining a good level of accuracy. We will show in the following chapters that, combined with anisotropic meshing and time-adaptation, the stabilized finite element methods can be perceived as powerful tools for approximating the solution of heat transfer and fluid flows in complex geometries.

## 1.2 CONVECTION-DIFFUSION-REACTION PROBLEM

The solution of the transient convection-diffusion-reaction (CDR) problem is a cornerstone for the numerical modelling of a wide range of fluid mechanics applications in particular the heat transfer equation and turbulence models. Over the last two decades, there has been a continuous growth in the development of numerical methods for the resolution of this type of equations. Usually the Galerkin Finite Element method is the first mentioned among the various numerical techniques available to solve these problems.

Unfortunately, the numerical solution of the unsteady convection-diffusion-reaction

equation using the Galerkin formulation normally exhibits global spurious oscillations in convection-dominated regimes, especially in the vicinity of sharp gradients; hence causing a loss of accuracy and stability. Indeed, the standard Galerkin finite element method is equivalent to a central finite difference discretization in which the approximation of the convective term pollutes the solution in convection dominated cases. A variety of stabilization approaches have been proposed to enhance the stability and accuracy of this formulation. Initially, these methods inherited from the upwind finite difference approaches. A stabilization derived using asymmetric test functions in a weighted residual finite element formulation was first proposed in [Christie 76, Heinrich 77]. These methods remove the spurious oscillations but a poor convergence is achieved due to the excessive additional diffusion hence degrading the quality of the numerical solution. Moreover these methods are not consistent with the problem by means that the solution of the variational form is no longer the solution of the convection-diffusion-reaction problem. Significant improvement resulted from the development of the streamline upwind Petrov-Galerkin (SUPG) method proposed by Brooks and Hughes [Brooks 82, Hughes 86b] and reconsidered in [Scovazzi 07]. This method substantially eliminates the instabilities of the standard Galerkin formulation and does not add crosswind diffusion. The convergence and accuracy rates associated to it are higher than the upwind methods. The SUPG approach maintains consistency by adding weighted residual terms to the weak formulation of the problem. They have grown in popularity, especially in fluid dynamics applications, heat transfer and fluid-structure interactions. On the other hand, alternative approaches were proposed in the literature, we recall the Galerkin/Least-Squares (GLS) method in [Harari 94, Harari 92], the gradient Galerkin/least-squares (GGLS) method in [Hughes 89] relying on the minimization of the squared residual of the equation, the unusual stabilized method (US-FEM) in [Franca 95, Franca 97] with the idea of swapping the sign of the Laplacian in the test function, the enriched method with time interpolation in [Hachem 10a], the subgrid scale method in [Codina 00a] and many others; each one of these schemes was used to optimize the performance of the finite element formulation of the unsteady advection-diffusion equation with or without production.

In this work, we focus on the SUPG method as it is the most popular approach for stabilizing the CDR problem. Although SUPG produces accurate and oscillation-free results in regions where the solution does not undertake sharp gradients, spurious oscillations (overshoots and undershoots) might appear in regions where the solution exhibits abrupt changes. This behavior is somehow expected as the method is neither monotone nor monotonicity preserving. An extension of the SUPG method consists in introducing additional, often nonlinear, artificial diffusion in the crosswind direction, so that monotonicity is recovered. The extension is referred to as the discontinuity or shock capturing Petrov Galerkin (SCPG). Despite the progress of the SUPG and the SCPG methods, the amount of added artificial diffusion is tuned by a stabilizing parameter whose choice is still an open problem nowadays. A possible remedy to this issue is the use of variational multiscale methods [Hughes 95] or equivalently the

Residual Free Bubble approach [Franca 98, Franca 97, Brezzi 94, Franca 96] in which the stabilization parameters are naturally driven by the resolution of a boundary value problem at the elements' level. On the other hand, when diffusion dominates, these methods are no longer active and another treatment shall be considered. The enriched space scheme [Baiocchi 93, Brezzi 92, Franca 95] was proposed in the literature, it consists in enriching the space of solution with a space of bubble functions defined at the elements' level. The small scale variations of the solution will be accounted for through an appropriate addition of stabilization terms. An extension of this method to obtain real time solutions was derived in [Hachem 10c] and will be used in this work.

The objective of this section is to recall the convection-diffusion-reaction equation and summarize the stabilized finite element methods that we considered in here for the resolution in case of convection dominated regimes. It is not our intention to provide a thorough analysis of the different possible stabilization techniques as their development is not the contribution of the current work.

This section is organized in the following way: we start with the description of the governing equations for the modelling of convection-diffusion-reaction problem. Then we introduce the standard Galerkin finite element formulation and point out the need for stabilized finite element methods. The third subsection is devoted to the formulation of the Streamline Upwind Petrov-Galerkin (SUPG) technique on unsteady convection dominated problems. It is followed by theoretical study on the existence, uniqueness and convergence order of its solution. The extension of the SUPG to account for abrupt solution changes will be discussed afterward. We move on next to explore how the analysis is carried to model conduction heat transfer problems. The enriched space approach is applied for the stabilization of the latter problems. Finally an extension to an enriched method involving time interpolation is addressed.

### 1.2.1 Governing equation

We consider the following equation that models the transport of a quantity  $u$  through convection, diffusion and reaction:

$$\begin{cases} \partial_t u - \nabla \cdot (\kappa \nabla u) + \mathbf{v} \cdot \nabla u + \sigma u = f, & \text{in } \Omega, \\ u(\cdot, 0) = u_0 & \text{in } \Omega, \\ u = g, & \text{on } \Gamma, \end{cases} \quad (1.1)$$

where  $\Omega \subset \mathbb{R}^d$  is a bounded polyhedral domain with boundary  $\Gamma$ ,  $\kappa \in L^\infty(\Omega)$  is the constant diffusion factor,  $\mathbf{v}(\mathbf{x}) \in [W^{1,\infty}(\Omega)]^2$  is the divergence-free velocity field,  $\sigma(x) \in L^2(\Omega)$  is a reaction term,  $f(x) \in L^2(\Omega)$  is a given source term,  $u_0$  is the initial data and  $g$  is a given boundary condition. For this problem, we distinguish four types of possible boundary layers [Nguyen 09]:

- Regular boundary layers: appear at the outflow boundary defined by

$$\Gamma^- = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} > 0\}$$

where the velocity field  $\mathbf{v}$  is not parallel to the boundary. Their width is of  $\mathcal{O}(\kappa)$ . This order results from the difference between the solution of the convection-diffusion-reaction problem (1.1) and the diffusion-only one (when  $\kappa = 0$ ,  $\sigma = 0$ ).

- Parabolic boundary layers: appear at the characteristic boundary defined by

$$\Gamma^0 = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} = 0\}$$

where the velocity field  $\mathbf{v}$  is parallel to the boundary. Their width is thicker than that of the regular boundary layers and is of  $\mathcal{O}(\sqrt{\kappa})$ .

- Corner boundary layers: appear at the neighborhood of the domain's corners i.e. at the intersection of boundaries.
- Interior layers: appear at the inflow boundary

$$\Gamma^+ = \{\mathbf{x} \in \Gamma : \mathbf{v} \cdot \mathbf{n} < 0\}$$

due to the discontinuities in the data that propagate inside the domain according to the velocity field  $\mathbf{v}$ . Their width is of the same order as that of the parabolic boundary layers i.e.  $\mathcal{O}(\sqrt{\kappa})$ .

In this work, we use the streamline upwind Petrov-Galerkin technique to stabilize the solution and enhance its stability. The method consists in adding artificial diffusion which acts only in the direction of the flow. The SUPG method is globally stable and has good order of accuracy in regions where the solution is smooth. However in practice, if the solution possesses steep layers, it is impossible to achieve optimal orders of convergence i.e. second order for the  $L^2$  norm and first order for the  $H^1$  norm [Nguyen 09].

The SUPG scheme was initially developed for the resolution of the steady CDR problem in order to overcome the instabilities encountered by the standard Galerkin approach when the diffusion term is small. The extension to the transient problem that we present here consists in: first discretizing the equation in time then applying the stabilized finite element method on the resulting spatially-continuous problem.

### 1.2.2 Standard Galerkin formulation

Let us consider the Sobolev space of functions having square integrable first order derivatives  $H_S^1(\Omega)$  in which we are searching for the solution in accordance with its regularity:

$$H_S^1 = \left\{ w \in H^1(\Omega) \mid w = s \forall x \in \Gamma \right\}$$

with

$$H^1(\Omega) = \{ w \in L^2(\Omega), \|\nabla w\| \in L^2(\Omega) \},$$

and  $L^2$  being the Hilbert vector space given by:

$$L^2(\Omega) = \left\{ w(x) \mid \int_{\Omega} |w|^2 dx < \infty \right\}$$

We also define the subspace  $H_0^1(\Omega) \subset H^1(\Omega)$  as the set of functions vanishing on the boundary.

The Galerkin variational formulation is obtained by multiplying equation (1.1) by an appropriate test function  $w \in H_0^1(\Omega)$  and integrating over the computational domain. Without loss of generality, and for simplicity of illustration, we assume a zero Dirichlet boundary condition. The discrete problem becomes:

Find  $u \in H^1(\Omega)$  such that  $u = 0$  on  $\Gamma$  and

$$\left( \frac{\partial u}{\partial t}, w \right) + \underbrace{(\kappa \nabla u, \nabla w) + (\mathbf{v} \cdot \nabla u, w) + (\sigma u, w)}_{\mathcal{B}(u,w)} = \underbrace{(f, w)}_{\mathcal{F}(w)} \quad \forall w \in H_0^1(\Omega), \quad (1.2)$$

where  $(u, w) = \int_{\Omega} uw d\Omega$ .

For the spatial discretization, we consider a finite element partition  $\mathcal{H}_h$  of  $\Omega$  into simplex elements  $K$ . Using these representations, the above-defined functional spaces  $H_s^1(\Omega)$  and  $H_0^1(\Omega)$  are approached by discrete spaces  $H_s^{1h}(\Omega)$  and  $H_0^{1h}(\Omega)$ . Let  $\mathcal{V}_h \subset H^{1h}(\Omega)$  be the space of piecewise linear functions defined on the elements of  $\mathcal{H}_h$ . Thus, the Galerkin finite element formulation reads as:

find  $u_h \in \mathcal{V}_h$  such that:  $u_h = 0$  on  $\Gamma$  and

$$\left( \frac{\partial u_h}{\partial t}, w_h \right) + \mathcal{B}(u_h, w_h) = \mathcal{F}_h(w_h) \quad \forall w_h \in \mathcal{V}_{h,0}(\Omega) \quad (1.3)$$

Finally, the problem defined by equation (1.3) yields the system of first order differential equations:

$$\mathbf{C} \frac{\partial \mathbf{U}}{\partial t} + \mathcal{K} \mathbf{U} = \mathcal{F} \quad (1.4)$$

where  $\mathbf{U}$  is the vector of nodal unknown solution values,  $\mathbf{C}$  the mass matrix,  $\mathcal{K}$  the stiffness matrix corresponding to the convection, diffusion and reaction terms and  $\mathcal{F}$  the vector of internal source term.

### 1.2.3 Time integration scheme

The system of ordinary differential equations (1.4) has to be integrated in time. Using the  $\theta$ -time discretization schemes, the derivative of the solution with respect to time can be approximated at time  $t^n$  by:

$$\mathbf{C} \frac{\mathbf{U}^n - \mathbf{U}^{n-1}}{\Delta t} + \mathcal{K} (\theta \mathbf{U}^n + (1 - \theta) \mathbf{U}^{n-1}) = \mathcal{F} \quad (1.5)$$

where  $\Delta t$  is the time-step,  $n = 1, \dots, N$ , and  $0 \leq \theta \leq 1$ . We recall that this family includes the backward Euler scheme ( $\theta = 1$ ), the Crank-Nicholson scheme ( $\theta = 0.5$ ) and the forward Euler scheme ( $\theta = 0$ ).

### 1.2.3.1 Stability of the standard Galerkin formulation

Since  $w \in \mathcal{V}_{h,0} \subset H_0^1(\Omega)$ , we get, under the assumption  $\sigma - \frac{1}{2} \nabla \cdot \mathbf{v}_h \geq c_0 > 0$  on  $\bar{\Omega}$  for some  $c_0 > 0$ :

$$\begin{aligned} \mathcal{B}(u_h, u_h) &= (\kappa \nabla u_h, \nabla u_h) + (\mathbf{v}_h \cdot \nabla u_h, u_h) + (\sigma u_h, u_h) \\ &= \kappa |u_h|_1^2 + \left( \sigma - \frac{1}{2} \nabla \cdot \mathbf{v}_h, u_h^2 \right) \\ &\geq \underbrace{\kappa |u_h|_1^2 + C \|u_h\|_0^2}_{=: C \|u_h\|_{1,\kappa}^2} \end{aligned} \quad (1.6)$$

where  $C$  is a constant,  $|\cdot|_1$  denotes the  $H^1(\Omega)$  semi-norm and  $\|\cdot\|_0$  the  $L^2(\Omega)$  norm. Compared to the standard  $H^1(\Omega)$  norm  $\|\cdot\|_1$ ,  $\|\cdot\|_{1,\kappa}$  is said to be a weak norm i.e. for the same velocity field, the stability induced by the coercivity inequality (1.6) for  $\kappa < 1$  is much weaker than the stability enjoyed by the case  $\kappa = 1$ .

Challenge arises when convection dominates diffusion i.e. when  $\kappa \ll \|\mathbf{v}\|$ . Note that this is in accordance with a Peclet number,  $Pe = \frac{\|\mathbf{v}\| h}{2\kappa}$  with  $h$  being global characteristic element's length, greater than one. In this case, the standard Galerkin finite element discretization, if the mesh is not well refined, gives rise to node-to-node oscillations in the solution. One way to eliminate these oscillations is to use upwind techniques such as the Streamline Upwind Petrov-Galerkin stabilized finite element method.

### 1.2.4 Streamline Upwind Petrov-Galerkin (SUPG) method

The SUPG approximation of (1.2) is formulated on the finite-dimensional subspaces by introducing an additional weight  $\tau_\kappa \mathbf{v} \cdot \nabla w_h$  to the standard Galerkin weighting functions  $w_h$  in the upwind direction for all terms in the equation. This modification is interpreted as allowing more weight to the nodes in the upstream direction and reducing the weight of the nodes in the downstream direction. Hence, the modified test function is defined by:

$$\tilde{w}_h = w_h + \tau_\kappa \mathbf{v} \cdot \nabla w_h.$$

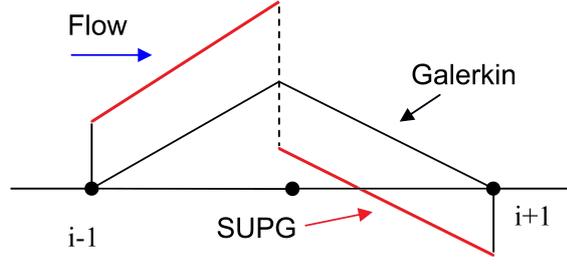


Figure 1.1 – Galerkin vs SUPG weighting function on linear elements adopted from [Brooks 82].

As a result of this upwinding, the variational formulation becomes:

Find  $u_h \in \mathcal{V}_h$  such that  $u_h = 0$  on  $\Gamma$  and

$$\begin{aligned}
 & \left( \frac{\partial u_h}{\partial t}, w_h \right) + \sum_{K \in \mathcal{H}_h} \tau_K \left( \frac{\partial u}{\partial t}, \mathbf{v} \cdot \nabla w_h \right)_K \\
 & \quad + \underbrace{\mathcal{B}(u_h, w_h) + \sum_{K \in \mathcal{H}_h} \tau_K (\mathcal{R}u_h, \mathbf{v} \cdot \nabla w_h)_K}_{\mathcal{B}_\tau(u_h, w_h)} \\
 & = \underbrace{\mathcal{F}(w_h) + \sum_{K \in \mathcal{K}} \tau_K (f, \mathbf{v} \cdot \nabla w_h)_K}_{\mathcal{F}_\tau(w_h)} \quad \forall w_h \in H_0^1(\Omega),
 \end{aligned} \tag{1.7}$$

where  $\tau_K$  is a stabilization parameter that tunes the amplitude of the added weight and

$$\mathcal{R}u_h = -\nabla \cdot \kappa \nabla u_h + \mathbf{v} \cdot \nabla u_h + \sigma u_h.$$

Comparing equations (1.2) and (1.7) we notice that the SUPG formulation is equivalent to introducing a local diffusion  $\mathcal{O}(\tau_K)$  along the convection streamline direction weighted by a stabilization parameter. As the additional stabilizing terms vanish for the exact solution of the problem, the SUPG discretization is consistent.

Finally, the problem defined by equation (1.7) can be reformulated into a system of first order differential equations as in equation (1.4), and the temporal discretization is performed in a way analogous to the standard Galerkin formulation (1.5).

### 1.2.5 Existence, uniqueness, stability and order of convergence

We are now interested in studying the stability of the solution computed using the SUPG formulation. We note that the coercivity analysis follows the lines of [Kopteva 05]. Let  $\sigma_K = \max_{x \in K} |\sigma(x)|$ ,  $\forall K \in \mathcal{H}_h$ . Without loss of generality, we assume that the mesh is uniform with local inverse inequality:

$$\|\Delta u_h\|_{0,K} \leq \zeta h_K^{-1} |u|_{1,K} \quad \forall K \in \mathcal{H}_h, \tag{1.8}$$

where  $h_K$  is the characteristic length of element  $K$ ,  $\zeta$  is a constant,  $\|\cdot\|_{0,K}$  is the  $L^2(K)$  norm, and  $|\cdot|_{1,K}$  is the  $H^1(K)$  semi-norm.

We assume that  $\tau_K$  satisfies:

$$0 \leq \tau_K \leq \frac{1}{2} \min \left( \frac{c_0}{\sigma_K^2}, \frac{h_K^2}{\kappa \zeta^2} \right) \quad \forall K \in \mathcal{H}_h. \quad (1.9)$$

Then for every  $u \in \mathcal{V}^h$ , we have:

$$\begin{aligned} \mathcal{B}_\tau(u_h, w_h) &= \mathcal{B}(u_h, u_h) + \sum_{K \in \mathcal{H}_h} \tau_K (-\kappa \Delta u_h + \mathbf{v} \cdot \nabla u_h + \sigma u_h, \mathbf{v} \cdot \nabla u_h)_K \\ &\geq \kappa |u_h|_1^2 + C \|u_h\|_0^2 + \sum_{K \in \mathcal{H}_h} \tau_K (-\kappa \Delta u_h + \sigma u_h, \mathbf{v} \cdot \nabla u_h)_K + \sum_{K \in \mathcal{H}_h} \tau_K (\mathbf{v} \cdot \nabla u_h, \mathbf{v} \cdot \nabla u_h)_K \\ &\geq \kappa |u_h|_1^2 + C \|u_h\|_0^2 + \sum_{K \in \mathcal{H}_h} \tau_K \|\mathbf{v} \cdot \nabla u_h\|_{0,K}^2 + \sum_{K \in \mathcal{H}_h} \tau_K (-\kappa \Delta u_h + \sigma u_h, \mathbf{v} \cdot \nabla u_h)_K. \end{aligned} \quad (1.10)$$

Using the local inverse inequality together with assumption (1.9) yields:

$$\begin{aligned} \left| \sum_{K \in \mathcal{H}_h} \tau_K (-\kappa \Delta u_h + \sigma u_h, \mathbf{v} \cdot \nabla u_h)_K \right| &\leq \sum_{K \in \mathcal{H}_h} \kappa^2 \tau_K \|\Delta u\|_{0,K}^2 \\ &\quad + \sum_{K \in \mathcal{H}_h} \sigma_K^2 \tau_K \|u\|_{0,K}^2 + \frac{1}{2} \sum_{K \in \mathcal{H}_h} \tau_K \|\mathbf{v} \cdot \nabla u_h\|_{0,K}^2 \\ &\leq \frac{\kappa}{2} |u|_1^2 + \frac{c}{2} \|u\|_0^2 + \frac{1}{2} \sum_{K \in \mathcal{H}_h} \tau_K \|\mathbf{v} \cdot \nabla u_h\|_{0,K}^2. \end{aligned} \quad (1.11)$$

Consequently, the energy streamline upwind norm is defined as:

$$\|w\|_{\text{SUPG}} := \left( \kappa |w|_1^2 + \sum_{K \in \mathcal{H}_h} \tau_K \|\mathbf{v} \cdot \nabla w\|_{0,K}^2 + C \|w\|_0^2 \right)^{\frac{1}{2}},$$

then the discrete bilinear form is coercive with respect to this norm:

$$\mathcal{B}_\tau(u_h, u_h) \leq \frac{1}{2} \|u_h\|_{\text{SUPG}}^2. \quad (1.12)$$

The existence and uniqueness of the solution obtained with the SUPG method can be studied using the Lax-Milgram theorem. We recall that in convection dominated problems, the standard Galerkin formulation fails to satisfy the coercivity condition leading to instabilities in the solution. For the energy norm, the linear functional  $\mathcal{F}_\tau(\cdot)$  is continuous and the bilinear form  $\mathcal{B}_\tau(\cdot, \cdot)$  satisfies:

$$\begin{aligned} \mathcal{B}(u_h, u_h) &\geq \frac{1}{2} \|u_h\|_{\text{SUPG}}^2 \quad \forall u \in H_0^1(\Omega) \\ \text{and } \mathcal{B}_\tau(u_h, w_h) &\leq \|u_h\|_{\text{SUPG}} \cdot \left( \|\mathbf{v}\|^{-1/2} \|u_h\|_\infty \|w_h\| \right). \end{aligned}$$

Thus the bilinear form is coercive and continuous with respect to the energy norm.

Now that we have discussed existence and uniqueness and derived a stability bound for the SUPG method, we move on to consider the convergence of the method. For

that purpose, we refer to [Nguyen 09, Zhou 97] for the proofs on the error estimates summarized in the following theorems.

**Theorem:** If linear approximation is used on a finite element triangulation, then there exists a constant  $\mathcal{C}$ , asymptotically proportional to the mesh Peclet number with:

$$\|\nabla(u - u_h)\| \leq \mathcal{C} h_K^{\max} \|u\|_{H^2(\Omega)}.$$

**Theorem:** If piecewise linear approximation is used on a uniform finite element triangulation with  $h > 2\kappa$ , then there exists a bounded constant  $\mathcal{C}$ , independent of the diffusion coefficient with:

$$\|u - u_h\|_{\text{SUPG}} \leq \mathcal{C} (h_K^{\max})^{3/2} \|u\|_{H^2(\Omega)}.$$

From this result using the energy norm, follows an  $O(h^{\frac{3}{2}})$  convergence in the  $L^2$  norm. Additional error analysis is presented in [Zhou 97].

Nevertheless, the error bounds in the  $L^2$  norm show a loss in the convergence rate by half-an-order for the SUPG method which reflects that in the presence of steep layers and convection dominated regimes it is difficult to achieve optimal convergence rates. Accuracy and convergence improvements can be obtained by resorting to anisotropic mesh adaptations (see chapter 2).

### 1.2.6 Choice of the stabilizing parameter

The stabilizing parameter  $\tau_K$  will govern the amplitude of the added artificial diffusion in the direction of the streamline as it determines and calibrates the amount of upwinding weights locally in each element.

An important drawback of SUPG stabilized methods is that an optimal choice of stabilization parameters is still problematic. The SUPG method was the center of excessive research work over the last two decades, much of which was devoted to the choice of the stabilization parameter [Brooks 82, Hughes 86b, Franca 92a, Codina 98, Codina 00a, Franca 00]. Theoretical analysis has led to some bounds on  $\tau_K$  for which the SUPG method is stable and yields a (quasi-)optimal convergence of the discrete solution.

An analysis of convection-diffusion problems, in one space dimension and on a uniform mesh, has shown in [Formaggia 04, Mitchell 78] that the SUPG solution is nodally exact for continuous piecewise linear finite elements if

$$\tau_K = \frac{h_K}{2\|\mathbf{v}\|_2} \left( \coth(\text{Pe}_K) - \frac{1}{\text{Pe}_K} \right),$$

where  $\text{Pe}_K = \frac{\|\mathbf{v}\|_2 h_K}{2|\kappa|}$  is the mesh Peclet number and  $h_K$  is the element size. However if  $\mathbf{v}$  or  $f$  are not constant, this choice of stabilization parameter does not generally lead to a nodally exact discrete solution. Another possible suggestion for the stabilizing

parameter taken from [Brooks 82] is:

$$\tau_k = \frac{h_K}{2\|\mathbf{v}\|_2} \max \left\{ 0, 1 - \frac{1}{\text{Pe}_K} \right\}.$$

Experimental studies have shown that in a convection dominated regime for large Peclet numbers this choice of the stabilizing parameters gives more stable numerical results. Note that  $\tau_k$  is positive for both convection and diffusion dominated problems with:

$$\tau_k = \begin{cases} c_1 h_K & \text{for } \text{Pe}_K > 3, \\ c_2 \frac{h_K^2}{\kappa} & \text{for } \text{Pe}_K \leq 3 \end{cases},$$

$c_1$  and  $c_2$  being positive constants.

For negative reaction terms, Codina suggested in [Codina 00a] a formula emanating from a Fourier analysis and the maximum principle. A similar idea taking into account any sign of the reaction term was provided in [Shakib 91]. From these two works, we infer the following parameter:

$$\tau_k = \left( \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2\|\mathbf{v}\|_K}{h_K} \right)^2 + 9 \left( \frac{4\kappa}{h_K^2} \right)^2 + \sigma^2 \right)^{-1/2}. \quad (1.14)$$

During the course of this thesis, we adopt this formula for the stabilization parameter of the Streamline Upwind Petrov-Galerkin method. For an extended overview on possible choices for the stabilization factor we refer the reader to [Hachem 09].

### 1.2.7 Element characteristic length

The element characteristic length significantly influences the amount of added diffusion in the SUPG method. A common choice would be to simply define this measure as the diameter of the element. In chapter 2, when using the stabilized finite element method on anisotropic meshes we will affect the choice of the element characteristic length to well respond to the nature of the mesh and result in conservation and accuracy features.

### 1.2.8 Shock Capturing Petrov-Galerkin

When approximating smooth functions, i.e. when the solution does not exhibit abrupt changes, the numerical solution to a convection dominated problem using the SUPG method is accurate and oscillation free. However this method still presents spurious oscillations (overshoots and undershoots) in the regions along steep layers. This indicates that it is not enough to stabilize only in the streamline direction. Although the amplitude of these nonphysical oscillations are smaller than those produced by the standard Galerkin formulation, they should be treated. The unsatisfactory behavior is related to the fact that the SUPG is neither monotone nor monotonicity preserving. In other words, these over/undershoots stem from the fact that the added artificial diffusion is in the streamline direction whereas at the level of boundary layers the arti-

ficial diffusion should be added along the solution's gradient. To remedy this problem, one resorts to methods that introduce additional crosswind diffusion. These schemes are often referred to as discontinuity/shock capturing methods. The key idea of these methods is to add a non-linear term to the SUPG formulation, resulting in a higher regularity of the function's derivative in the crosswind direction. Several approaches for deriving stabilization methods along boundary and interior layers were proposed in the literature. The most popular ones are the Consistent Approximate Upwind (CAU) methods [Galeão 88, Galeão 04] and the Spurious Oscillations at Layers Diminishing (SOLD) or equivalently the Shock Capturing Petrov-Galerkin (SCPG) methods [Tezduyar 86, Hughes 86a, Volker 07]. Nevertheless, very few are the attempts in the context of transient heat transfer problems.

The Shock capturing Petrov-Galerkin (SCPG) scheme, an extension of the SUPG method, introduces an extra term known as discontinuity-capturing operator  $\tau_K^c \mathbf{v}_c \cdot \nabla w_h$ . The extra term affects, and provides extra control, on the numerical solution in the crosswind direction i.e. in the direction of the solution's gradient  $\nabla u_h$ . The new weighting function taking into account this added term is given by:

$$\tilde{w}_h = w_h + \tau_K \mathbf{v} \cdot \nabla w_h + \tau_K^c \mathbf{v}_c \cdot \nabla w_h,$$

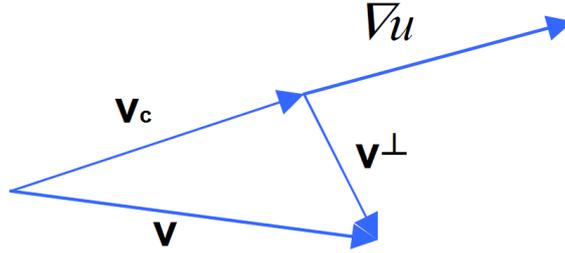


Figure 1.2 – The auxiliary vector  $\mathbf{v}_c$ : projection of the advection direction onto the solution's gradient  $\nabla u$  adopted from [Hachem 09].

where,

$$\mathbf{v}_c = \begin{cases} \frac{\mathbf{v} \cdot \nabla u_h}{\|\nabla u_h\|_2} \nabla u_h & \text{if } \nabla u_h \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (1.15)$$

One can clearly identify the nonlinearity of the newly defined term. Applying the Galerkin procedure to equation (1.1) taking into account the new weighting function,

yields:

$$\begin{aligned}
 (\mathcal{R}u_h, \tilde{w}_h) &= \left( \frac{\partial u_h}{\partial t}, w_h \right) + \sum_{K \in \mathcal{H}_h} \tau_K \left( \frac{\partial u}{\partial t}, \mathbf{v} \cdot \nabla w_h \right)_K + \mathcal{B}_\tau(u_h, w_h) \\
 &+ \underbrace{\sum_{K \in \mathcal{H}_h} \tau_K \left( \frac{\partial u}{\partial t}, \mathbf{v}_c \cdot \nabla w_h \right)_K + \sum_{K \in \mathcal{H}_h} (\mathcal{R}u_h, \tau_K^c \mathbf{v}_c \cdot \nabla w_h)_K}_{\text{SSCPG}(u_h, w_h): \text{discontinuity-capturing term}} \\
 &= (f, \tilde{w}_h).
 \end{aligned} \tag{1.16}$$

There are many possible choices for the stabilization parameter  $\tau_K^c$ . In most cases, the latter is closely related to the SUPG stabilization factor. One possibility is to set  $\tau_K^c$  using the same formula as  $\tau_K$  and replace the velocity field  $\mathbf{v}$  by  $\mathbf{v}_c$ . However an important drawback arises in the case when  $\mathbf{v} \cong \mathbf{v}_c$  as a double artificial diffusivity is induced. A remedy to this disadvantage was proposed by Tezduyar and Park in [Tezduyar 86] where the parameter is defined by:

$$\tau_K^c = \frac{h_K}{2|\mathbf{v}_c|} \eta \left( \frac{|\mathbf{v}_c|}{2|\mathbf{v}|} \right), \tag{1.17}$$

with  $\eta(\beta) = 2\beta(1 - \beta)$ ,  $\beta \in [0, 1]$ . In the current work, we adopt this stabilizing parameter for the shock capturing Petrov Galerkin method.

However, sometimes excessive diffusion due to the SCPG stabilization might affect the solution's accuracy. Therefore a tuning needs to be employed so as to deactivate the added diffusion in regions where the solution is smooth. One possibility, taken from [Galeão 04] and [Galeão 88], that we adopt in this work, is to control the stabilizing parameter using a feedback function defined by:

$$\alpha_h = \frac{\mathbf{v} \cdot \nabla u_h}{\mathcal{R}u_h}.$$

The consequent modified factors will have the following form:

$$\tau_K^c = \tau_K^c \max \left\{ 0, \frac{\|\mathbf{v}\|}{\|\mathbf{v}_c\|} - \zeta \right\}, \text{ with } \zeta = \max(1, \alpha_h)$$

The previously mentioned methods will serve for the resolution of convection dominated regimes. We move on now to tackle another form of the heat transfer equation.

### 1.2.9 Transient heat transfer by conduction

In this subsection, we are interested in modelling and resolving heat transfer by conduction, or unsteady diffusion problems. These problems are encountered in many industrial applications including solid quenching processes, solidification, metal forming and casting, and polymer injection and molding. The use of the standard Galerkin formulation yields unphysical instabilities when solving unsteady diffusion problems,

when  $\kappa \leq h^2 \Delta t^{-1}$ , i.e. the diffusion factor  $\kappa$  is small, or a small time-stepping is applied, with  $h$  being the element mesh size [Badia 09]. In order to remedy these instabilities, the idea is to enrich the finite element space of solutions with a space of bubbles followed by a linear interpolation in time allowing a time synchronization. The oscillation-free solution stems from adding a stabilization term calibrated by a local time-step. We adopt in this thesis the Enriched Method with Interpolation (EM-I) proposed in [Hachem 10c].

For a thermal conductivity  $\lambda$ , a mass density  $\rho$  and a specific heat capacity  $C_p$ , the transient heat transfer by conduction is modelled as follows: *Starting with an initial domain temperature  $T_0$ , find the temperature field  $T \in \mathbb{R}$  such that:*

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (\lambda \nabla T) = f \quad \text{on } \Omega \times (0, t_f) \quad (1.18a)$$

$$T = T_s \quad \text{on } \Gamma_s \quad (1.18b)$$

$$\lambda \nabla T \cdot n = q_w \quad \text{on } \Gamma_q \quad (1.18c)$$

$$\lambda \nabla T \cdot n = -h_c(T - T_{\text{out}}) \quad \text{on } \Gamma_c \quad (1.18d)$$

with  $f$  being an energy source term,  $T_s$  an imposed temperature on a portion of the boundary denoted by  $\Gamma_s$  and  $q_w$  an imposed heat flux at boundary  $\Gamma_q$ . Moreover, using a heat transfer coefficient  $h_c$  and the temperature outside the domain  $T_{\text{out}}$ , a convective heat transfer condition  $-h_c(T - T_{\text{out}})$  is imposed on  $\Gamma_c$ .

We start the formulation by considering the same functional spaces defined above for the Galerkin method, with  $H_s^1$  the solution space. We also derive the variational form by multiplying by an appropriate test function and integrating over the domain. Then we discretize the computational domain using a finite element mesh  $\mathcal{H}_h$ . Eventually, the obtained Galerkin finite element formulation of problem (1.18a) reads as:

find  $T_h \in \mathcal{V}_h$  such that:

$$\begin{aligned} & \left( \rho C_p \frac{\partial T_h}{\partial t}, w_h \right)_\Omega + (\lambda \nabla T_h, \nabla w_h)_\Omega + (h_c T_h, w_h)_{\Gamma_c} \\ & = (f_h, w_h)_\Omega + (q_w, w_h)_{\Gamma_q} + (h_c T_{\text{out}}, w_h)_{\Gamma_c} \quad \forall w_h \in \mathcal{V}_h^0 \end{aligned} \quad (1.19)$$

This problem reduces into a system of first order differential equations:

$$\mathbf{C} \frac{\partial \mathbf{T}}{\partial t} + \mathcal{K} \mathbf{T} = \mathcal{F} \quad (1.20)$$

In here, the nodal temperature field is represented by the vector  $\mathbf{T}$ ,  $\mathbf{C}$  is the capacitance matrix with entries  $\mathbf{C}_{ij} = (\rho C_p \varphi_i, \varphi_j)_\Omega$ ,  $\mathcal{K}$  the conductivity matrix with components  $\mathcal{K}_{ij} = (\lambda \nabla \varphi_i, \nabla \varphi_j)_\Omega + (h \varphi_i, \varphi_j)_{\Gamma_c}$  and  $\mathcal{F}$  is a vector taking into account the internal source term and the external flux with entries  $\mathcal{F}_i = (f_h, \varphi_i)_\Omega + (q_w, \varphi_i)_\Gamma + (h T_{\text{out}}, \varphi_i)_{\Gamma_c}$ .

The system is then discretized in time using the  $\theta$ -scheme in an analogous way to (1.5).

### 1.2.10 Enriched space approach

As we have mentioned earlier, in the case of heat transfer by conduction, the standard Galerkin formulation yields a poor numerical solution if  $\Delta t \leq \frac{h^2}{\kappa}$ , where  $\kappa = \frac{\lambda}{\rho C_p}$  is the diffusivity coefficient. For that reason, we resort to the use of the enriched space approach to stabilize the problem [Baiocchi 93, Brezzi 92, Franca 95]. However, using this method, the solution is shifted backwards with respect to the real time; for that purpose, we provide next an extension following the lines of [Hachem 10c] to achieve a time-synchronous solution.

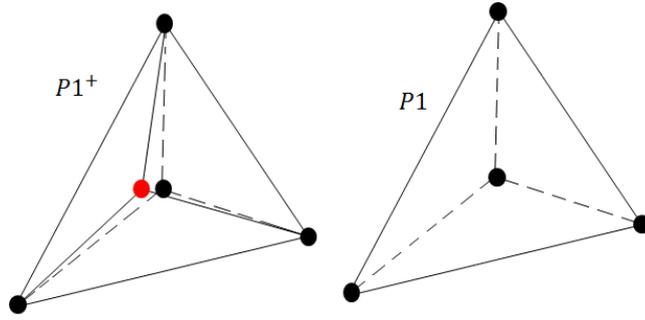


Figure 1.3 –  $P1^+ / P1$  elements.

The key idea of this method is to locally enrich the finite element space  $\mathcal{V}_h$ , spanned by continuous piecewise linear polynomials, with a space of bubbles  $\mathcal{V}_b$  whose support lies inside the mesh elements. We consider an element  $K \in \mathcal{H}_h$ . The bubble function associated with  $K$  is such that:

$$\begin{cases} \psi(x) > 0 & \forall x \in K \\ \psi(x) = 0 & \forall x \in \partial K \\ \psi(x) = 1 & \text{at the centroid of } K \end{cases} \quad (1.21)$$

One now seeks the solution in the space  $\tilde{\mathcal{V}} = \mathcal{V}_h \oplus \mathcal{V}_b$ . In the same manner, the space of test functions will be decomposed into a macro-scale and a micro-scale  $\tilde{\mathcal{W}} = \mathcal{W}_h^0 \oplus \mathcal{W}_b$ . For the sake of simplicity of illustration, we assume a zero Dirichlet boundary condition on  $\Gamma$ .

Therefore the problem's variational formulation will be given by:

Find  $\tilde{T}_h = T_h + \sum_{K \in \mathcal{H}_h} T_{b_K} \psi$  in  $\tilde{\mathcal{V}}$  such that:

$$\left( \rho C_p \frac{\partial \tilde{T}_h}{\partial t}, \tilde{w} \right)_\Omega + \left( \lambda \nabla \tilde{T}_h, \nabla \tilde{w} \right)_\Omega = (f_h, \tilde{w})_\Omega \quad \forall \tilde{w} \in \tilde{\mathcal{W}} \quad (1.22)$$

From the fine-scale expansion proposed in [Masud 04], we write:

$$T_b = \sum_{K \in \mathcal{H}_h} T_{b_K} \psi \quad \text{and} \quad w_b = \sum_{K \in \mathcal{H}_h} w_{b_K} \psi \quad (1.23)$$

At the element's level the two problems can be decoupled and one can then resolve a large-scale and a small-scale ones. The fine-scale problem reads, for all  $w_{b_K}$ :

$$\begin{aligned} \left( \rho C_p \frac{\partial T_h}{\partial t}, w_{b_K} \psi \right)_K + \left( \rho C_p \frac{\partial T_{b_K} \psi}{\partial t}, w_{b_K} \psi \right)_K + (\lambda \nabla T_h, \nabla w_{b_K} \psi)_K \\ + (\lambda \nabla T_{b_K} \psi, \nabla w_{b_K} \psi)_K = (f_h, w_{b_K} \psi)_K \end{aligned} \quad (1.24)$$

Since the large-scale solution is defined linearly on the elements, the term  $(\lambda \nabla T_h, \nabla w_{b_K} \psi)_K$  vanishes. On the other hand, as the equation is satisfied for any choice of  $w_{b_K}$  then it is in particular valid for:

$$w_{b_K} = \begin{cases} 1 & \text{on the elements' center,} \\ 0 & \text{elsewhere} \end{cases} \quad (1.25)$$

Without loss of generality, we apply a backward Euler scheme in time. The generalization to other time discretization is straightforward.

At this stage, we can solve for the bubble coefficients; the latter are determined by:

$$\boxed{T_{b_K}^n = \frac{1}{\frac{\rho C_p}{\Delta t} \|\psi\|_{0,K}^2 + \lambda \|\nabla \psi\|_{0,K}^2} \left( f - \rho C_p \left( \frac{T_h^n - T_h^{n-1} - T_b^{n-1}}{\Delta t} \right), \psi \right)_K} \quad (1.26)$$

where  $T_b^{n-1}$  denotes the solution field on the bubble functions at the previous time increment.

The following step of the enriched space approach is the static condensation that consists in modelling the effects of the small scales on the large scales. For that purpose, we consider the coarse scale equation:

$$\left( \rho C_p \frac{\partial \tilde{T}_h}{\partial t}, w_h \right)_\Omega + (\lambda \nabla \tilde{T}_h, \nabla w_h)_\Omega = (f_h, w_h)_\Omega \quad \forall w_h \in \mathcal{W}_h^0 \quad (1.27)$$

and use  $\forall w_h \in \mathcal{W}_h^0$ , the integration by parts on  $(\nabla(T_{b_K} \psi), \nabla w_h)$ :

$$(\nabla(T_{b_K} \psi), \nabla w_h) = -((T_{b_K} \psi), \Delta w_h) + (\nabla(T_{b_K} \psi) \cdot \mathbf{n}, w_h)_{\partial K} \quad \forall K \in \mathcal{H}_h \quad (1.28)$$

As we are considering linear weighting functions, the first term on the equation's right hand side vanishes. The second term also vanishes by virtue of the bubble functions' definition. Therefore, using an implicit discretization in time,  $\forall w_h \in \mathcal{W}_h^0$  equation (1.27)

reduces to:

$$\left(\rho C_p \frac{T_h^n}{\Delta t}, w_h\right)_\Omega + \sum_{K \in \mathcal{H}_h} \left(\rho C_p \frac{T_{b_K}^n \psi}{\Delta t}, w_h\right)_K + (\lambda \nabla T_h^n, \nabla w_h)_\Omega = (f_h, w_h)_\Omega + \left(\frac{T_h^{n-1} + T_b^{n-1}}{\Delta t}, w_h\right) \quad (1.29)$$

Combining (1.26) and (1.29), implies on each element  $K$ :

$$T_{b_K} \left(\frac{\rho C_p}{\Delta t} \psi, w_h\right)_K = \frac{1}{\frac{\rho C_p}{\Delta t} \|\psi\|_{0,K}^2 + \lambda \|\nabla \psi\|_{0,K}^2} \left(f - \rho C_p \left(\frac{T_h^n - T_h^{n-1} - T_b^{n-1}}{\Delta t}, \psi\right)_K, \psi\right)_K \left(\frac{\rho C_p}{\Delta t} \psi, w_h\right)_K \quad (1.30)$$

Moreover, following the lines of the work on steady CDR problems in [Baiocchi 93, Brezzi 92, Franca 95], we consider cubic bubble functions. One gets under these assumptions:

$$\int_K \psi dK = C_1 h_K^2, \text{ and } \|\psi\|_{0,K} = C_2 h_K^2$$

Replacing in (1.30), yields:

$$T_{b_K} \left(\frac{\rho C_p}{\Delta t} \psi, w_h\right)_K = \underbrace{\frac{C h_K^2}{\frac{\rho C_p}{\Delta t} h_K^2 + \lambda \bar{C}}}_{\tau_K} \left(f - \rho C_p \left(\frac{T_h^n - T_h^{n-1} - T_b^{n-1}}{\Delta t}, \frac{\rho C_p}{\Delta t} w_h\right)_K\right) \quad (1.31)$$

where we take  $C = 1$  and  $\bar{C} = 6$  constants as in [Hachem 10b].

Finally, substituting back into the large scale problem, we obtain for all  $w_h \in \mathcal{V}_h^0$ :

$$\begin{aligned} & \left(\rho C_p \frac{T_h^n}{\Delta t}, w_h\right)_\Omega - \sum_{K \in \mathcal{H}_h} \tau_K \left(\rho C_p \left(\frac{T_h^n}{\Delta t}, \frac{\rho C_p}{\Delta t} w_h\right)_K + (\lambda \nabla T_h^n, \nabla w_h)_\Omega \right. \\ & = (f_h, w_h)_\Omega + \left(\frac{T_h^{n-1} + T_b^{n-1}}{\Delta t}, w_h\right)_\Omega - \sum_{K \in \mathcal{H}_h} \tau_K \left(f + \rho C_p \left(\frac{T_h^{n-1} + T_b^{n-1}}{\Delta t}, \frac{\rho C_p}{\Delta t} w_h\right)_K \right). \end{aligned} \quad (1.32)$$

In this formulation, two aspects of stabilization can be identified, the first is at the level of the transient term, and the second is on the right hand side of the equation taking into account the previous information on the solution.

This stabilization is equivalent to modifying the test function of the standard Galerkin formulation into:

$$\widetilde{w}_h = w_h \left(1 - \tau_K \frac{\rho C_p}{\Delta t}\right). \quad (1.33)$$

Hence, the modified problem can be interpreted as a modified Galerkin with larger

time-step size and can be written equivalently as:

$$\boxed{\begin{aligned} & \left( \rho C_p \frac{T_h^n}{\Delta t}, \widetilde{w}_h \left( 1 - \tau_k \frac{\rho C_p}{\Delta t} \right) \right)_{\Omega} + (\lambda \nabla T_h^n, \nabla \widetilde{w}_h)_{\Omega} \\ & = \left( f + \rho C_p \left( \frac{T_h^{n-1} + T_b^{n-1}}{\Delta t} \right), \widetilde{w}_h \left( 1 - \tau_k \frac{\rho C_p}{\Delta t} \right) \right)_{\Omega}. \end{aligned}} \quad (1.34)$$

### 1.2.11 Time interpolated enriched (EM-I) method

It was pointed out in [Hachem 10b] that if zero source term is assumed then the solution to problem (1.34) may not converge to the solution of problem (1.18a). A remedy to this situation was proposed in [Hachem 10b], it consists in interpolating the solution to real time. To do so, we define a new time-step  $\Delta t^*$  such that:

$$\frac{\Delta t}{\Delta t^*} = \left( 1 - \tau_k \frac{\rho C_p}{\Delta t} \right) = \frac{1}{\xi} \quad \xi > 0. \quad (1.35)$$

The corresponding real-time interpolation will read as:

$$T^* = \xi T_h^n + (1 - \xi) T_h^{n-1}. \quad (1.36)$$

Now substituting back into (1.34) and rearranging the terms yields:

$$\begin{aligned} & \left( \frac{\rho C_p}{\Delta t^*} \left( \xi T_h^n + (1 - \xi) T_h^{n-1} \right), \widetilde{w}_h \right) + \left( \xi \lambda \nabla \left( T_h^n + (1 - \xi) T_h^{n-1} \right), \nabla \widetilde{w}_h \right) \\ & = \left( \rho C_p \frac{T_h^{n-1} + T_b^{n-1}}{\Delta t^*}, \widetilde{w}_h \right), \end{aligned} \quad (1.37)$$

and equivalently since  $\Delta t = \frac{\Delta t^*}{\xi}$ ,

$$\left( \rho C_p \frac{T_h^n - T_h^{n-1} - \xi T_b^{n-1}}{\Delta t}, w_h \right) + (\xi \lambda \nabla T_h^n, \nabla w_h) = \left( (\xi - 1) \lambda \nabla \left( T_h^{n-1}, \nabla w_h \right) \right). \quad (1.38)$$

This modification can be interpreted as a time-dependent stabilization involving modified thermal conduction  $\xi \lambda$  defined at the elements' interior and controlled by a stabilization parameter  $\tau_k$ .

More details on the derivation of this stabilization technique for diffusion dominated problems and numerical validations are provided in [Hachem 10b].

## 1.3 THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

The motion of fluids is known to be embodied by the nonlinear Navier-Stokes equations. Although these equations were introduced in the 19th century and have a very

wide range of applications including aerodynamics, microfluids, circulatory and respiratory systems, household appliances, geophysical and astrophysical phenomena etc., the understanding, and theoretical validation of these equations is still at a primitive level. Their resolution consists in finding a velocity/pressure pair  $(\mathbf{v}, p)$  satisfying the problem setup. There are a few analytical solutions in the literature but these were computed on the basis of introducing model simplifications which are generally non-physical. Several attempts to solve these equations numerically were addressed in the literature. Accurate solutions with different scale heterogeneity are still a challenge and induce very stiff systems. Considerable advancements need to be achieved including spatial accuracy, temporal accuracy, error control and mesh adaptation.

Two non-dimensional numbers characterize a flow: the Mach number and the Reynolds number. The former, named after Ernst Mach (1836-1916), measures the ratio of a characteristic velocity  $\mathbf{v}$  of the flow to the velocity of sound  $c$  in the fluid:

$$\text{Ma} = \frac{\|\mathbf{v}\|}{c}. \quad (1.39)$$

On the other hand, the latter parameter describes the flow behavior by quantifying the ratio between inertial and viscous forces for a given flow:

$$\text{Re} = \frac{\rho \bar{\mathbf{V}} L}{\mu}, \quad (1.40)$$

with  $L$  being a characteristic length scale,  $\rho$  the fluid density,  $\bar{\mathbf{V}}$  its mean velocity and  $\mu$  its dynamic viscosity. Osborne Reynolds (1842-1912), the eponym of this dimensionless number, observed that the flow is said to be laminar when the Reynolds number is small and turbulent otherwise. In that case, the flow is characterized by chaotic fluctuations and develops eddies with different sizes. In the following section we aim at addressing ubiquitous turbulent behavior of flows and use a multiscale approach to capture it and turbulence models to represent it.

This section presents the mathematical formulation of the governing equations of fluid flows; in particular, we are interested in modelling incompressible flows (i.e. ignoring density changes). We set the problem in a finite element framework. However there are two cases where the standard Galerkin finite element scheme fails and the numerical solution is corrupted by nonphysical oscillations. The first is due to the nonlinearity of the convective term in convection dominated regimes and the second stems from the incompatibility of the finite element functional spaces of the velocity and pressure fields [[Ladyzhenskaya 63](#), [Temam 77](#), [Arnold 84](#), [Franca 01](#)]. This space compatibility condition is known as the inf-sup condition, also called the Ladyzhenskaya-Brezzi-Babuška condition. To deal with the instabilities arising from the space incompatibility, two possible remedies can be adopted: either we circumvent the condition by adding stabilization terms to the Galerkin formulation, or we enrich the functional space in order to satisfy it, i.e. we derive a stable formulation. The finite element

functional space combinations satisfying the Ladyzhenskaya-Brezzi-Babuška condition have been studied on the Stokes problem. A popular mixed finite element formulation, known as the Mini element method [Arnold 84], consists in using equal order spaces for both fields while enriching the velocity space with bubble functions. The latter vanishes at the elements' boundaries and hence can be eliminated and statically condensed yielding a stabilized formulation. The choice of optimal shape for the bubble functions affects the stabilization of the problem. It was pointed out in [Franca 01] that, unlike diffusion dominated regimes where the numerical solution is satisfactory, in convection dominated problems the method's performance is reduced. To solve this issue, Franca and Nesliturk [Franca 01], used the residual-free bubble (RFB) method with a natural way of choosing stabilization terms. Several works on the derivation of stabilization methods for the resolution of the Stokes and the Navier-Stokes problems can be found in the literature. First, the Unusual Stabilized finite element method was introduced by Franca and Farhat in [Franca 95]. Next, the residual based stabilization methods are well described by Brezzi *et al.* [Brezzi 96], Donea and Huerta [Donea 03] and Codina *et al.* [Codina 00a, Codina 00b, Codina 01, Codina 07, Badia 09]. Finally, multiscale approaches were proposed by Hughes *et al.* [Hughes 95, Hughes 98, Hughes 01] and Gravemeier [Gravemeier 06] and the SUPG/PSPG methods by Tezduyar [Tezduyar 92].

A finite element solver for the Stokes problem based on the Mini element method has been developed, implemented and validated in the CIMLIB library by Coupez, Perchat and Digonnet [Coupez 96, Perchat 00, Digonnet 03]. An extension of these works to transient Navier-Stokes equations at low Reynolds number is credited to Basset [Basset 06]. The Variational Multiscale method was brought forth for high Reynolds numbers, implemented and validated in CIMLIB by Hachem [Hachem 09]. The current work is a continuation upon these references and intends to couple the stabilized finite element solvers with anisotropic mesh adaptation and time adaptive algorithm in the view of optimizing the numerical solutions. A slight modification will be considered in chapter 3 to account for the synergy with adaptation tools.

In this chapter, we are interested in retaining the advantages, in terms of accuracy and computational costs, of using linear ( $P_1$ ) finite element approximations for both velocity and pressure fields. The Variational Multiscale approach will be used to stabilize the problem. It involves an orthogonal decomposition of the solution (velocity and pressure) spaces, circumvents the inf-sup condition and provides stabilization in the case of convection dominated regimes.

### 1.3.1 Governing equations

Let  $\Omega \subset \mathbb{R}^d$ ,  $d \in \{2, 3\}$  be a bounded domain and  $(0, T)$  be a time interval. The modelling of incompressible flows inside  $\Omega$  is embodied by the transient incompressible Navier-Stokes equations. It consists in finding the pair velocity/pressure  $\mathbf{v}(\mathbf{x}, t)$  and

$p(\mathbf{x}, t)$  such that:

$$\begin{aligned}\rho (\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) - \nabla \cdot \boldsymbol{\gamma} &= \mathbf{f} \quad \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{v} &= 0 \quad \text{in } \Omega \times (0, T)\end{aligned}\tag{1.41}$$

where  $\boldsymbol{\gamma}$  is the stress tensor,  $\rho$  the fluid density and  $\mathbf{f}$  a given vector of external forces acting on the flow. The first equation models the conservation of mass whereas the second, known as the continuity equation, reflects the incompressibility of the flow.

Let  $\mathbb{I}$  be the identity tensor. The stress tensor  $\boldsymbol{\gamma}$  associated with a Newtonian fluid i.e., for a fluid dynamic viscosity  $\mu$ , is defined by the fundamental constitutive law:

$$\boldsymbol{\gamma} = 2\mu\boldsymbol{\varepsilon}(\mathbf{v}) - p\mathbb{I}\tag{1.42}$$

where  $\boldsymbol{\varepsilon}(\mathbf{v})$  is the strain rate tensor, defining the symmetric part of the velocity gradient:

$$\boldsymbol{\varepsilon}(\mathbf{v}) = \frac{1}{2} \left[ \nabla \mathbf{v} + (\nabla \mathbf{v})^t \right]\tag{1.43}$$

Substituting (1.42) and (1.43) in (1.41), the following momentum equation results:

$$\begin{aligned}\rho (\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) - 2\mu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{v}) + \nabla p &= \mathbf{f} \quad \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{v} &= 0 \quad \text{in } \Omega \times (0, T)\end{aligned}\tag{1.44}$$

We close the system with appropriate initial and boundary conditions:

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \quad \text{in } \Omega\tag{1.45a}$$

$$\mathbf{v}(\cdot, t) = \mathbf{v}_D \quad \text{on } \Gamma_d \times (0, T)\tag{1.45b}$$

$$\mathbf{v} \cdot \mathbf{n} = h_N \quad \text{on } \Gamma_h \times (0, T)\tag{1.45c}$$

where  $\partial\Omega = \Gamma = \Gamma_d \cup \Gamma_h$  and  $\Gamma_d \cap \Gamma_h = \emptyset$ . We note that when prescribing Dirichlet boundary conditions on the domain boundary  $\partial\Omega$ , a condition on the pressure  $\int_{\Omega} p(\mathbf{x}) = 0$  shall be added.

### 1.3.2 Standard Galerkin formulation

To derive the variational formulation of problem (1.44-1.45), we use the following subspaces of the usual Lebesgue function space of square integrable functions, and Hilbert space defined on  $\Omega$ :

$$\begin{aligned}\mathcal{V} &= \left\{ \mathbf{v}, \mathbf{v} \in \left( H^1(\Omega) \right)^d \mid \mathbf{v} = \mathbf{v}_d \text{ on } \Gamma_D \right\}, \\ \mathcal{V}^0 &= \left\{ \mathbf{v}, \mathbf{v} \in \left( H^1(\Omega) \right)^d \mid \mathbf{v} = 0 \text{ on } \Gamma_D \right\}, \text{ and} \\ \mathcal{Q} &= \{ q, q \in L^2(\Omega) \}.\end{aligned}$$

We multiply the first and second equation of (1.44) by test functions  $(\mathbf{w}, q) \in (\mathcal{V}^0, \mathcal{Q})$ . Then we integrate by parts the viscous and pressure terms of the first equation. An additional term emerges and involves an integral over the boundary  $\Gamma_N$ . Therefore the weak form of equations (1.44) consists in finding  $(\mathbf{v}, p) \in (\mathcal{V}, \mathcal{Q})$  such that:

$$\begin{cases} \left( \rho \frac{\partial \mathbf{v}}{\partial t}, \mathbf{w} \right) + (\rho \mathbf{v} \cdot \nabla \mathbf{v}, \mathbf{w}) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{w})) - (p, \nabla \cdot \mathbf{w}) = (f, \mathbf{w}) + (h_N, \mathbf{w})_{\Gamma_N} & \forall \mathbf{w} \in \mathcal{V}^0 \\ (\nabla \cdot \mathbf{v}, q) = 0 & \forall q \in \mathcal{Q} \end{cases} \quad (1.46)$$

For spatial discretization, we consider the finite element partition  $\mathcal{H}_h$  of  $\Omega$  into simplex elements  $K$ . The above-defined functional spaces  $\mathcal{V}$  and  $\mathcal{V}^0$  are approached on this partition by discrete spaces spanned by piecewise linear functions:

$$\begin{aligned} \mathcal{V}_h &= \left\{ \mathbf{v}_h \in (C^0(\Omega))^d, \mathbf{v}_h|_K \in P^1(K)^d, \forall K \in \mathcal{H}_h \right\}, \\ \mathcal{V}_{h,0} &= \left\{ \mathbf{w}_h \in \mathcal{V}_h, \mathbf{w}_h|_{\Gamma} = 0 \right\}, \text{ and} \\ \mathcal{Q}_h &= \left\{ q_h \in C^0(\Omega), q_h|_K \in P^1(K), \forall K \in \mathcal{H} \right\} \end{aligned}$$

The Galerkin discrete problem associated with the finite element approximation involves solving the following mixed problem:

Find the pair  $(\mathbf{v}_h, p_h) \in (\mathcal{V}_h, \mathcal{Q}_h)$  and  $(\mathbf{w}_h, q_h) \in (\mathcal{V}_{h,0}, \mathcal{Q}_h)$  such that:

$$\boxed{\begin{aligned} \left( \rho \frac{\partial \mathbf{v}_h}{\partial t}, \mathbf{w}_h \right) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)) - (p, \nabla \cdot \mathbf{w}_h) \\ = (f, \mathbf{w}_h) + (h_N, \mathbf{w}_h)_{\Gamma_N} & \quad \forall \mathbf{w}_h \in \mathcal{V}_{h,0} \\ (\nabla \cdot \mathbf{v}_h, q_h) = 0 & \quad \forall q_h \in \mathcal{Q}_h \end{aligned}} \quad (1.47)$$

To obtain stable finite element solutions of the Navier-Stokes problem, the velocity and pressure fields must satisfy the inf-sup condition given by:

$$\inf_{\mathbf{q}_h \in \mathcal{Q}_h} \sup_{\mathbf{v}_h \in \mathcal{V}_{h,0}} \frac{(\nabla \cdot \mathbf{v}_h, \mathbf{q}_h)_{\Omega}}{|\mathbf{q}_h|_0 |\mathbf{v}_h|_1} \geq \beta > 0 \quad (1.48)$$

where  $\beta$  is a constant independent of the mesh size  $h$ .

To illustrate on this point, we consider the Stokes problem that can be obtained by omitting the inertial forces and dropping the time variation. The weak formulation of this problem reads as:

$$\begin{cases} (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h)) - (p, \nabla \cdot \mathbf{w}_h) = (f, \mathbf{w}_h) & \forall \mathbf{w}_h \in \mathcal{V}_{h,0} \\ (\nabla \cdot \mathbf{v}_h, q_h) = 0 & \forall q_h \in \mathcal{Q}_{h,0}. \end{cases} \quad (1.49)$$

Let  $(\mathbf{v}_h, p_h) \in (\mathcal{V}_h, \mathcal{Q}_h)$  be a solution couple to the discrete Stokes problem. If condition (1.48) is not satisfied, then the pressure solution is not unique i.e. we can construct  $\tilde{q}_h \in \mathcal{Q}_h$  with  $\int_{\Omega} \tilde{q}_h dx = 0$  such that  $(\nabla \cdot \mathbf{v}_h, \tilde{q}_h) = 0, \forall \mathbf{v}_h \in \mathcal{V}_h$  and consequently, the pair  $(\mathbf{v}_h, p_h + \tilde{q}_h) \in (\mathcal{V}_h, \mathcal{Q}_h)$  is also a solution of problem (1.49). This example is known as a ‘spurious pressure mode’.

The Mini element method was proposed as a remedy to this failure [Arnold 84]. The idea behind it is to enrich the velocity piecewise linear functional space with a space of bubble functions defined at the elements’ centroids and vanishing on element boundaries. A static condensation is then performed resulting in a stable solution couple. We omit the development of this procedure in this work for brevity and direct the interested reader to [Basset 06, Hachem 09].

A three-level model is proposed in [Calderer 13] whereby both large and fine scales are stabilized. A nested scale resolution requiring successive VMS steps is favored.

In this thesis, we adopt the Variational Multiscale stabilization for curing numerical instabilities arising in convection dominated regimes and solution functional spaces incompatibility.

### 1.3.3 Variational Multiscale (VMS) stabilization method

Following the lines of Hughes et al. [Hughes 95, Hughes 98, Hughes 01], we apply an orthogonal decomposition of the functional spaces  $\tilde{\mathcal{V}}$  and  $\tilde{\mathcal{Q}}$  into:

$$\begin{aligned} \tilde{\mathcal{V}} &= \mathcal{V}_h \oplus \mathcal{V}' \\ \text{and } \tilde{\mathcal{Q}} &= \mathcal{Q}_h \oplus \mathcal{Q}' \end{aligned} \quad (1.50)$$

By taking  $\tilde{\mathbf{v}} = \mathbf{v}_h + \mathbf{v}' \in \tilde{\mathcal{V}}$  and  $\tilde{p} = p_h + p' \in \tilde{\mathcal{Q}}$ , we decompose respectively the velocity and pressure fields into resolvable coarse and unresolvable fine scales. In particular the fine-scales will provide additional stabilization at locations of steep velocity gradient. The same decomposition applies to the test functions  $\tilde{\mathbf{w}} = \mathbf{w}_h + \mathbf{w}'$  and  $\tilde{q} = q_h + q'$ .

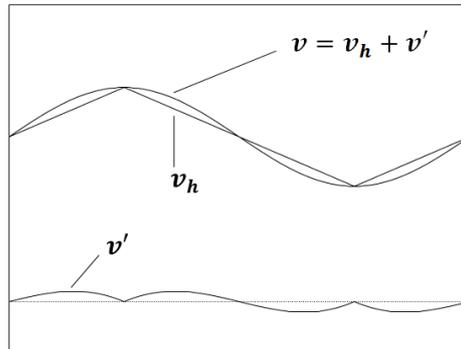


Figure 1.4 – VMS approach solution decomposition adopted from [Hughes 98]

We assume that the fine-scale velocity and pressure are represented by piecewise polynomials continuous in space but discontinuous/piecewise constant in time.

Bearing these sum decompositions in mind, the mixed finite element formulation of problem (1.44) becomes:

Find  $(\tilde{\mathbf{v}}, \tilde{p}) \in (\tilde{\mathcal{V}}, \tilde{\mathcal{Q}})$  such that:

$$\begin{aligned}
 & \left( \rho \frac{\partial(\mathbf{v}_h + \mathbf{v}')}{\partial t}, \mathbf{w}_h + \mathbf{w}' \right) + (\rho (\mathbf{v}_h + \mathbf{v}') \cdot \nabla (\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h + \mathbf{w}') \\
 & + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}_h + \mathbf{w}')) - ((p_h + p'), \nabla \cdot (\mathbf{w}_h + \mathbf{w}')) \\
 & = (f, \mathbf{w}_h + \mathbf{w}') + (h_N, \mathbf{w}_h + \mathbf{w}')_{\Gamma_N} \quad \forall \mathbf{w}_h + \mathbf{w}' \in \mathcal{V}_{h,0} \oplus \mathcal{V}'_0 \\
 & (\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q_h + q') = 0 \quad \forall q_h + q' \in \mathcal{Q}_{h,0} \oplus \mathcal{Q}'_0
 \end{aligned} \tag{1.51}$$

To maintain the consistency of the approach, the function spaces of the fine-scales and large-scales should be linearly independent [Masud 04]. Although the variational formulation (1.51) is nonlinear in the convection term, it is linear with respect to the test functions. Hence the problem can be uncoupled at the element level and the fine-scales vanish on the element boundaries. Subsequently, the weighting functions' splitting yields the following two subproblems:

- The coarse-scale problem:

$$\begin{aligned}
 & \left( \rho \frac{\partial(\mathbf{v}_h + \mathbf{v}')}{\partial t}, \mathbf{w}_h \right) + (\rho (\mathbf{v}_h + \mathbf{v}') \cdot \nabla (\mathbf{v}_h + \mathbf{v}'), \mathbf{w}_h) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}_h)) \\
 & - ((p_h + p'), \nabla \cdot \mathbf{w}_h) = (f, \mathbf{w}_h) + (h_N, \mathbf{w}_h)_{\Gamma_N} \quad \forall \mathbf{w}_h \in \mathcal{V}_{h,0} \\
 & (\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q_h) = 0 \quad \forall q_h \in \mathcal{Q}_{h,0}
 \end{aligned} \tag{1.52}$$

- The fine-scale problem:

$$\begin{aligned}
 & \left( \rho \frac{\partial(\mathbf{v}_h + \mathbf{v}')}{\partial t}, \mathbf{w}' \right) + (\rho (\mathbf{v}_h + \mathbf{v}') \cdot \nabla (\mathbf{v}_h + \mathbf{v}'), \mathbf{w}') + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h + \mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}')) \\
 & - ((p_h + p'), \nabla \cdot \mathbf{w}') = (f, \mathbf{w}') + (h_N, \mathbf{w}')_{\Gamma_N} \quad \forall \mathbf{w}' \in \mathcal{V}'_0 \\
 & (\nabla \cdot (\mathbf{v}_h + \mathbf{v}'), q') = 0 \quad \forall q' \in \mathcal{Q}'_0
 \end{aligned} \tag{1.53}$$

As an orthogonal decomposition was used, the crossed viscous terms  $(\boldsymbol{\varepsilon}(\mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}_h))$  in equation (1.52) and  $(\boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}'))$  in equation (1.53) vanish.

At this stage, we consider the small-scale problem at the elements' level. A static condensation will then be applied whereby the fine-scale solution will be embedded back onto the large-scale problem yielding a local time dependent stabilization term that models the effects of the small-scales on the large-scales without explicitly resolving them. Thereby the accuracy and stability of the mixed formulation will be improved.

We assume zero Dirichlet boundary condition for ease of elucidation, the extension to other types of boundary conditions is straightforward. As the bubble functions are locally defined at the elements' interiors, the small-scale problem can be re-written as:

$$\left\{ \begin{array}{l} \underbrace{(\rho \partial_t \mathbf{v}', \mathbf{w}')_K}_{\text{time dependent subscale}} + \underbrace{(\rho (\mathbf{v}_h + \mathbf{v}') \cdot \nabla \mathbf{v}', \mathbf{w}')_K}_{\text{non-linear convection term}} + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}'))_K + (\nabla p', \mathbf{w}')_K \\ = (f - \rho \partial_t \mathbf{v}_h - \rho (\mathbf{v}_h + \mathbf{v}') \cdot \nabla \mathbf{v}_h - \nabla p_h, \mathbf{w}')_K \quad \forall \mathbf{w}' \in \mathcal{V}'_0, \forall K \in \mathcal{H}_h \\ = (\mathcal{R}_m, \mathbf{w}')_K \quad \forall \mathbf{w}' \in \mathcal{V}'_0 \\ (\nabla \cdot \mathbf{v}', q')_K = (-\nabla \cdot \mathbf{v}_h, q')_K = (\mathcal{R}_c, q')_K \quad \forall q' \in \mathcal{Q}'_0, \forall K \in \mathcal{H}_h \end{array} \right. \quad (1.54)$$

where  $(\mathcal{R}_m, \mathbf{w}')$  and  $(\mathcal{R}_c, q')$  are respectively the residuals of the momentum and continuity large-scale equations projected onto the fine-scale spaces.

We recall that our objective is to approximate the fine-scale solutions. To do so, we introduce simplifying assumptions on the time-dependency and the non-linear terms in equations (1.54):

- i) Tracking the time evolution of the small-scales necessitates considerable memory resources and induces an important computational cost. Therefore to maintain practical and affordable numerical simulations, especially when modelling complex 3D problems, we drop the contribution of terms involving past fine-scale solutions (see [Dubois 99] for a justification of this choice). However, it is worth mentioning that the effects of the time history are still modelled via the residuals of the coarse-scales that account for their time derivatives. Interested reader about accounting for small-scale dynamics can consult [Codina 07].
- ii) As the large-scale gradients are dominant with respect to small-scales, the convective velocity can be limited to its large-scale part as the latter is important for turbulence modelling [Bazilevs 07, Calderer 13]:

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

Note that these terms are useful for evaluating turbulent fluctuations. In the following section, we will resort to turbulence models (such as the LES,  $k - \varepsilon$ , and Spalart Allmaras models) to deal with turbulent flows.

Respecting these assumptions, equations (1.54) become:

$$\left\{ \begin{array}{l} (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}', \mathbf{w}')_K + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}') : \boldsymbol{\varepsilon}(\mathbf{w}'))_K + (\nabla p', \mathbf{w}')_K = (\mathcal{R}_m, \mathbf{w}')_K \quad \forall \mathbf{w}' \in \mathcal{V}'_0, \forall K \in \mathcal{H}_h \\ (\nabla \cdot \mathbf{v}', q')_K = (\mathcal{R}_c, q')_K \quad \forall q' \in \mathcal{Q}'_0, \forall K \in \mathcal{H}_h \end{array} \right. \quad (1.55)$$

Taking into account the small-scale pressure ensures the continuity of the small-scales and leads to additional stabilization for high Reynolds flows as discussed in [Wall 00, Tezduyar 00]. Nevertheless, it is hard to solve for both small-scale velocity and pressure. As proposed in [Wall 00], we replace the small-scale continuity equation by a modified (consistent) Pressure Poisson equation. The effects of small-scales pressure Poisson's equation will be accounted for in the coarse-scale problem by the addition of the following estimated stabilization term that we adopt from[Codina 00a]:

$$p' \approx \tau_c \mathcal{R}_c, \quad \text{where} \quad \tau_c = \left( \left( \frac{\mu}{\rho} \right)^2 + \left( \frac{c_2}{c_1} \frac{\|\mathbf{v}\|_K}{h_K} \right)^2 \right)^{1/2} \quad (1.56)$$

and  $c_1$  and  $c_2$  being two constants independent from the characteristic element length  $h_K$ .

We focus our attention now on the fine-scale momentum equation. It was demonstrated in [Codina 00a] that setting  $p' = 0$  in the small-scale momentum equation is a reasonable choice to verify the inf-sup condition as it yields a larger function space for the velocity field. This assumption stems from the fact that the fine-scale velocity is driven by the residual of the large-scale momentum equation and not that of the continuity equation as pointed out in [Wall 00].

We use the fine-scale expansion proposed by Masud and Khurram [Masud 04]:

$$\mathbf{v}' = \sum_{K \in \mathcal{H}_h} \mathbf{v}'_K b_K \quad \text{and} \quad \mathbf{w}' = \sum_{K \in \mathcal{H}_h} \mathbf{w}'_K b_K, \quad (1.57)$$

where  $b_K$  are bubble shape functions,  $\mathbf{v}'_K$  denotes the fine-scale velocity on element  $K$  and  $\mathbf{w}'_K$  the corresponding weighting function.

Substituting  $\mathbf{v}'$  and  $\mathbf{w}'$  by their expansions and taking into account that bubble functions are zero on the boundaries of the elements we obtain:

$$(\rho \mathbf{v}_h \cdot \nabla b_K \mathbf{v}'_K, b_K \mathbf{w}'_K)_K + (2\mu \boldsymbol{\varepsilon}(b_K \mathbf{v}'_K) : \boldsymbol{\varepsilon}(b_K \mathbf{w}'_K))_K = (\mathcal{R}_m, b_K \mathbf{w}'_K)_K \quad \forall K \in \mathcal{H}_h \quad (1.58)$$

As this equation is verified for any choice of  $\mathbf{w}'_K$ , then it is in particular valid for  $\mathbf{w}'_K = 1$ , consequently, the fine-scales velocity coefficients can be deduced:

$$\mathbf{v}'_K = \frac{1}{\underbrace{(\rho \mathbf{v}_h \cdot \nabla b_K, b_K)_K + (2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K}_{\tau_K}} \cdot (\mathcal{R}_m, b_K)_K \quad \forall K \in \mathcal{H}_h \quad (1.59)$$

However, the convective term in the formula of the stabilization parameter  $\tau_K$  cancels out when using the same bubble functions for the velocity and test functions. To promote the influence of the convective term, we use a combination of standard and

upwind bubbles for the weighting functions and standard bubbles for the velocity solution as suggested in [Masud 04, Hachem 09]:

$$\mathbf{w}'|_K = \mathbf{w}'_K b_K^* = \mathbf{w}'_K (b_K + b_K^v) \quad (1.60)$$

The resulting stabilization parameter reads as:

$$\tau_K = \frac{b_k \int_K b_k^* d\Omega}{(\rho \mathbf{u}_h \cdot \nabla b_K, b_K^v)_K + (2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K} \quad \forall K \in \mathcal{H}_h \quad (1.61)$$

In here, the upwind bubble functions vanish in the viscous term as we are using linear elements. Clearly different bubble functions yield different stabilization parameters.

We are now ready for a static condensation whereby the small-scale effects are to be integrated onto the coarse-scale equations. Taking into account the quasi-static assumption on the fine-scales and applying integration by parts to the nonlinear term in the large-scale momentum equation and to the second term of the continuity equation (1.52) imply:

$$\left\{ \begin{array}{l} (\rho \partial_t \mathbf{v}_h, \mathbf{w}_h)_\Omega + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h)_\Omega - \sum_{K \in \mathcal{H}_h} (\tau_K \mathcal{R}_m, \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h)_K + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h))_\Omega \\ \quad - (p_h, \nabla \cdot \mathbf{w}_h)_\Omega + \sum_{K \in \mathcal{H}_h} (\tau_c \mathcal{R}_c, \nabla \cdot \mathbf{w}_h)_K = (\mathbf{f}, \mathbf{w}_h)_\Omega \quad \forall \mathbf{w}_h \in \mathcal{V}_{h,0} \\ (\nabla \cdot \mathbf{v}_h, q_h)_\Omega - \sum_{K \in \mathcal{H}_h} (\tau_K \mathcal{R}_m, \nabla q_h)_K = 0 \quad \forall q_h \in \mathcal{Q}_{h,0} \end{array} \right. \quad (1.62)$$

Substituting the residuals by their explicit formulas onto the large-scale problem we get:

$$\begin{aligned} & \underbrace{(\rho \partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \mathbf{w}_h)_\Omega + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w}_h))_\Omega - (p_h, \nabla \cdot \mathbf{w}_h)_\Omega + (\nabla \cdot \mathbf{v}_h, q_h)_\Omega - (\mathbf{f}, \mathbf{w}_h)_\Omega}_{\text{Galerkin terms}} \\ & + \underbrace{\sum_{K \in \mathcal{H}_h} \tau_K (\rho (\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla \mathbf{v}_h) + \nabla p_h - \mathbf{f}, \rho \mathbf{v}_h \cdot \nabla \mathbf{w}_h)_K}_{\text{Upwind stabilization terms}} \\ & + \underbrace{\sum_{K \in \mathcal{H}_h} \tau_K (\rho (\partial_t \mathbf{v}_h + \mathbf{v}_h \cdot \nabla \mathbf{v}_h) + \nabla p_h - \mathbf{f}, \nabla q_h)_K}_{\text{Pressure stabilization terms}} \\ & + \underbrace{\sum_{K \in \mathcal{H}_h} (\tau_c \nabla \cdot \mathbf{v}_h, \nabla \cdot \mathbf{w}_h)_K = 0}_{\text{grad-div stabilization term}} \quad \forall \mathbf{w}_h \in V_{h,0}, \quad \forall q_h \in Q_h \end{aligned}$$

This formulation can be dissected into 4 main parts: the first is the standard Galerkin formulation, the second and third are fine-scale velocity ingredients and the last part models the effects of small-scale pressure. All the added terms provide additional

stabilization in convection dominated regimes and permit to circumvent the inf-sup condition. Being derived from residual components, the method is established in a consistent way.

### 1.3.4 Choice of the stabilizing parameter

Several definitions are given in the literature for the stabilization parameters. In CIM-LIB, the following formula proposed in [Codina 00a, Tezduyar 00, Hachem 10b] is implemented:

$$\tau_k = \left[ \left( \frac{2\rho}{\Delta t} \right)^2 + \left( \frac{2\rho \|\mathbf{v}\|_{\mathbf{K}}}{h_K} \right)^2 + \left( \frac{4\nu}{m_K h_K^2} \right)^2 \right]^{-1/2}, \tau_c = \frac{h_K \|\mathbf{v}\|_{\mathbf{K}}}{2} \min(1, Re_K) \quad (1.63)$$

where  $\nu$  is the kinematic viscosity and  $Re_K$  denotes the local Reynolds number given by  $Re_K = \frac{\rho \|\mathbf{v}\|_{\mathbf{K}} h_K}{2\mu}$  and  $m_K$  is a constant independent from  $h_K$  [Codina 00a]. In these definitions  $h_K$  refers to the characteristic length of the element, usually it is set to the element's diameter. However, as it will be shown in chapter 2, this choice is not the optimal one when using anisotropic mesh adaptation.

## 1.4 TURBULENCE MODELLING

The understanding of turbulence is nowadays considered as the main intriguing and frustrating difficulty in fluid dynamics. It was recognized by da Vinci ( $\sim 1500$ ) as a distinguished physical behavior, “*turbolenza*”, a phenomenon in which “*the smallest eddies are almost numberless, and large things are rotated only by large eddies and not by small ones, and small things are turned by small and large eddies*”. Reynolds ( $\sim 1880$ ), through conducting an experimental study on a flow past a pipe with smooth transparent walls, observed the transition from laminar to turbulent flows. He introduced the adimensional number characterizing the flow's behavior. Researchers have been investigating a proper definition for this phenomenon. Richardson in 1922 described turbulence using these words: “*Big whorls have little whorls, which feed on their velocity; And little whorls have lesser whorls, And so on to viscosity*”. He referred to the energy cascade exhibiting the breaking down of eddies into smaller ones accompanied with a kinetic energy transfer. A turbulent flow is characterized by restless chaotic fluctuations of velocity and pressure that occur at high Reynolds number. As the Reynolds number increases, bifurcations arise and the flow develops, more frequently, an irregular turbulent behavior in both space and time.

Analytical solutions to turbulent flows are to this day deficient, all the existing efforts culminate in the numerical investigations. A straightforward option would be to solve the Navier-Stokes equations with appropriate initial and boundary conditions. This way is known as the Direct Numerical Simulation (DNS). However the broad range of space and time scales to be resolved using the NS equations is very large

and beyond the realms of present computational powers. Hence, DNS method is still limited to a moderate Reynolds number. As sketched in figure 1.5, two potential redeemers are the so-called Large Eddy Simulation (LES) and Reynolds Averaged Navier-Stokes (RANS). The former of these approaches consists in filtering the flow scales, fully resolving the large-scale structure while only modelling the effect of the small-scales. However this method does not provide the ultimate cure, its computational cost is still considerable. RANS on the other hand, averages the equations in space and time completely removing the fluctuations and yielding computationally less expensive set of equations. However as the RANS method does not solve the small scales, closure models are required to account for turbulence effect. Of these models, we are interested in considering the  $k - \varepsilon$  and the Spalart Allmaras ones.

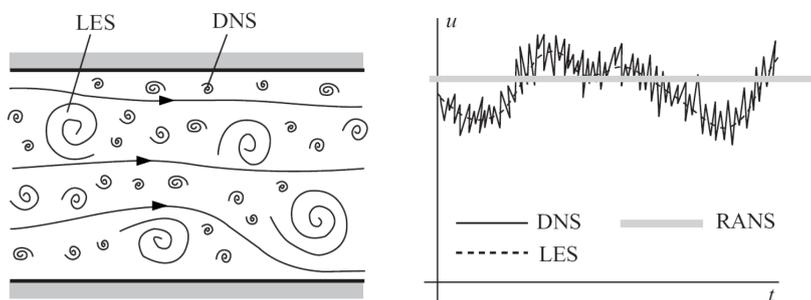


Figure 1.5 – Schematic of turbulent flow scales (left) and the three modelling approaches (right), adopted from [Ferziger 96, Hachem 09]

### 1.4.1 Direct Numerical Simulation (DNS)

The direct numerical simulation method relies exclusively on the standard Galerkin formulation of the Navier-Stokes problem. For a fine enough mesh (the mesh size  $h \rightarrow 0$ ) the DNS method is capable of resolving all the solution's scales. However a very wide range of space and time scales emerges in turbulent flows. According to the Kolmogorov theory [Kolmogorov 91], a measure of the needed computational power is quantified using the ratio of the largest (L) to the smallest (s) dissipative scale. So if we consider a computational domain with dimensions equal to the largest eddy, the number of mesh nodes required for capturing a 3D turbulent flow is of the order

$$N = \frac{L}{s} \text{Re}^{\frac{9}{4}}$$

For a time-step size having the same order as the characteristic element length, a total number  $\text{Re}^3$  degrees of freedom in space and time is needed. Consequently, the current available computational power, even with the existing supercomputers, are not capable of simulating high Reynolds number flows [Moin 98].

### 1.4.2 Reynolds Averaged Navier-Stokes (RANS)

When analyzing a turbulent flow, there are two main variables of interest: the velocity and the pressure. These variables present fluctuations in both space and time. Denote by  $\varphi = \varphi(\mathbf{x}, t)$  the generic variable of interest. In order to separate the time average  $\bar{\varphi}$  from the small scale fluctuations  $\varphi'$ , we apply the following decomposition [Gaston 97]:

$$\varphi(\mathbf{x}, t) = \bar{\varphi}(\mathbf{x}, t) + \varphi'(\mathbf{x}, t), \quad (1.64)$$

where the time average is computed as:

$$\bar{\varphi}(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_0^T \varphi(\mathbf{x}, t) dt \quad \text{and} \quad \overline{\varphi'} = 0. \quad (1.65)$$

A discrete version of this entity is known as the “ensemble average” and is defined by:

$$\bar{\varphi}(\mathbf{x}) = \lim_{T \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \varphi^n(\mathbf{x}, t) dt, \quad (1.66)$$

where  $\varphi^n(\mathbf{x}, t)$  denotes the variable at the  $n^{\text{th}}$  time instant. In addition, we have the two important ingredients of the averaging (filtering) operator:

$$\overline{\varphi + \psi} = \bar{\varphi} + \bar{\psi} \quad \text{and} \quad \overline{\varphi \cdot \psi} = \bar{\varphi} \cdot \bar{\psi}$$

From this property we infer that the large (resolvable) scale and the small (unresolvable) scale do not interact:

$$\overline{\varphi \cdot \psi} = \overline{(\bar{\varphi} + \varphi') \cdot (\bar{\psi} + \psi')} = \bar{\varphi} \cdot \bar{\psi} + \overline{\varphi' \cdot \psi'}. \quad (1.67)$$

When the two quantities are uncorrelated, the last term in equation (1.67) vanishes.

Now that we have defined the filtering operator, we plug in the averaged velocity and pressure into the Navier-Stokes problem and obtain:

$$\begin{aligned} \rho (\partial_t \bar{\mathbf{v}} + \bar{\mathbf{v}} \cdot \nabla \bar{\mathbf{v}}) - \nabla \cdot (\mu \nabla \bar{\mathbf{v}} + \nabla \bar{\mathbf{v}}^t) + \nabla \bar{p} + \nabla \cdot (\overline{\rho \mathbf{v}' \otimes \mathbf{v}'}) &= f \quad \text{in } \Omega \times (0, T) \\ \nabla \cdot \bar{\mathbf{v}} &= 0 \quad \text{in } \Omega \times (0, T) \end{aligned} \quad (1.68)$$

We can detect the presence of a new tensor that takes into account turbulent motion and represents the influence of the small scale fluctuations on the mean flow. This tensor is called Reynolds stresses or turbulent stresses tensor and is usually denoted by  $\mathcal{R}$ . Several models were proposed in the literature to approximate this term. At this level, a turbulence model is needed to close the set of equations; we resort to the Boussinesq

hypothesis that relates a turbulent or eddy viscosity to the mean shear stress:

$$\mathcal{R} = \mu_T (\nabla \bar{\mathbf{v}} + \nabla \bar{\mathbf{v}}^t) - \frac{2}{3} \rho k \mathbb{I} \quad (1.69)$$

under the condition of isotropic and small scale eddies. Inhere,  $\mu_T$  denotes the eddy viscosity,  $k = \frac{1}{2} \overline{\mathbf{v}' \otimes \mathbf{v}'}$  the averaged turbulent kinetic energy and  $\mathbb{I}$  the identity tensor. For more details, interested reader is referred to [Schmitt 07]. Note that the eddy viscosity can be interpreted as the product of the fluid density and squared velocity of fluctuations. Under the Boussinesq approximation, we obtain the following Reynolds averaged Navier-Stokes momentum equation:

$$\rho (\partial_t \bar{\mathbf{v}} + \bar{\mathbf{v}} \cdot \nabla \bar{\mathbf{v}}) + \nabla \left( \bar{p} + \frac{2}{3} \rho k \right) + \nabla \cdot ((\mu + \mu_t) (\nabla \bar{\mathbf{v}} + \nabla \bar{\mathbf{v}}^t)) = f \quad (1.70)$$

The solution of the RANS equations is now waded to finding the appropriate evaluation of the eddy viscosity and the turbulent kinetic energy. Various models in the literature are devoted to the definition of the eddy viscosity, and are classified according to the number of additional transport equations to be solved together with the RANS equation. Among these models we cite:

- Zero-equation models (e.g. the Prandtl mixing length model or the Baldwin-Lomax),
- One-equation models (e.g. the Spalart-Allmaras and Prandtl and Kolmogorov  $\sqrt{k}$  models),
- Two-equation models (e.g. the  $k - \varepsilon$  and the  $k - \omega$  models).

It is beyond the scope of this thesis to provide a detailed analysis on the different models present in the literature. We focus our attention on: the  $k - \varepsilon$  and the Spalart-Allmaras models.

#### 1.4.2.1 $k - \varepsilon$ turbulence model

First proposed in 1972 [Jones 72], the  $k - \varepsilon$  model has gained popularity in designing CFD libraries for industrial numerical simulations. The robustness and reliability of the model were demonstrated by Rodi in [Rodi 80]. It was the first model describing at the same time, the creation, the transport and the dissipation of the turbulent energy as well as the length scale. This scale is equivalent to the size of a small eddy. The model can be split into two parts that together present a dynamic system: an equation on the turbulent energy contained in the eddies, accounting for a running balance of energy generation and dissipation. The second equation represents the energy dissipation. The turbulent viscosity reflects the effect of fluctuating unresolved velocity and is computed by:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon}, \quad (1.71)$$

where  $C_\mu$  denotes an empirical constant usually equal to 0.09. To evaluate  $\mu_t$ ,  $k$  and  $\varepsilon$ , we solve a system of two transport equations [Van Maele 06] given by:

$$\rho (\partial_t k + \bar{\mathbf{v}} \cdot \nabla k) - \nabla \cdot \left( \left( \mu + \frac{\mu_t}{\text{Pr}_k} \right) \nabla k \right) = \mu_t (P_k + P_b) - \rho \varepsilon \quad \text{in } \Omega \quad (1.72)$$

$$\rho (\partial_t \varepsilon + \bar{\mathbf{v}} \cdot \nabla \varepsilon) - \nabla \cdot \left( \left( \mu + \frac{\mu_t}{\text{Pr}_\varepsilon} \right) \nabla \varepsilon \right) = \frac{\varepsilon}{k} (C_{1\varepsilon} P_k + C_{1\varepsilon} (1 - C_{3\varepsilon}) P_b - C_{2\varepsilon} \rho \varepsilon) \quad \text{in } \Omega \quad (1.73)$$

Inhere,  $P_k$  stands for the production of turbulent kinetic energy due to the mean velocity gradients,  $P_b$  represents the production due to the buoyancy effects,  $\text{Pr}_k$  and  $\text{Pr}_\varepsilon$  denote respectively the turbulent Prandtl number for  $k$  and  $\varepsilon$ . They are given by:

$$P_k = 2\mu_t (\boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{v})) \quad \text{and} \quad P_b = -\frac{\mu_t}{\rho \text{Pr}_t} \mathbf{g} \nabla \rho \quad (1.74)$$

with  $\text{Pr}_t$  being the turbulent Prandtl number that we set to 0.85. In addition,  $C_{1\varepsilon}$ ,  $C_{2\varepsilon}$  and  $C_{3\varepsilon}$  are model constants that we take, as suggested in [Van Maele 06]:

$$C_{1\varepsilon} = 1.44, \quad C_{2\varepsilon} = 1.92, \quad \text{and} \quad C_{3\varepsilon} = 0.88. \quad (1.75)$$

Finally, we determine the real pressure field from the effective pressure and the turbulent kinetic energy according to:

$$p = \bar{p} - \frac{2}{3} \rho k. \quad (1.76)$$

To facilitate the numerical resolution of the model, enhance its convergence, stability and robustness we apply the following modifications (suggested in [Hachem 09]):

- (i) We use a Newton Raphson linearization.
- (ii) We transform the destruction term into a reaction term in the turbulent kinetic energy equation to improve the stability:

$$\rho \varepsilon = \rho \frac{\varepsilon^i}{k^i} k \quad (1.77)$$

- (iii) We transform the destruction term in the dissipation equation as follows:

$$C_{2\varepsilon} \rho \frac{\varepsilon}{k} = 2C_{2\varepsilon} \rho \frac{\varepsilon^i}{k^i} \varepsilon - C_{2\varepsilon} \rho \frac{(\varepsilon^i)^2}{k^i} \quad (1.78)$$

where  $k^i$  and  $\varepsilon^i$  denote the turbulent kinetic energy and dissipation at iteration  $i$ .

We apply the Streamline Upwind Petrov Galerkin method to solve the  $k - \varepsilon$  model which has the form of a transient convection diffusion reaction equation. The variational multiscale approach is in turn used to solve the RANS problem.

### 1.4.2.2 Wall treatment

The  $k - \varepsilon$  model is active only in regions far from walls, thus special treatment needs to be done in the boundary layer. Experimental studies have shown that near wall flows exhibit a multilayered structure. Close to the wall, a “viscous sublayer” can be identified where the flow behaves in a laminar manner. Far from the wall, a “fully turbulent sublayer” is created where the flow is actively turbulent. In between, we have a “buffer sublayer” characterized by equal importance of the viscous and turbulent properties of the flow.

Two possible treatments can be adopted. The first one consists in employing wall functions in the buffer and viscous sublayers. The second approach involves modifying the turbulence model to enable their resolution up to the wall. In this regard, we follow the work of Grotjans and Menter (1998) [Grotjans 98] and use logarithmic wall functions so that to impose Neumann boundary conditions for the turbulent kinetic energy. This method has been well studied, implemented and validated in [Hachem 09]. Inhere for brevity of exposure, we retain only its main features.

We define a bridging value between the viscous and inertial sub-layers as the solution of the nonlinear equation [Thangam 91]:

$$y_c^+ = \frac{1}{\kappa} \ln (E y^+) , \quad (1.79)$$

where  $\kappa = 0.41$  is the Von Kármán constant, and  $E = 9.81$  is an empirical constant. Hence the solution of this nonlinear equation is given by  $y_c^+ = 11.06$ . For this value of  $y_c^+$ , we can compute the friction velocity and the wall shear stress using:

$$\mathbf{v}_\tau = \frac{|\mathbf{v}|}{y_c^+} \quad \text{and} \quad \tau_\omega = -\rho \mathbf{v}_\tau \mathbf{v}_k \frac{\mathbf{v}}{|\mathbf{v}|} \quad \text{on } \Gamma_\delta , \quad (1.80)$$

with  $\Gamma_\delta$  being the zone of the computational domain located at a distance  $\delta$  from the walls and  $\mathbf{v}_k = C_\mu^{\frac{1}{4}} \sqrt{k}$  is the bulk velocity. Note that the distance  $\delta$  is prescribed by the user. The velocity profile is then defined by:

$$\mathbf{v}^+ = \begin{cases} y^+ & \text{for } y^+ < y_c^+ \\ \frac{1}{\kappa} \ln (E y^+) & \text{for } y^+ > y_c^+ \end{cases} \quad (1.81)$$

A Neumann boundary condition is associated with the turbulent kinetic energy variable:

$$\frac{\partial k}{\partial \mathbf{n}} = 0 , \quad (1.82)$$

where  $\mathbf{n}$  is the normal to the wall. Finally, to complete the asymptotic behavior of all the variables, we set the dissipation rate in the near wall region to

$$\varepsilon_\omega = \frac{u_k^3}{\kappa \delta} \quad \text{on } \Gamma_\delta . \quad (1.83)$$

### 1.4.2.3 Spalart Allmaras turbulence model

The Spalart Allmaras (SA) model, first introduced in 1992, by P. Spalart and S. Allmaras [Spalart 94], has rapidly gained widespread popularity especially in aerodynamics. It showed good stability and performance at high pressure gradients. The turbulence model involves the resolution of a nonlinear advection-diffusion-reaction equation for a modified eddy viscosity  $\tilde{\nu} = \frac{\mu}{\rho}$ ,  $\mu$  being the dynamic viscosity and  $\rho$  the density of the fluid, such that [Oliver 09]:

$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{v} \cdot \nabla \tilde{\nu} - c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} + \left[ c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2} \right] \left( \frac{\tilde{\nu}}{d} \right)^2 - \frac{c_{b2}}{\sigma} \nabla \tilde{\nu} \cdot \nabla \tilde{\nu} - \frac{1}{\sigma} \nabla \cdot [(\nu + \tilde{\nu})\nabla \tilde{\nu}] = 0 \quad (1.84)$$

The eddy viscosity is computed by  $\mu_t = f_{v1}\tilde{\nu}$  where:

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad f_{t2} = c_{t3}e^{-c_{t4}\chi^2} \quad (1.85)$$

$$f_w = g \left[ \frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right]^{\frac{1}{6}}, \quad g = r + c_{w2}(r^6 - r), \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2},$$

with

$$\tilde{S} = \begin{cases} S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, & \text{for } \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \geq -c_{v2}S \\ S + \frac{S(c_{v2}^2 S + c_{v3} \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2})}{(c_{v3} - 2c_{v2})S - \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}}, & \text{for } \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} < -c_{v2}S, \end{cases} \quad (1.86)$$

In these formulas,  $S$  represents the vorticity magnitude and it is evaluated by:

$$S = \sqrt{2S_{ij}S_{ij}}, \quad S_{ij} = \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \quad (1.87)$$

inhere we denote by  $d$  the shortest distance to the wall, and by  $\kappa = 0.41$  the Von Kármán constant. We set the remaining model coefficients to:

$$c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad \sigma = 2/3, \quad c_{v1} = 7.1, \quad c_{v2} = 0.7, \quad c_{v3} = 0.9, \\ c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{t3} = 1.2, \quad c_{t4} = 0.5. \quad (1.88)$$

To improve the accuracy and convergence of the model, and to avoid negative viscosity values, some modifications shall be performed. We find in the literature several variations of this model, the interested reader is advised to consult NASA's excellent *turbulence modeling resource* webpage [Rumsey 13]. In this work we will use the *Negative Spalart Allmaras Model* because of its robustness in avoiding the creation of negative turbulent viscosity [Allmaras 12] when the closure function  $f_{v2}$  is negative, i.e. for  $1 \leq \chi \leq 18.4$ . In fact, the exact solution of the model is indeed non-negative, whereas the discrete solution does not share the same property and hence can adversely im-

pace its convergence. Two modifications are encountered, the first consists in setting the turbulent viscosity  $\mu_t$  to zero when  $\tilde{v}$  takes negative values. The second involves replacing (1.84) when  $\tilde{v}$  is negative by:

$$\frac{\partial \tilde{v}}{\partial t} + \left( \mathbf{v} - \frac{c_{b2}}{\sigma} \nabla \tilde{v} \right) \cdot \nabla \tilde{v} - c_{b1}(1 - c_{t3})S\tilde{v} - c_{w1} \left( \frac{\tilde{v}}{d} \right)^2 - \frac{1}{\sigma} \nabla \cdot [(\nu + f_n \tilde{v}) \nabla \tilde{v}] = 0, \quad (1.89)$$

with  $f_n = \frac{c_{n1} + \chi^3}{c_{n1} - \chi^3}$ ,  $c_{n1} = 16$ . That way, when  $\tilde{v} < 0$ , the added terms act in a way to dissipate the energy of the turbulent model  $(\frac{\tilde{v}}{d})^2$ .

An additional enhancement of the model to ensure the reduction of the energy of the turbulence model's working variable over time was proposed in [Oliver 08]. We omit the details for brevity and direct the interested reader to consult [Oliver 08].

We then solve equations (1.84) and (1.89) using the Streamline Upwind Petrov-Galerkin (SUPG) method described earlier in this chapter. The stabilization is preceded by a temporal discretization of equation (1.84). Without loss of generality we use a backward Euler scheme, and a Newton Raphson linearization. Rearranging the terms, equation (1.84) can be cast in a convection-diffusion-reaction form:

$$\begin{aligned} \frac{\tilde{v}^{i+1} - \tilde{v}^i}{\Delta t} + \underbrace{\left( \mathbf{v}^i - \frac{c_{b2}}{\sigma} \nabla \tilde{v}^i \right) \cdot \nabla \tilde{v}^{i+1}}_{\text{convection}} - \underbrace{\frac{1}{\sigma} \nabla \cdot [(\nu + \tilde{v}^i) \nabla \tilde{v}^{i+1}]}_{\text{diffusion}} \\ - \underbrace{\left[ c_{b1}(1 - f_{t2})\tilde{S}^i + \left( c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2} \right) \frac{\tilde{v}^i}{d^2} \right]}_{\text{reaction}} \tilde{v}^{i+1} = 0, \end{aligned} \quad (1.90)$$

with  $i$  being the iteration number. The advantage of the Spalart Allmaras over the  $k - \epsilon$  model resides in its ease of application. The user does not have to worry about near wall regions because the method automatically detects and resolves boundary layers.

## 1.5 CONJUGATE HEAT TRANSFER

Driven by the increasing industrial needs for the understanding and modelling of the critical heating processes involved in workpiece treatment inside furnaces, we attempt at investigating, in the last part of this chapter, transient conjugate heat transfer and turbulent fluid flow problems. These applications constitute a serious subject for researchers and engineers in the field of material sciences especially with the increasing concerns about reducing energy consumption and pollutant emissions, and lowering economic expenses [Ishii 98, Nieckele 04]. In the aim of attaining optimal temperature and metallurgical properties of treated pieces, several factors get into the game: lowering thermal gradients and ensuring a uniform temperature distribution within the load, avoiding surface defects (e.g. skid marks), optimizing the furnace functioning in terms of time scheduling and control strategies [Song 97].

In view of these demands, the design of a computational fluid dynamics (CFD) tool

is mandatory for offline usage to investigate what-if scenarios concerning the complex three-dimensional furnace simulations, such as sensitivity analysis related to the positioning of the treated workpieces, the location and orientation of the burners, and the intensity and speed of hot gas injections.

To conduct these kinds of simulations, we intend to employ the Immersed Volume method for multi-domain computations [Bernacki 08, Valette 09, Hachem 13]; this approach will be detailed in chapter 5. In a nutshell, the method consists in using a single global mesh to handle the solid and fluid sub-domains and consequently solve only one set of equations with different thermo-mechanical properties, instead of considering one set of equations for each sub-domain. By following this approach, we avoid the challenging task of determining heat transfer coefficients at the level of the fluid-solid interface. The conjugate heat transfer and flow dynamics inside the enclosure require the simultaneous resolution of the Navier-Stokes and the heat transfer equations. In convection dominated regimes, the equations are stabilized using the previously mentioned SUPG and VMS approaches (in sec. 1.2.4 and 1.3.3 respectively). The EM-I (discussed in sec. 1.2.11) is used to numerically deal with thermal shocks that occur as a result of a solid's sudden heating. Finally, the radiative heat transfers are accounted for by solving the radiative transfer equation (RTE) coupled with the P-1 radiation model [Modest 93]. The resolution of the P-1 model generates a volume term,  $\nabla q_r$ , that is added as a source term to the heat transfer equation and rendered by the sharp discontinuities of the temperature field and material properties.

The fluid dynamics and heat transfers are modelled by the Reynolds-Average Navier-Stokes, the energy equations and the k- $\epsilon$  model. The Boussinesq approximation is used to accommodate the buoyancy forces on the fluid motion within the domain. Accordingly, one has to solve the following coupled non-linear system [Jones 72]:

$$\begin{array}{rcl} \nabla \cdot \mathbf{v} = 0 & \text{in } \Omega & (1.91a) \\ \rho(\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) - \nabla \cdot (2\mu_e \boldsymbol{\varepsilon}(\mathbf{v}) - p_e \mathbf{I}_d) = \rho_0 \beta (T - T_0) \mathbf{g} & \text{in } \Omega & (1.91b) \\ \rho C_p (\partial_t T + \mathbf{v} \cdot \nabla T) - \nabla \cdot (\lambda_e \nabla T) = f - \nabla \cdot \mathbf{q}_r & \text{in } \Omega & (1.91c) \end{array}$$

where  $\mathbf{v}$  denotes the velocity field,  $p_e$  the effective pressure,  $T$  the temperature,  $\boldsymbol{\varepsilon}(\mathbf{v})$  the deformation-rate tensor,  $\rho$  and  $\mu$  the density and dynamic viscosity,  $\rho_0$  and  $T_0$  are reference density and temperature,  $\beta$  the thermal expansion coefficient and  $\mathbf{g}$  the gravitational acceleration.

Eventually, equation (1.91c) denotes the energy conservation and it involves the constant pressure specific heat capacity  $C_p$ , the effective thermal conductivity  $\lambda_e$ , a volume source term  $f$  and the heat radiative flux  $\mathbf{q}_r$ . We highlight the presence of the effective viscosity  $\mu_e$  and effective thermal conductivity  $\lambda_e$  terms which are computed by:

$$\mu_e = \mu + \mu_t \quad \text{and} \quad \lambda_e = \lambda + \frac{C_p \mu_t}{Pr_t}, \quad (1.92)$$

with  $Pr_t = 0.85$  being the turbulent Prandtl number. The turbulent viscosity  $\mu_t$  in equation (1.92) is a function of the turbulent kinetic energy  $k$  and the turbulent dissipation  $\epsilon$  computed through the resolution of the  $k - \epsilon$  model introduced in section 1.4.2.1 of this chapter. It is important to mention that we omitted the over-lined notation used previously for denoting the averaging of the problem variables ( $\mathbf{v}$ ,  $p_e$  and  $T$ ) for sake of simplicity; however, when using the RANS method, all the variables stand for their averaged counterpart.

The radiation's contribution to heat transfers will be accounted for by solving the radiative transfer equation (RTE) coupled with the P-1 radiation model. This model enables the approximation of radiation intensity using a series of spherical harmonics. For additional details, the reader is directed to [Modest 93, Siegel 02]. It also allows the simplification of the RTE into an elliptical partial differential equation in terms of the incident radiation  $G$  as follows:

$$\begin{cases} \nabla \cdot \left( \frac{1}{3\kappa_a} \nabla G \right) - \kappa_a G = 4\kappa_a \sigma T^4 & \text{in } \Omega \\ \frac{\partial G_w}{\partial \mathbf{n}} = \frac{3\kappa_a \epsilon_w}{2(2 - \epsilon_w)} (4\sigma T_w^4 - G_w) & \text{in } \partial\Omega \end{cases} \quad (1.93)$$

with  $\kappa_a$  denoting the mean absorption coefficient, and  $\sigma$  the Stefan-Boltzmann constant. In equation (1.93) subscript w stands for the wall quantities,  $\mathbf{n}$  the normal to the wall and  $\epsilon_w$  the wall emissivity.

Finally, the radiative source term in equation (1.91c) can be determined from the local temperature and the incident radiation according to the following equation:

$$-\nabla \cdot \mathbf{q}_r = \kappa_a (G - 4\kappa_a \sigma T^4) \quad (1.94)$$

All throughout this thesis, a gray-medium assumption is considered whereby  $\kappa_a$  is determined from the emissivity  $\epsilon$  according to Bouguer's law:

$$\kappa_a = -\frac{1}{L_m} \ln(1 - \epsilon), \quad (1.95)$$

where  $L_m$  represents the mean beam length and is computed as follows:

$$L_m = 3.6 \frac{\Delta V}{\Delta S}, \quad (1.96)$$

with  $\Delta V = \Delta x \Delta y \Delta z$  and  $\Delta S = 2(\Delta x \Delta y + \Delta y \Delta z + \Delta z \Delta x)$  being respectively the volume and surface of each element in the mesh [Siegel 02].

An  $M_1$  radiation model is under development in the thesis work of Schmid [Schmidt 16] and intends to increase the precision in modelling the radiative effects onto the heat transfer equation.

### System Resolution

All the previously described finite elements solvers were implemented in CIMLIB library. The resulting algebraic problems were assembled and solved using a General-

ized Minimal Residual Method (GMRES) associated with an Incomplete LU, ILU(o), preconditioner provided by the Portable Extensive Toolkit for Scientific Computation (PETSc) library [Balay 97]. The latter is well designed and robustly supported to account for massively parallel resolutions of linear systems of equations. The computations were performed in a master-slave parallel framework [Digonnet 03] made up of 2.4 Ghz Opteron cores in parallel (linked by an Infiniband network) and involving SPMD (Single Program, Multiple Data) modules and the MPI (Message Passing Interface) library standard.

## 1.6 NUMERICAL VALIDATION

In this section, we will present a series of numerical experiments to assess the validity, accuracy and efficiency of the proposed framework of stabilization techniques and turbulence modelling. The numerical simulations were carried out using the C++ CIMLIB finite element library. We employ an Euler implicit discretization in time and a Newton-Raphson linearization approach to treat the nonlinear convective terms.

### 1.6.1 Turbulent flow behind a prismatic cylinder (2D)

In the first example, we intend to evaluate the performance of the VMS stabilization together with the Spalart-Allmaras model in simulating convection dominated flows past an obstacle with sharp corners. For comparison purposes, the geometry and setup of the problem (fig. 1.6) are taken the same as in [Bao 11]. The problem involves a free stream flow past a prismatic obstacle with possibility of varying its aspect ratio.

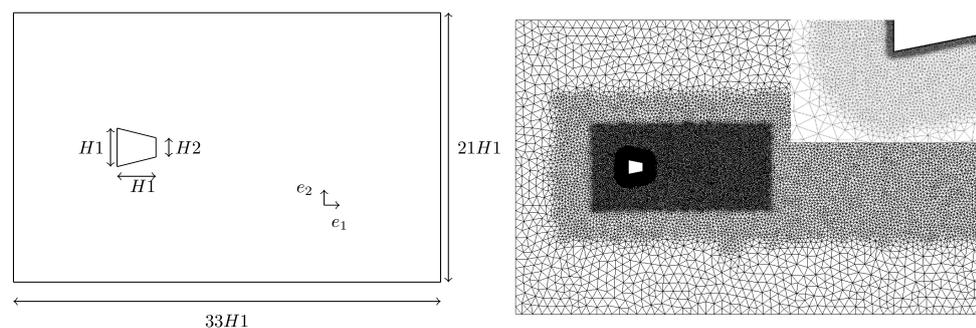


Figure 1.6 – Flow past a prismatic cylinder: sketch of the geometrical domain (left) and the corresponding gradual mesh construction (right).

The intention in this example is to demonstrate the capability of the implemented method to well reproduce an oscillation free solution for complex flows in the presence of sharp angles. For that purpose we will limit the study to an aspect ratio  $\frac{H_2}{H_1} = 1$ . The numerical investigation for different shape aspect ratio obstacles were performed and reported (with and without anisotropic mesh adaptation) in a recent publication

by Cremonesi et al. [Cremonesi 14]. In here we retrace the results of [Cremonesi 14] obtained with  $\frac{H_2}{H_1} = 1$ .

We prescribe the inflow velocity  $(V_{in}, 0)$  and the flow viscosity  $\mu$  to yield a Reynolds number  $Re = 2.2 \times 10^4$ . The turbulent eddy viscosity is set to  $\tilde{\nu} = 3\mu$  as suggested in [Bao 11]. No slip boundary conditions  $\mathbf{v} = \mathbf{0}, \tilde{\nu} = 0$  are assigned at the level of the solid body whereas slip conditions are imposed on the horizontal walls  $\mathbf{v} \cdot \mathbf{n} = 0, (\nabla \mathbf{v} \mathbf{e}_1) \cdot \mathbf{n} = 0, \nabla \tilde{\nu} \cdot \mathbf{n} = 0$ . The solution is advanced in time with a time-step equal to 0.002s. For this numerical test case, we have pre-adapted the mesh by forecasting the solution's behavior assuming that the size of eddies grows as we move away from the solid body. We also noted that upwind to the object, the flow is laminar. Based on these assumptions on the characteristic nature of the expected flow, we have generated a sublayered mesh, depicted in figure 1.6(right), where the elements' sizes decrease as one approaches the prismatic solid. Further attention is given to the construction of the first layer of elements by imposing the elements mesh sizes to  $h = 3.8 \times 10^{-4} H_1$  in accordance with [Bao 11]. This choice of mesh size implies an optimal  $y^+$  value, equal to 1, according to Schlichting's formula [Schlichting 68]. We note that the  $y^+$  value (sec. 1.4.2.1) is often considered as a mesh refinement indicator in aerodynamic applications.

Two aerodynamic quantities are of interest in this simulation: the drag and lift coefficients. We will briefly outline the procedure to compute these quantities and refer the reader to [Volker 04a] for a detailed derivation. The drag and lift coefficients are obtained from the drag and lift forces given by:

$$F_d = \int_S \left( \mu \frac{\partial \mathbf{v}_t}{\partial n} n_y - p n_x \right) dS \quad \text{and} \quad F_l = - \int_S \left( \mu \frac{\partial \mathbf{v}_t}{\partial n} n_x + p n_y \right) dS, \quad (1.97)$$

where  $S$  represents the fluid-solid interface with normal  $\mathbf{n} = (n_x, n_y)$ ,  $\mathbf{v}_t$  the tangential velocity and  $\mathbf{t} = (n_y, -n_x)$  the tangential vector. The drag  $C_d$  and lift  $C_l$  coefficients are then defined as:

$$C_d = \frac{2F_d}{\rho \bar{V}^2 D} \quad \text{and} \quad C_l = \frac{2F_l}{\rho \bar{V}^2 D} \quad (1.98)$$

where  $\bar{V}$  is the mean velocity of the fluid,  $\rho$  its density and  $D$  is the characteristic length of the prismatic cylinder. According to [Volker 04a] these coefficients are determined using the vector of aerodynamic forces  $\mathbf{f} = (F_d, F_l)$  given by:

$$\begin{aligned} \mathbf{f} &= \int_S (\mu (\nabla \mathbf{v} \cdot \mathbf{n}) \cdot \mathbf{t}) \mathbf{t} - p \mathbf{n} \, dS = \int_S 2\mu ((\boldsymbol{\epsilon}(\mathbf{v}) \cdot \mathbf{n}) \cdot \mathbf{t}) \mathbf{t} - p \mathbf{n} \, dS \\ &= \int_S (\boldsymbol{\sigma} \mathbf{n}) \, dS \end{aligned} \quad (1.99)$$

obtained using the properties  $\nabla \mathbf{v} \cdot \mathbf{t} = \mathbf{0}$  and  $(\boldsymbol{\epsilon}(\mathbf{v}) \cdot \mathbf{n}) \cdot \mathbf{n} = 0$ . We set:  $\mathbf{w}_d \in [H_{\partial\Omega \setminus S}^1(\Omega)]^d$  so that  $\mathbf{w}_d|_S = \mathbf{e}_x$  in the sense of traces. Multiplying the Navier-Stokes momentum equation by  $\mathbf{w}_d$  and integrating by parts (while taking the inward pointing

normal to the boundary) implies:

$$\int_{\Omega_f} \partial_t \mathbf{v} \cdot \mathbf{w}_d + \int_{\Omega_f} \boldsymbol{\sigma} : \nabla \mathbf{w}_d + \int_{\partial\Omega_f} (\boldsymbol{\sigma} \mathbf{n}) \cdot \mathbf{w}_d + \int_{\Omega_f} (\mathbf{v} \cdot \nabla) \mathbf{v} \cdot \mathbf{w}_d = 0. \quad (1.100)$$

Rearranging the terms and based on the fact that  $\mathbf{w}_d|_{\partial\Omega \setminus S} = \mathbf{0}$ , a new formulation for the drag force results:

$$F_d = - \int_{\Omega_f} \partial_t \mathbf{v} \cdot \mathbf{w}_d - \int_{\Omega_f} \boldsymbol{\sigma} : \nabla \mathbf{w}_d - \int_{\Omega_f} (\mathbf{v} \cdot \nabla) \mathbf{v} \cdot \mathbf{w}_d. \quad (1.101)$$

Once we have determined the drag force, we substitute it in definition (1.98) to obtain the drag coefficient. We compute the lift coefficient in an analogous way:

$$F_l = - \int_{\Omega_f} \partial_t \mathbf{v} \cdot \mathbf{w}_l - \int_{\Omega_f} \boldsymbol{\sigma} : \nabla \mathbf{w}_l - \int_{\Omega_f} (\mathbf{v} \cdot \nabla) \mathbf{v} \cdot \mathbf{w}_l. \quad (1.102)$$

Figure 1.7(left) shows the turbulent eddy viscosity  $\tilde{\nu}$  once a periodic steady vortex shedding profile is reached. We can identify that the turbulence model has been activated in specific zones and thus the effect of the averaging and damping functions are well reflected. The obtained velocity streamlines, figure 1.7(right), are quite stable and present localized recirculation close to the solid body. This velocity profile is in accordance with the one described in the reference [Bao 11] hence confirming the validity of coupling the VMS solver to the Spalart Allmaras model.

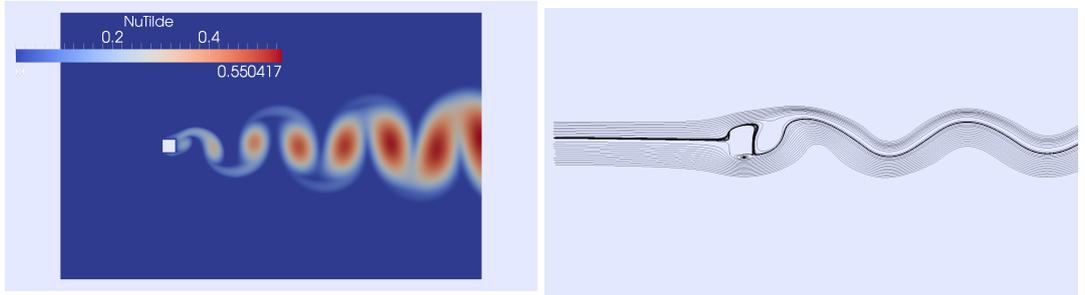


Figure 1.7 – Flow past a prismatic cylinder: profile of the turbulent eddy viscosity  $\tilde{\nu}$  (left) and velocity streamlines (right).

In Figure 1.8 we present the evolution of drag and lift coefficients once a steady vortex shedding behavior is obtained. A comparative study, on the mean drag coefficient and root mean squared lift coefficient obtained in this work together with experimental and numerical results from the literature, is reported in table 1.1. We identify that the results are in good agreement with the references though few differences can be detected. These differences might be due to the fact that the computational mesh is not the same and more importantly to the first order discretization in time, as highlighted in [Volker 04a].

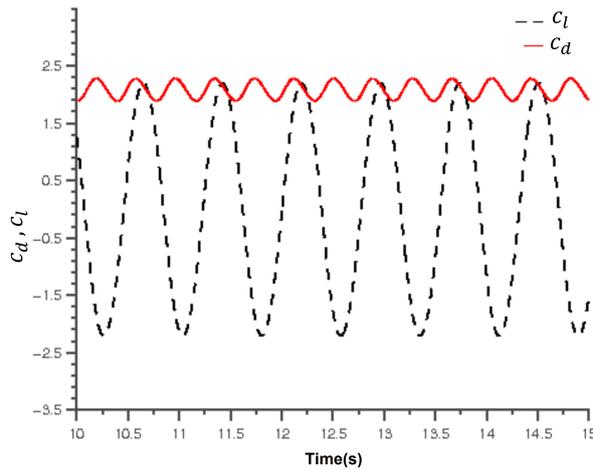


Figure 1.8 – Flow past a prismatic cylinder: drag (solid) and lift (dotted) coefficients time evolution.

|                                   | mean $C_D$ | r.m.s. $C_L$ |
|-----------------------------------|------------|--------------|
| Bearman and Obasaju [Bearman 82]  | 2.10       | 1.20         |
| Sohankar et al. [Sohankar 00]     | 2.25       | 1.50         |
| Shimada and Ishihara [Shimada 02] | 2.05       | 1.43         |
| Bao et al. [Bao 11]               | 2.04       | 1.24         |
| Present work                      | 2.08       | 1.57         |

Table 1.1 – Flow past a prismatic cylinder: drag and lift coefficients' comparison with the references.

## 1.6.2 Natural convection (2D)

In the second numerical test case, we aim at evaluating the accuracy of the stabilization methods that we use for solving coupled problems. We compare the performance of the methods to results found in the literature on the natural convection benchmark. The problem consists in solving a classical flow in a cavity with differentially heated vertical walls. The fluid motion inside the cavity is driven by the thermal gradient and the Boussinesq forces. In fact, Boussinesq [Boussinesq 03] stated that in a natural convection flow, the density exhibits small fluctuations while maintaining the incompressibility condition. The density field can thus be decomposed into a constant uniform value  $\rho_0$  and small scale variations  $\rho'(\mathbf{x}, t)$ :

$$\rho(\mathbf{x}, t) = \rho_0 + \rho'(\mathbf{x}, t). \quad (1.103)$$

According to Boussinesq, the density variations are computed with respect to the temperature reference value  $T_0$  as follows:

$$\rho'(\mathbf{x}, t) = -\rho_0 \frac{T'}{T_0} = -\rho_0 \frac{T - T_0}{T_0} \quad (1.104)$$

On the other hand, the thermal expansion coefficient is defined by:  $\beta[K^{-1}] = \frac{1}{\rho_0} \left( \frac{\partial \rho}{\partial T} \right)_{T_0} = \frac{1}{T_0}$ . Hence the small scale density fluctuations are computed by:

$$\rho' = -\rho_0 \beta (T - T_0). \quad (1.105)$$

Consequently, an additional force term is introduced into the momentum equation to account for these variations in the density field.

The problem was first investigated for small temperature field differences. It has recently been studied for large temperature differences and low Mach number and benchmarking solutions were established. For comparison purposes, we study the test case described in [Le Quéré 05, Paillère 05]. The computational domain, depicted in figure 1.9 is a square cavity of length  $L$  (inhere we set  $L = 1$ ).

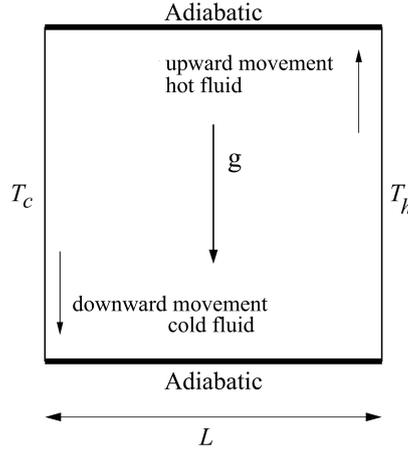


Figure 1.9 – Natural convection 2D: sketch of the computational domain.

We assume constant fluid properties and negligible radiation effects. The left wall is kept at a high temperature  $T_h$  whereas the right wall is maintained at a cold temperature  $T_c$ . The horizontal walls are set to adiabatic temperature. We recall that the state equation for air follows the perfect gas law ( $R = 287 \text{ J kg}^{-1} \text{ K}^{-1}$ ). The Rayleigh number is an adimensional number characterizing coupled heat transfer and fluid flows. It quantifies the ratio between buoyant forces and the product of viscous drag by the thermal conductivity. It is defined by:

$$\text{Ra} = \frac{\rho_0 g \beta (T_h - T_c) L^3 C_p}{\mu \lambda}, \quad (1.106)$$

where  $\mu [\text{Pas}]$  is the dynamic viscosity,  $\lambda [\text{W m}^{-1} \text{ K}^{-1}]$  the thermal conductivity,  $C_p [\text{J K}^{-1} \text{ kg}^{-1}]$  the specific heat transfer and  $g [\text{m s}^{-2}]$  the gravitational acceleration.

Another important dimensionless characteristic number is the Prandtl number, it measures the ratio between viscous diffusion rate and thermal diffusion rate:

$$\text{Pr} = \frac{\mu C_p}{k} \quad (1.107)$$

We define the problem parameters in such a way to get a Prandtl number equal to 0.71, Rayleigh numbers  $10^6$ ,  $10^7$  and  $10^8$ , a relative temperature difference  $\varepsilon_T = \frac{T_h - T_c}{T_0} = 1.2$

and an isentropic exponent  $\gamma = 1.4$  so that the specific heat capacity evaluates to:

$$C_p = \frac{\gamma R}{\gamma - 1} = 1004.5.$$

We set the initial temperature field to:

$$T(\mathbf{x}, 0) = T_0 = \frac{T_h + T_c}{2} = 600\text{K} \quad (1.108)$$

and the initial pressure to  $P_0 = 101325\text{Pa}$ . We can now determine the constant density of the fluid from  $\rho = \frac{P_0}{RT_0}$ . The fluid is initially at rest inside the cavity, and no slip boundary condition is prescribed on the boundaries of the domain. The temperature values at the hot and cold walls are computed from  $T_h = T_0 \left(1 + \frac{\epsilon_T}{2}\right)$  and  $T_c = T_0 \left(1 - \frac{\epsilon_T}{2}\right)$ .

The computations were performed on a mesh made up of around 50,000 nodes and evolved in time using a constant time-step  $\Delta t = 0.05\text{s}$ . We show in figure 1.10 the velocity streamlines at time  $t = 20\text{s}$  obtained for Rayleigh numbers  $10^6$ ,  $10^7$  and  $10^8$ . We can clearly identify that for a fixed Prandtl number, with the increase of the Rayleigh number, the gravitational forces become more dominant and consequently the flow starts developing turbulent motion with the emergence of different sizes vortices. The symmetric profile obtained about the centerlines reflects well the incompressibility property of the flow.

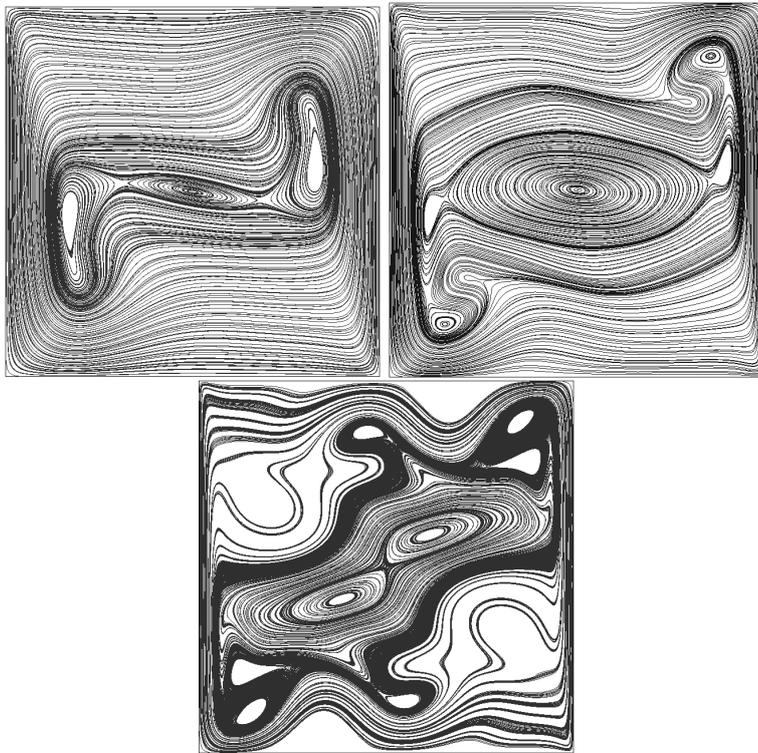


Figure 1.10 – Natural convection 2D: velocity streamlines at time  $t = 20\text{s}$  for  $Ra=10^6$ ,  $10^7$  and  $10^8$ .

Figure 1.11 depicts the temperature isotherms at time  $t = 20\text{s}$  for the different

Rayleigh numbers. We can identify how the temperature responds to expectations. Indeed, the fluid initially at rest inside the cavity gets in contact with the hot boundary; it gets warmer and its density decreases. Consequently the heated fluid rises. On the other hand, near the cold boundary, the fluid gets colder, its density increases and consequently falls. This process is at the origin of the motion inside the cavity.

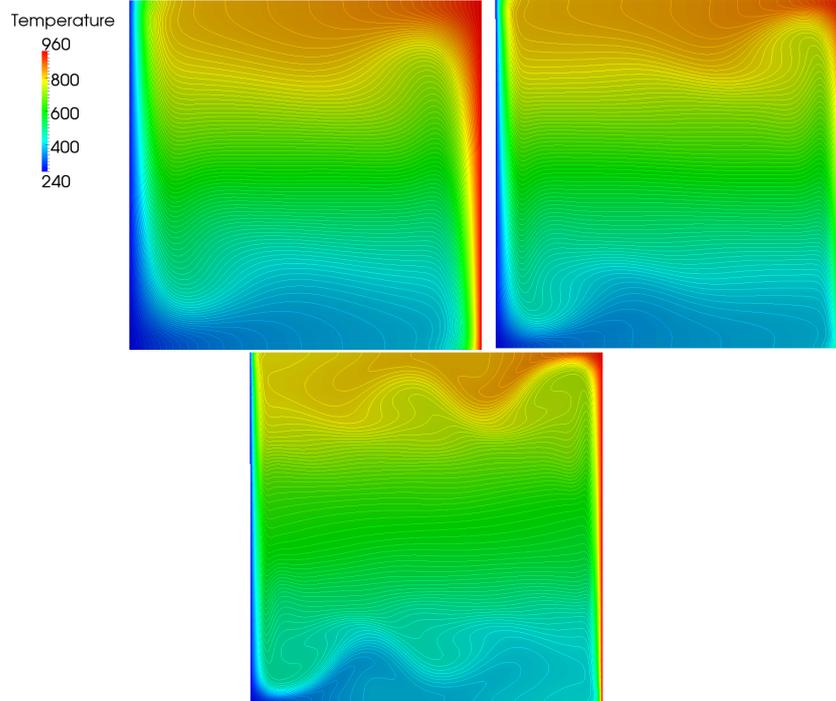


Figure 1.11 – Natural convection 2D: temperature isotherms at time  $t = 20s$  for  $Ra=10^6$ ,  $10^7$  and  $10^8$ .

Quantitatively, we are interested in evaluating the accuracy of the implemented stabilized finite element methods by comparing the Nusselt number to existing benchmark values present in the literature [Paillère 05]. We recall that the Nusselt number measures the ratio of the convective to conductive heat transfer across the solid interface. In this example the interface is considered at the vertical walls of the domain and the Nusselt number is calculated by:

$$\text{Nu} = \frac{1}{T_h - T_c} \left| \frac{\partial T}{\partial x} \right| \quad \text{and} \quad \overline{\text{Nu}} = \frac{1}{L} \int_0^L \text{Nu}(x, y) dy \quad (1.109)$$

We present in table 1.2 statistics on the Nusselt number obtained in the present work and in the literature [Paillère 05] for  $Ra = 10^6$ . We can clearly identify a good agreement with the references. In addition, we approach better the benchmark values as we increase the number of degrees of freedom in the mesh. The small discrepancies in the results with respect to the references can be attributed to the difference in the discretization schemes. It is important to mention that in the references, quadratic finite element method and semi-implicit time discretization have been applied whereas

our results were obtained using a linear finite element method and an implicit time discretization on unstructured meshes.

|                 | Vierendeels       | Dabbene  | Beccantini | Kloczko  | Heuveline | Present work | Present work |
|-----------------|-------------------|----------|------------|----------|-----------|--------------|--------------|
| $Nu(0, y)$      | 8.85978           | 8.86380  | 8.85990    | 8.86200  | 8.859778  | 8.73698      | 8.79891      |
| $Nu(1, y)$      | 8.85978           | 8.86200  | 8.86007    | 8.86380  | 8.85978   | 8.73056      | 8.81761      |
| $Nu(0, 0.5)$    | 7.81938           | 7.82170  | 7.81978    | 7.82010  | N.A.      | 7.65966      | 7.7664       |
| $Nu(1, 0.5)$    | 8.79636           | 8.81710  | 8.79646    | 8.79750  | N.A.      | 8.61058      | 8.76575      |
| $\max Nu(0, y)$ | 19.59642          | 19.62600 | 19.59538   | 19.61070 | 19.59633  | 19.19471     | 19.4634      |
| $\min Nu(0, y)$ | 1.07345           | 1.07690  | 1.07356    | 1.07380  | 1.07345   | 1.05583      | 1.06888      |
| $\max Nu(1, y)$ | 16.36225          | 16.35200 | 16.36333   | 16.37510 | 16.36226  | 15.9499      | 16.2189      |
| $\min Nu(1, y)$ | 0.85512           | 0.86102  | 0.85542    | 0.85620  | 0.85513   | 0.86811      | 0.85007      |
| Mesh density    | $4.2 \times 10^6$ | 102400   | 87616      | 57600    | 200000    | 50000        | 175000       |

Table 1.2 – Natural convection: statistics on the Nusselt number for  $Ra = 10^6$  for mesh densities 50,000 and 175,000. Reference values are taken from [Paillère 05].

### 1.6.3 Heat treatment of workpieces inside an industrial furnace

Now that we have validated the numerical solvers, we intend in this section to simulate 12 hours of heating inside an industrial furnace provided by our industrial partner Industeel Arcelor Mittal. The furnace is modelled as a hexagonal section duct of  $2.7 \times 8.1 \times 5.3$  m forming one heat transfer zone. The hot gas is pumped into the furnace through a cylindrical burner, with diameter 660mm located on the vertical wall at a speed of 38m/s and a temperature of 1350°C. We can clearly identify the burner positioning in figure 1.13 and how the workpieces are inserted into the load from the opened top hatch. For more details about the geometry, we present in figure 1.14 different angle views. All computations were conducted by initially considering a gas at rest and having a constant temperature of 700°C. At all other boundaries, a constant flux of  $400W/m^2$  is applied for the sake of simplicity. The air is vented out of the furnace through two outlets, with diameters 700mm, positioned at the bottom of the vertical wall. We use an adaptive time-step that starts at 0.001s and increases as the solution stabilizes. The volumetric mesh used in the numerical simulation consists of 157,347 nodes and 884,941 tetrahedral elements.

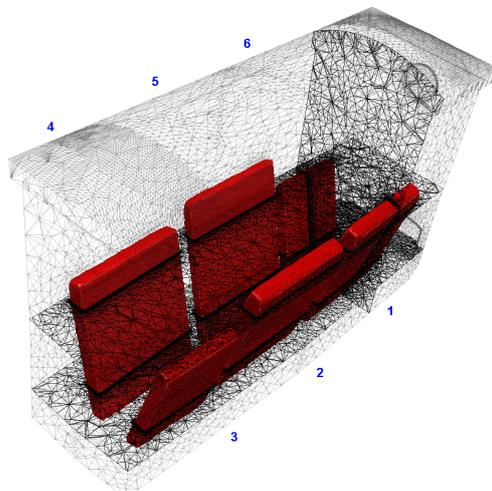


Figure 1.12 – Computational domain after anisotropic mesh adaptation.

Figure 1.12 shows six ingots taken initially at  $400^{\circ}\text{C}$  and positioned at different locations inside the furnace. We can identify (from figures 1.12 and 1.13) two types of ingots; the thick ones are placed on the left wall (1, 2 and 3) and the thin ones are placed on the right wall (4, 5 and 6).

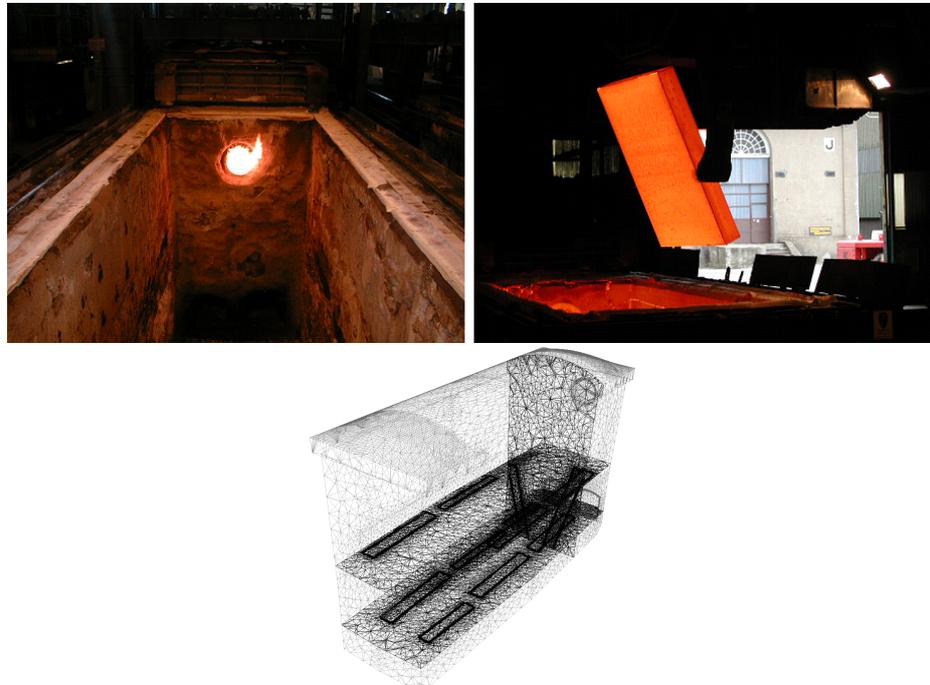


Figure 1.13 – A top view of the furnace (left) and the immersion of an ingot (right)

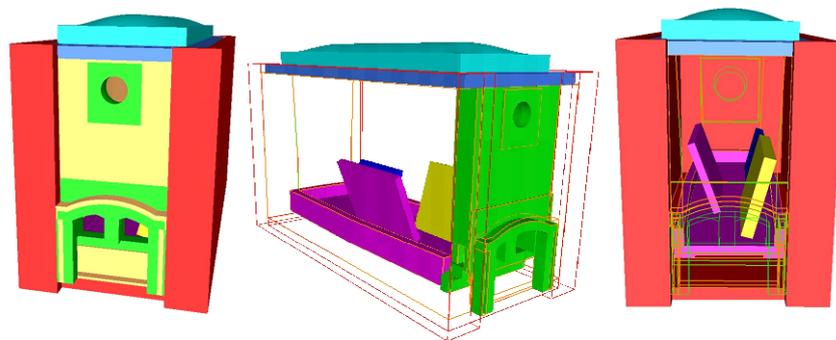


Figure 1.14 – Different angle views of the furnace.

Once the mesh has been well adapted along the ingots' interfaces, we define the distribution of the thermo-physical properties between the physical domains by means of the level set function. Consequently, the same set of conjugate heat transfer equations with turbulence modelling given by system (1.91) (including the momentum equation, energy equation, the turbulent kinetic and dissipation energy equations, and the radiative transport equation) are simultaneously solved over the entire domain. We highlight the key feature of the immersed volume method allowing a resolution on a single mesh for the whole domain including both fluid and solid regions with variable material properties (see table 1.3). Details are given in chapter 5.

Recall that the use of a high value for the relative kinematic viscosity in the solid

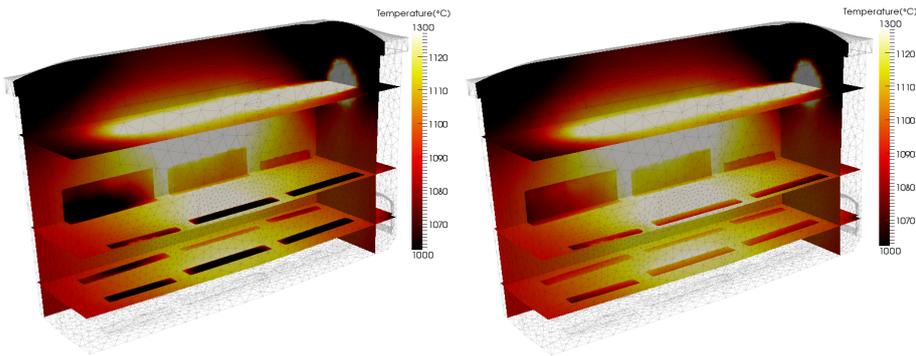


Figure 1.15 – Isotherms inside the furnace at two different simulation times.

region makes the velocity components negligible and satisfies the no-slip condition at the refined interfaces. Therefore, as it is the case for the energy equation, all the convective terms as well as the source (*i.e.* destruction) terms in the two-equations of the  $k - \varepsilon$  model drop out.

Consequently, the turbulent quantities are computed naturally at the interface. This idea was also emphasized by the authors of [Kuzmin 07]. They showed that by allowing  $k$  to be "naturally" computed at the boundary, a better prediction of the turbulent quantities in the near wall regions and correct solution's behavior were obtained.

| Properties                          | Gaz    | Steel 40CDVL3 |
|-------------------------------------|--------|---------------|
| density $\rho$ [kg/m <sup>3</sup> ] | 1.25   | 7,800         |
| heat capacity $C_p$ [J/(kg K)]      | 1000   | 600           |
| viscosity $\mu$ [kg/(m s)]          | 1.9e-5 | –             |
| conductivity $\lambda$ [W/(m K)]    | 0.0262 | 37            |
| emissivity $\epsilon$               | –      | 0.87          |

Table 1.3 – Material properties.

In the numerical simulation, the heat capacity  $C_p$ , the conductivity  $\lambda$  and the emissivity  $\epsilon$  of the smoke and the steel are thermo-dependent. The emissivity of the smoke was computed from the proportions of the  $H_2O$  and  $CO_2$  issued from the combustion, the thickness of the smoke and the temperature as in the model studied in [Heiligenstaedt 71].

All the given parameters used for the numerical simulations do not reflect the true measurements from the experimental tests, due to the complexity of the wall properties, the gas composition and other technical issues. However, we made sure that the chosen parameters have at least the real physical representations and are appropriate to simulate and give predictions on the real problem.

The 3D computations have been obtained in two steps using 16 and 40 2.4Ghz Opteron cores in parallel (linked by an Infiniband network) respectively. The first one involves an iterative process of mesh adaptation around the fixed solids. This step is a preparation phase and its cost is separated and relatively small compared to the

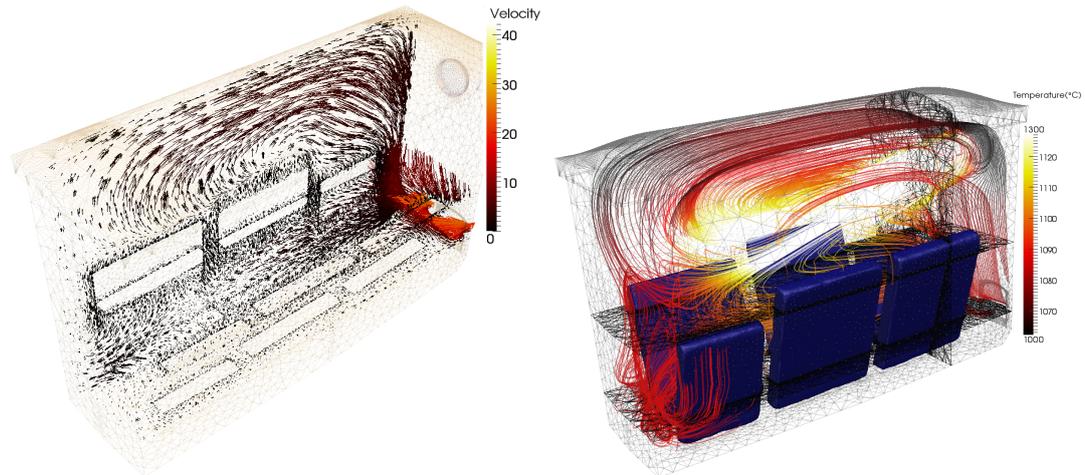


Figure 1.16 – Velocity field and streamlines distribution inside the furnace and around the ingots.

full numerical simulation. The process consists in repetitively adapting and refining the mesh in the zone around the solid/fluid interface until a well respected shape in terms of curvature, and angles is obtained. The construction took around 3 hours of computational time to render a high resolution and a good geometry capture around the fixed ingots. Once the mesh is ready, it can be used in the next step to model the heat transfer and the turbulent flows inside the furnace. This second step has cost around 9 days of computational time. The slow aspect of the simulation highlights the need to devise algorithms that will accelerate the calculations while maintaining a good level of accuracy. This is exactly the objective of this thesis. It intends to develop unsteady anisotropic mesh adaptation and adaptive time stepping capable to handle this kind of full heating sequences within reasonable time.

Figure 1.15 shows the temperature distribution on four mutual planes in the furnace at two different times ( $t= 5.5h$  and  $6h$ ).

The temperature distribution clearly reflects the expected flow pattern. At the ingots' level, we observe that the injected air from the top burner is slowed down and slightly influences the main air circulation in this part of the domain. This behavior explains the difference in the flow pattern between the top and bottom parts of the furnace. When the hot fluid spreads inside the volume of the furnace, it induces a turbulent and swirling motion within the geometry. The fast expansion of the velocity streams from the burner throat yields sharp gradients and emergence of small eddies in the zones of stationary fluid inside the furnace. This phenomenon can be interpreted as a toroidal forced convection.

The air flow pattern around the workpieces is quite complex and interesting; i.e. it allows the study of the influence of different configurations and ingots' positions to optimize the heat treatment. A number of air recirculation between the objects and their surrounding can be observed due to the turbulence dissipation and mixing between the hot and cold air. All these observations are highlighted by the streamlines in figures 1.16 and the velocity components in figures 1.17 and 1.18.

Moreover, we can clearly see on these vertical plane cross sections along the ingots that the solid region respects the zero velocity and, hence, the no-slip condition on the extremely refined interface is also verified. The obstacles (6 ingots) slow down the air circulation in the lower zone of the furnace and slightly influence the main air circulation along the walls.

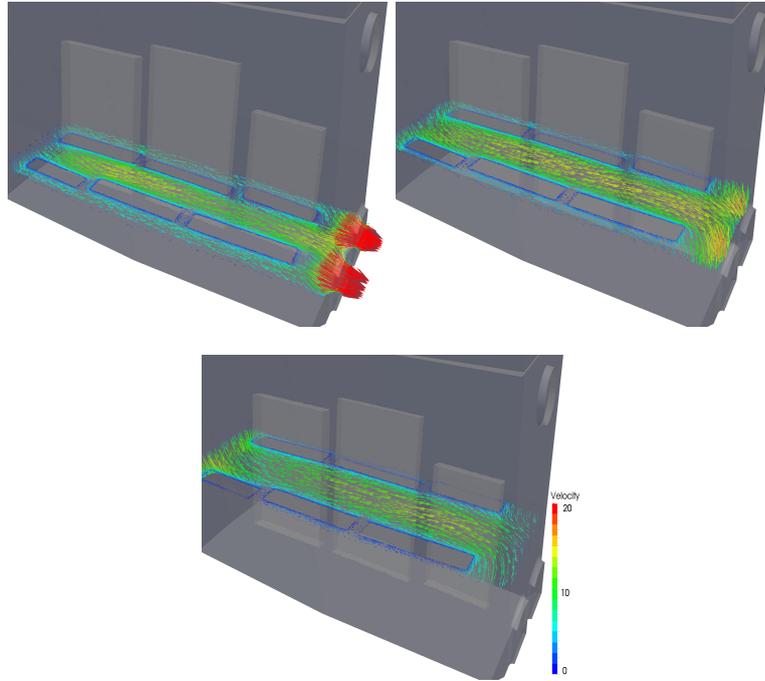


Figure 1.17 – Velocity vectors on different horizontal cut-planes inside the furnace.

To get a clear idea on the time history of the temperature, we plot in figures 1.19 and 1.20 the evolution captured at the center and at the surface of the ingots respectively. As expected, we notice that the thin ingots (4, 5 and 6) are in general heated faster than the thick ones (1, 2 and 3). At the same time, the temperature of the ingots positioned in the center and facing the flame jet continuously, increases faster than the others. This is due to the fact that the flames hit the walls and deviate towards the center yielding a slight counter clockwise rotating flow. Near the center of the furnace and under the flame jet, a full rotating gas flow is always present, and it ends near the impeller bottom-surface and exits through the two outlets.

One can also observe in figure 1.19 the presence of a certain austenitizing phase change in the material properties around  $800^{\circ}\text{C}$ . All the temperature results converge to the desired value of  $1150^{\circ}\text{C}$ . The favorable and reasonable nature of these results validate the good potential of the developed formulations. However, comparisons with experimental data having real workpiece geometries and positioning will be investigated when the proper setup of the problem is determined (combustion effects at the burner's level, gaz composition, material properties of the furnace walls, exact orientation of the burner, and several other factors that might affect the solution).

We point out that the temperature profiles are not polluted with spurious oscilla-

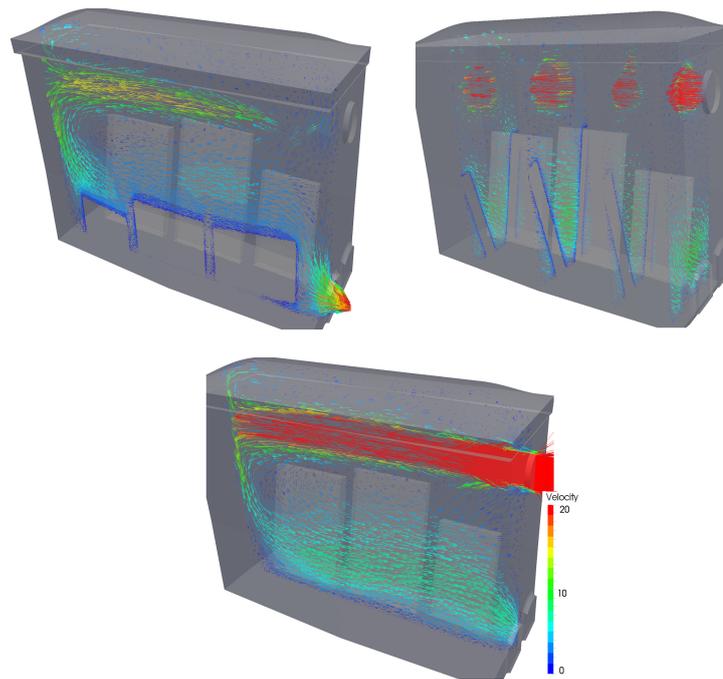


Figure 1.18 – Velocity vectors on different vertical cut-planes inside the furnace.

tions (undershoots or overshoots) that are frequently observed in the presence of high temperature gradients at the interface or in convection dominated problems. This can be attributed to the stabilized finite element discretization applied on the system of equations (1.91). Summing up, the combination of local mesh adaptation and stabilized iterative solvers with the smoothed distribution of the thermo-physical properties across the interface enables the simulation to overcome the numerical instabilities and lead to good numerical behavior.

Moreover, the numerical results validate that the proposed coupled approach (IVM) is suitable for parallel numerical simulations of conjugate heat transfer with different loads. These calculations allow the prediction of parametric properties and problem configurations as well as the understanding of the flow characteristics for heat treatment furnaces.

## 1.7 CONCLUSION

In this chapter, we have evoked different approaches for the numerical resolution of fluid flows and conjugate heat transfer problems. We adopted a finite element discretization of the computational domain for its accuracy and efficiency in solving 3D problems with complex geometries. However the standard Galerkin formulation fails to produce an oscillation free solution in convection or diffusion dominated problems especially in the vicinity of sharp gradients. Different stabilization techniques were discussed and analyzed to enhance the stability and accuracy of the standard Galerkin formulation. The Streamline Upwind Petrov-Galerkin (SUPG) method was first introduced to deal with convection dominated convection-diffusion-reaction problems. This

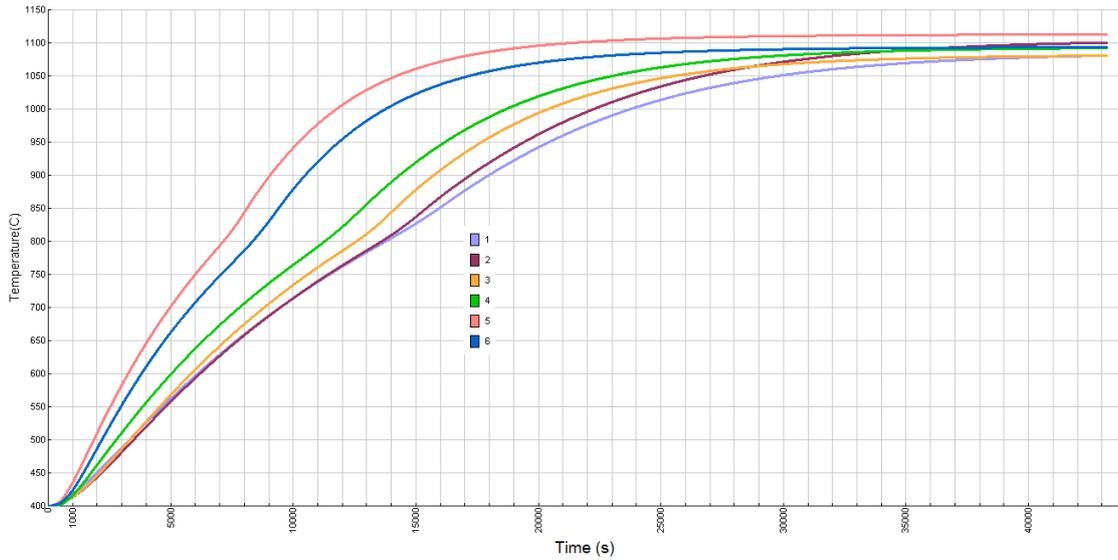


Figure 1.19 – Temperature profile evolution over time captured at the center of the immersed ingots.

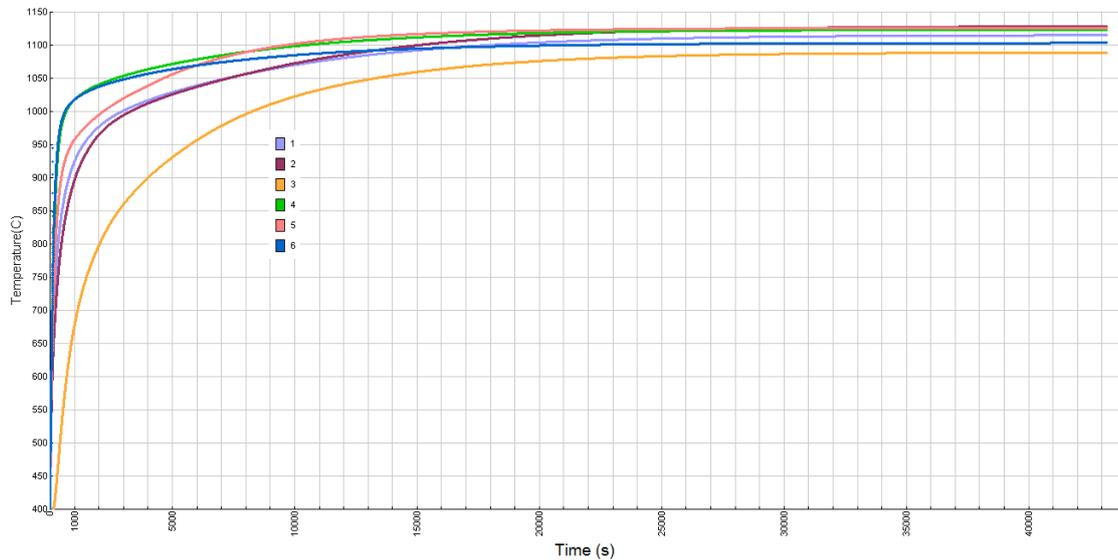


Figure 1.20 – Temperature profile evolution over time captured at the surface of the immersed ingots.

method eliminates the instabilities in the solution by adding artificial diffusion terms in the convection direction. Nevertheless, the SUPG scheme, which is neither monotone nor monotonicity preserving, does not undertake thermal shocks. For that reason, an extension to treat over/undershoots in the case of thermal shocks, known as the Shock Capturing Petrov-Galerkin (SCPG) approach, was emphasized. It provides additional stability and control in the crosswind direction. On the other hand, we have proposed to use the enriched method with time interpolation (EM-I) for its good performance in simulating conduction heat transfer problems.

We moved on next to study fluid flow problems. The variational multiscale stabilization method was employed to solve the transient incompressible Navier-Stokes equations. This method demonstrated to be a powerful tool for treating convection dominated flows and circumventing the inf-sup stability condition. In the case of high

Reynolds number flows, turbulence models were proposed in the aim of modelling the turbulent effect of small scale velocity fluctuations without solving them. These methods, by calming down the chaotic behavior of the velocity field and by avoiding the full scale resolution permit an acceleration of the computations while providing a good representation of the averaged flow profile.

Then, we presented a fully coupled system of fluid flows and heat transfer. The effects of radiation were accounted for by solving the P-1 radiation model. In the presence of solid and fluid subdomains, the thermal coupling was enhanced by using an immersed volume method (IVM). This approach will be discussed further in chapter 5. The advantage of this method resides in its simplicity and efficiency in distributing the thermo-physical properties of different materials. No heat transfer coefficient is needed and a single mesh is used for the different sub-domains.

Finally, the accuracy of the methods was studied on two and three dimensional problems. The results obtained with the proposed stabilization tools were in very good agreement with the ones found in the literature. The models also proved their capability in simulating complex three dimensional heat transfer and turbulent flow problems in the presence of conductive solids.

However, preserving a good level of accuracy necessitates a very fine mesh and small time-steps hence yields a considerable computational time and requires an important computer power. This drawback makes the resolution of complex 3D real simulations and industrial applications quasi-impossible. The objective of this thesis is to overcome these difficulties and improve the computations while maintaining accurate solutions. The work consists in devising space and time adaptive algorithms well suited for the simulation of real phenomena.

## 1.8 RÉSUMÉ FRANÇAIS

Dans ce chapitre, nous avons évoqué différentes approches numériques pour la résolution des problèmes d'écoulements de fluides et de transferts thermiques. Nous avons adopté une discrétisation du domaine de calcul par éléments finis pour sa précision et son efficacité dans la résolution de problèmes 3D avec des géométries complexes. Cependant, la formulation Galerkin standard présente des oscillations numériques dans le cas de problèmes à convection ou diffusion dominantes en particulier dans les voisinages de forts gradients. Différentes techniques de stabilisation ont été discutées et analysées pour améliorer la stabilité et la précision de la formulation Galerkin standard. Une approche Streamline Upwind Petrov Galerkin a été introduite pour traiter les problèmes à convection dominante. Cette approche sera discutée plus en détail dans le chapitre 5. Enfin, la précision des méthodes adoptées a été étudiée sur des problèmes à deux et trois dimensions. Les résultats obtenus avec les outils de stabilisation étaient en très bon accord avec la littérature et ont démontré leur applicabilité dans le cadre de problèmes de traitement thermiques en présence de métaux conductibles. Cependant, maintenir un bon niveau de précision nécessite un maillage très fin et de

petits pas de temps et induit alors un coût de calcul important. Cet inconvénient rend la résolution des applications 3D réelles et industrielles quasi-impossible. L'objectif de cette thèse est de surmonter ces difficultés et d'améliorer les coûts de calculs tout en gardant une bonne précision des résultats. L'idée consiste à développer des méthodes d'adaptation espace et temps et sera élaborée dans les chapitres suivants. Ensuite, on a décrit la méthode "Variational MultiScale" (VMS) pour la stabilisation et la résolution des problèmes d'écoulement de fluide. Cette méthode a démontré être un outil puissant pour contourner la condition inf-sup de stabilité. Des modèles de turbulence ont été proposés pour traiter les cas de flux à haut nombre de Reynolds. La résolution des problèmes couplés des écoulements de fluide et de transfert thermique a été élaborée par la suite et un modèle P1 de rayonnement a été introduit pour modéliser les effets de rayonnement. La méthode Immersed Volume Method (IVM) était adoptée pour prendre en compte les interactions fluide/structure et la distribution des propriétés thermo-mécaniques. Cette méthode élimine les instabilités dans la solution en ajoutant des termes de stabilisation dans la direction de convection dominante. Néanmoins, l'approche SUPG ne traite pas les chocs thermiques dans les cas de réchauffement ou de refroidissement de pièces métalliques. Pour cette raison, une extension Shock Capturing Petrov-Galerkin a été introduite, la méthode implique un contrôle des chocs thermiques dus à la diffusion transitoire. D'autre part, nous avons proposé d'utiliser la méthode "Enriched Method with Time interpolation" (EM-I) dans le cadre de la simulation de problèmes de transfert de chaleur par conduction.

# ANISOTROPIC MESH ADAPTATION

# 2

"One's mind, once stretched by a new idea, never regains its original dimensions."

Oliver Wendell Holmes

## CONTENTS

---

|       |                                                                         |     |
|-------|-------------------------------------------------------------------------|-----|
| 2.1   | INTRODUCTION                                                            | 73  |
| 2.2   | MESH GENERATION                                                         | 73  |
| 2.2.1 | The three classical mesh generation algorithms                          | 74  |
| 2.2.2 | The topological optimization mesh generation algorithm                  | 75  |
| 2.3   | AMR: ADAPTIVE MESH REFINEMENT                                           | 80  |
| 2.3.1 | Techniques of mesh adaptation                                           | 82  |
| 2.4   | METRIC-BASED ANISOTROPIC MESH ADAPTATION                                | 83  |
| 2.4.1 | State of the art                                                        | 83  |
| 2.4.2 | Preliminary definitions and theoretical framework                       | 83  |
| 2.4.3 | Metric based mesh adaptation                                            | 85  |
| 2.4.4 | Metric construction at the nodes' level                                 | 87  |
| 2.5   | ERROR ESTIMATION FOR ANISOTROPIC MESH ADAPTATION                        | 95  |
| 2.5.1 | Overview on error estimation techniques                                 | 95  |
| 2.5.2 | Global optimization problem                                             | 96  |
| 2.5.3 | Gradient and strong continuity along the edges                          | 97  |
| 2.5.4 | Least square gradient recovery                                          | 102 |
| 2.5.5 | A posteriori error analysis                                             | 103 |
| 2.5.6 | Edge-based error estimation                                             | 106 |
| 2.5.7 | Optimal control on the $L_p$ norm of the interpolation error            | 109 |
| 2.5.8 | Optimal metric construction                                             | 114 |
| 2.5.9 | Privileged length distribution tensor                                   | 122 |
| 2.6   | ANISOTROPIC MESH ADAPTATION THROUGH LOCAL TOPOLOGICAL OPTI-<br>MIZATION | 124 |
| 2.7   | FIELDS' INTERPOLATION BETWEEN MESHES                                    | 125 |
| 2.8   | NUMERICAL EXPERIMENTS                                                   | 128 |
| 2.8.1 | Convergence tests on a quadratic function                               | 128 |
| 2.8.2 | Numerical validations on functions with steep gradients                 | 130 |
| 2.8.3 | Numerical validations on functions with multiscale variations           | 138 |

|                                |     |
|--------------------------------|-----|
| 2.9 CONCLUSION . . . . .       | 140 |
| 2.10 RÉSUMÉ FRANÇAIS . . . . . | 141 |

---

**I**N this chapter, we are interested in devising a new approach for dynamic mesh adaptation, the method relies on an edge-based error estimation and intends to construct a metric tensor that would prescribe the mesh sizes and the orientation of the elements. The mesh adaptation problem is formulated as an optimization approach whereby one seeks the optimal metric that yields the best control on the solution's interpolation error. The method that we derive is independent from the problem at hand, does not require an a priori knowledge about the solution's behavior over time and takes into account a prescribed fixed number of nodes which is an essential feature for practical and efficient computations.

## 2.1 INTRODUCTION

Nowadays, with the increase in computational power, numerical modeling has become an intrinsic tool for predicting physical phenomena and developing engineering designs. Over the last few decades, the development of computational fluid dynamics simulations and thermal analysis have considerably attracted researchers especially with the continuous needs for explanations of the naturally occurring physical flow phenomena, including pipe flow, aerospace flow, climate predictions, respiratory system and blood circulation, convective heat transfer inside combustion chambers and industrial furnaces, and many other applications. The modeling of these phenomena poses scientific complexities whose resolution requires considerable computational resources and long lasting calculations. The development of mesh adaptation techniques was motivated by the desire to devise realistic configurations and to limit the shortcomings of the traditional non-adaptive resolutions in terms of lack of solution's accuracy and computational efficiency.

Indeed, the resolution of unsteady problems with multi-scale features on a prescribed uniform mesh with a limited number of degrees of freedom often fails to capture the fine scale physical features, have excessive computational cost and might produce incorrect results. These difficulties brought forth investigations towards generating meshes with local refinements where higher resolution was needed. Mesh adaptation can thus be regarded as an essential ingredient in this recipe. We intend in this chapter to derive a new approach for dynamic mesh adaptation based on an error estimator computed on the edges of the mesh.

The chapter is structured as follows: we start, in section 2, with an overview about mesh generation techniques, then we focus on the 'topological optimization' method that we adopt throughout this thesis. In section 3, we outline the different techniques of mesh adaptation and state the one that we use in this work. Section 4 presents some preliminary definitions and exploits the theoretical framework of the metric construction at the mesh nodes. The edge-based error estimation, the derivation of the metric tensor and the theoretical validation on the control of the  $L^p$  norm of the interpolation error will be detailed in section 5. The link between the constructed metric and the generation of the mesh is pointed out in section 6. Then in section 7, we provide the interpolation method used in the numerical simulations. We present in section 8 some numerical examples validating the convergence of the developed approach, its accuracy and efficiency.

## 2.2 MESH GENERATION

Finite element methods are considered as powerful techniques for computing approximate solutions of partial differential equations. These methods replace the continuous problem with a discrete one and approximate the solution on the nodes of the discrete domain. Mesh generation is a keystone step for the numerical resolution of physical

problems by the finite element method; hence hindering the efficiency of the latter if generated manually as it can be time consuming and error prone. The construction becomes more challenging when simulating industrial problems with arbitrary and complex geometries and internal or curved boundaries. To deal with these difficulties, a number of algorithms were developed to automate the mesh generation process by dividing the physical domain into small pieces called elements. We are interested in using unstructured simplex meshes made up of triangular and tetrahedral elements that are easier to generate than quadrilateral and hexahedral ones and better at capturing complicated geometries. Note that the resolution of the mesh strongly affects the accuracy of the numerical solution. We aim in the first part of this chapter at providing a state of the art on the existing finite element mesh generation techniques and specify the advantages of the method that we use to discretize our domains. We move on next to depict the different adaptation methods. The third part details the newly developed mesh optimization technique that would generate anisotropic elements which respond to the anisotropy of the physical problems.

In this section we are interested in answering the following problematic: given the mesh of the boundary of a domain  $\Omega$ , how do we generate a mesh of  $\Omega$ ? The history of mesh generation techniques goes back to the period around 1990 that brought forth the most popular classical mesh generation techniques used nowadays: the advancing front methods, the octree methods, the Delaunay methods and the mesh local optimization methods. We will briefly explain the first three techniques; for more details, the reader is referred to [Frey 01], and [George 99]. We then introduce the MTC topological mesh generator, a local optimization algorithm, that is developed at CEMEF by Coupez [Coupez 91] and Gruau [Gruau 05] and show its advantages over the classical methods.

### 2.2.1 The three classical mesh generation algorithms

- *Advancing front methods*: The advancing front methods were introduced in [Lo 85]. Starting from the boundary, elements are constructed one by one forming a front, a border, between the meshed and the unmeshed parts of the domain. New points are created and connected to the existing elements hence forming new elements and moving the front forward. Note that the points are created in such a way that the new elements do not intersect with already existing ones. The process repeats iteratively advancing the front until the whole domain is meshed and the front vanishes. These methods produce high quality elements at the domain boundary, but the quality might deteriorate as the front advances causing convergence problems. Thus some nodes need to be removed and the process should be restarted. Ensuring the elements' good quality and hence the convergence of the method is a difficult task especially for a complicated three dimensional geometries.
- *Delaunay method*: First introduced by Delaunay [Delaunay 34], the Delaunay triangulation satisfies the Delaunay property. The latter states that in the two di-

dimensional case the circumscribed disk of a triangle should not enclose any vertex. In the three dimensional case, no vertex is enclosed by the circumscribing sphere of a tetrahedron. The first step in generating a Delaunay triangulation consists in generating a non-overlapping Voronoi tessellation from the set of nodes on the discrete domain boundary. The second step corresponds to joining all the nodes belonging to two adjacent Voronoi cells. This triangulation is unique and the elements recover the convex hull of the set of nodes belonging to the boundary. For more details the reader is referred to [Frey 01, George 99, Delaunay 34]. Delaunay meshing algorithm starts with a Delaunay triangulation of the domain using only the nodes of its boundary mesh. This mesh is then refined by inserting new nodes inside the domain. The elements whose circumscribed disk contains these nodes will be removed and new elements are generated with the surfaces of the removed elements while respecting the Delaunay property hence updating the mesh topology. Note that the new nodes are placed far from existing ones so as to prevent the creation of short edges. Finally the elements that lie outside the domain are removed. The main drawback of the Delaunay method is that the Delaunay property is not a mesh quality criterion. The algorithm can create degenerate elements with very bad qualities especially near the boundaries of the domain. Therefore an optimization step is generally employed to improve the elements' quality.

- *Quadtree/octree methods*: Introduced in 1983, the grid meshers [Yerry 83] cover a domain with a background Cartesian mesh (a rectangular one in 2D and a box in 3D). The method consists first in recursively refining the cells that contain several boundary nodes until each element contains at most one node. The refinement is done by adding points to the intersection of two adjacent cells and to the intersection of the boundary with a cell. Then cells that lie outside the domain are removed and the other ones are triangulated. Again the triangular cells outside the domain boundary are removed, thus obtaining the domain triangulation. Although these mesh generators are robust, fast, and easy to implement and parallelize, the generated elements tend to be aligned in a certain direction influencing the solvers' solutions and posing the difficulty of constructing anisotropic meshes. Another disadvantage of these methods is their tendency to create poor quality elements near the boundary.

### 2.2.2 The topological optimization mesh generation algorithm

The three above mentioned classical meshing techniques require an additional step of optimization in order to improve the quality of the generated elements. The MTC mesher that we use in our laboratory, was first developed by T. Coupez in [Coupez 91] then extended to generate anisotropic meshes in [Coupez 00]. Starting from an initial random meshing of the domain, the MTC mesher proceeds by iterative local optimization of the mesh topology based on the minimal volume criterion and the

mesh geometric quality. In this work we retrace the steps of the method development. For more information the reader is referred to the following publications [Coupez 91, Coupez 96, Coupez 00, Gruau 05, Boussetta 06].

Rather than building up a mesh that satisfies optimal criteria, the proposed method is based on the improvement of an existing mesh. A surface and volume meshing are coupled through the use of a virtual boundary elements layer.

Before we proceed into the outline of the method let us present some useful definitions and notations.

**Definition 2.** A **d-simplex** is the convex hull of its  $d + 1$  vertices. For instance a triangle is a 2-simplex and a tetrahedron is a 3-simplex.

**Definition 3.** Consider a domain  $\Omega \subset \mathbb{R}^d$ . Denote by  $\mathcal{N}$  a finite set of nodes in  $\Omega$  and  $\mathcal{T}$  a set of  $d$ -simplexes generated from the  $\mathcal{N}$  nodes. Let  $\mathcal{F}(\mathcal{T})$  be the set of  $\mathcal{T}$ 's elements' faces. We say that  $\mathcal{T}$  is a **mesh topology** of  $\Omega$  if and only if:

- (i)  $\forall F \in \mathcal{F}(\mathcal{T}), 1 \leq \text{card} \{T \in \mathcal{T} \text{ s.t. } F \in \partial T\} \leq 2.$
- (ii) The couple  $(\mathcal{N}, \partial \mathcal{T})$  is a mesh of the boundary  $\partial \Omega$ .

In other words, each  $F \in \mathcal{F}(\mathcal{T})$  is shared by at most two elements of  $\mathcal{T}$ , with one of these elements having a face on the boundary of  $\Omega$ .

A local mesh modification of a mesh topology is a cut/paste operation whereby a subset of elements  $\mathcal{T}_A$  is replaced by another subset  $\mathcal{T}_B$ . In such an operation  $\mathcal{T}_B$  represents a mesh topology having the same boundary as  $\mathcal{T}_A$ .

The topological meshing algorithm consists in considering a patch at a time with a patch being the set of elements sharing a node or an edge. The generation of new mesh topologies preserving the boundary of the domain is done using the “starring” operator. This operator involves connecting one node  $S$  to the boundary faces that do not contain it.

Denote by  $\mathcal{N}(\mathcal{T}_A)$  and  $\mathcal{F}(\mathcal{T}_A)$  respectively the set of nodes and the set of faces present in the mesh topology  $\mathcal{T}_A$  and by  $\mathcal{B}(\mathcal{T}_A)$  the centroid of the nodes on  $\partial \mathcal{T}_A$ . Let  $S \in \mathcal{N}(\mathcal{T}_A) \cup \mathcal{B}(\mathcal{T}_A)$  be a node, the “starring” operator applied to the node  $S$  in the mesh topology  $\mathcal{T}_A$  generates the following mesh topology:

$$\mathcal{T}^*(S, \partial \mathcal{T}_A) = \{T \text{ s.t } T = \{\{S\} \cup F\}, F \in \partial \mathcal{T}_A, F \in \mathcal{F}(\mathcal{T}_A) \text{ and } S \notin F\}.$$

In other words, the new topology is made up of the triangles formed by joining a node  $S$  to the boundary faces that do not contain it.

Note that, as mentioned earlier, the cut/paste operation should be performed on mesh topologies. Therefore we need to make sure that the patch created with the “starring” operator is indeed a mesh topology. For that purpose, we introduce the adherence of  $A$ :

$$\bar{A} = \{T \in \mathcal{T}, T \subset \mathcal{T}_A\}.$$

Finally the cut/paste operation is given by:

$$Q(\mathcal{T}) = \mathcal{T} - \overline{\mathcal{T}_A} + \underbrace{\mathcal{T}_B}_{\mathcal{T}^*(S, \partial\mathcal{T}_A)} .$$

The boundary patches are given special treatment, as sketched in figure 2.1. In order to remove a node  $S$  from the boundary, first, a layer of virtual elements is constructed by inserting a virtual node  $O$ , the duplicate of  $S$ , and connecting the nodes of boundary elements to it leading to the creation of new virtual  $d$ -simplexes, represented by the dashed lines in figure 2.1. Hence the nodes and faces on the boundary can be moved or deleted and new nodes can also be added. The mesh optimization algorithm consists

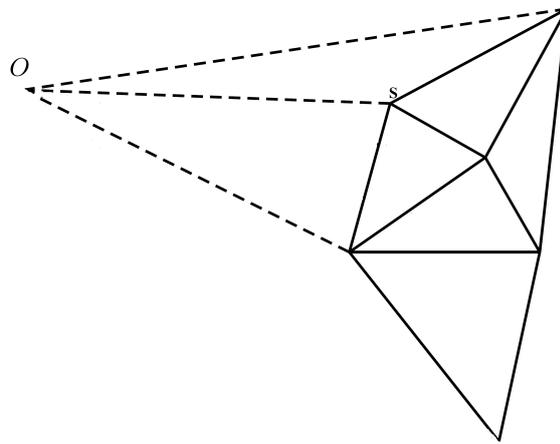


Figure 2.1 – Virtual elements connected to a virtual node outside the domain.

of iterative local mesh improvements:

$$\mathcal{T}^{k+1} = Q(\mathcal{T}^k).$$

The process repeats until no significant improvement is detected.

Now that we have presented the concept of the local mesh optimization process two questions remain to be answered: the first one is how do we choose the nodes  $S$  and the second is how do we define mesh topology improvement.

### 2.2.2.1 Selection criteria for optimal local mesh topology

There are two selection criteria for generating the optimal local mesh topology: the minimum volume principle and the geometric quality criterion. While the former ensures the conformity of the elements in the mesh, the latter handles the optimization of their shapes.

- **The minimum volume principle:** We consider a mesh topology  $\mathcal{T}_A \subset \mathcal{T}$ . According to the minimum volume principle, the optimal mesh topology generated

by the “starring” operator and improving the quality of  $\mathcal{T}_A$  is given by:

$$\mathcal{T}_B = \arg \min_{\mathcal{T}} \sum_{T \in \mathcal{T}} |T|, \quad \text{where } \mathcal{T} = \{ \mathcal{T}^*(S, \partial \mathcal{T}_A), \text{ s.t. } S \in \mathcal{N}(\mathcal{T}_A) \cup \mathcal{B}(\mathcal{T}_A) \},$$

where  $|T|$  denotes the volume of the element  $T$ . The minimization is done over the set of all possible mesh topologies generated by the “starring” operator for all the nodes in  $\mathcal{T}_A$ . Note that for the patches on the boundary, the volume of the virtual elements is not counted in the total volume of the mesh topology. This selection criterion guarantees that the candidate mesh topology is conformal with no overlapping between the elements. However a mesh topology satisfying the minimum volume principle might not be unique. Thus choosing the best among these mesh topologies requires another selection criterion.

- **The geometrical quality criterion:** We evaluate the quality of each element in the candidate mesh topologies by:

$$Q(T) = \frac{|T|}{h_T^d},$$

where  $d$  is the space dimension and  $h_T$  represents the mean of the edge lengths. Then the worst elements in the candidate mesh topologies are compared. The geometrical quality principle selects the mesh topology having the best quality among these elements, therefore handling the optimization of elements’ shapes.

### 2.2.2.2 Mesh generation algorithm: local topology optimization

Now that we have well defined the mesh local optimization procedure we can explain how to generate a mesh of a domain given its boundary mesh.

The first step consists in generating a first mesh topology  $\mathcal{T} = \mathcal{T}^*(S_0, \partial \Omega)$  of  $\Omega$ , where  $S_0$  is a node on the boundary  $\partial \Omega$ . This is done by joining, with a “starring” operator, one node to all the faces that do not contain it. Obviously the result of this step is not yet a mesh but it is a starting mesh topology. Recursive local mesh improvements are then applied to optimize the mesh topology according to algorithm 1.

We note that, during the local optimization process, new nodes can be added to the list of nodes and previous nodes can be removed from it. Figure 2.2 shows an example of such a process.

We consider a mesh topology  $\mathcal{T}_A$  around the node  $S$  with boundary

$$\partial \mathcal{T}_A = \{S_1\} \cup \{S_2\} \cup \{S_3\} \cup \{S_4\} \cup \{S_5\}.$$

Using the “starring” operator, several candidate configurations for improving the topology can be generated:

- mesh topology around  $S$  (eliminating node  $S_6$ ).

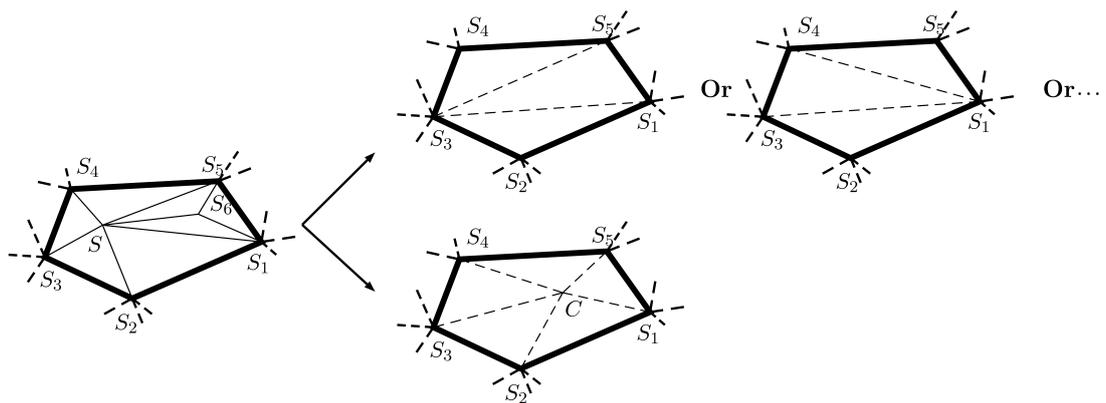
**Algorithm 1:** Local optimization of a mesh topology algorithm**Input:**  $(\mathcal{N}, \partial\mathcal{T})$  a mesh topology of  $\Omega$ .**Output:**  $(\mathcal{N}_{\text{opt}}, \partial\mathcal{T}_{\text{opt}})$  the optimal mesh topology.**begin****while**  $\sum_{T \in \mathcal{T}} |T| \geq |\Omega|$  *i.e. the volume of the mesh topology is not optimal, do***for** *each node and edge in*  $\mathcal{T}$  **do**Remove the local mesh topology  $\mathcal{T}_A$  associated with the node/edge.;Replace it with a new local mesh topology  $\mathcal{T}_B = \mathcal{T}^*(S_A, \partial\mathcal{T}_A)$  thatminimizes the volume  $\sum_{T \in \mathcal{T}_B} |T|$  and maximizes the elements' qualities $Q_T = \frac{|T|}{h_T^d}$ , with  $S_A$  being a node in  $\mathcal{T}_A$  and  $h_T$  being the average length of element  $T$ 's edges.Update  $\mathcal{T} = \bigcup_B \mathcal{T}_B$ .

Figure 2.2 – Example on the local optimization process by the ‘starring’ operator, adopted from [Coupez 00].

- mesh topology around any of the nodes  $S_1, S_2, S_3, S_4$  or  $S_5$  (eliminating the nodes  $S$  and  $S_6$ ).
- mesh topology around the centroid  $C$  (eliminating the nodes  $S$  and  $S_6$ ).

### 2.2.2.3 Mesh generation responding to a mesh size map

Once a first mesh of  $\Omega$  is generated, we move on to address another problematic: *how does the topological mesher respond to a given mesh size map during the process of meshing or remeshing?*

We consider a mesh size map defined on the nodes of the mesh. The mesh size on an element is then deduced by averaging the imposed mesh sizes at the nodes sharing this element.

The mesh optimization process involves the comparison and minimization between the imposed and the actual mesh sizes,  $h_T^{\text{imp}}$  and  $h_T^{\text{act}}$  respectively. For that purpose the geometric quality factor is changed to:

$$Q_T(h_T^{\text{act}}, h_T^{\text{imp}}) = Q_T(h_T^{\text{act}}) \frac{\min(|T|, |T_{h_T^{\text{imp}}}|)}{\max(|T|, |T_{h_T^{\text{imp}}}|)},$$

with  $|T_{h_T^{\text{imp}}}|$  being the volume of the element having  $h_T^{\text{imp}}$  edge lengths. No modification is applied to the actual mesh generation algorithm; the only difference is the use of this quality factor instead of the previously presented one.

### 2.2.2.4 Advantages/disadvantages of the local mesh optimization algorithm compared to the other mesh generation techniques

Although the local mesh optimization algorithm is more expensive than the other techniques as it involves a recursive looping over mesh topologies, it is easy to implement and parallelize, works in  $4D$  and can easily be used for mesh adaptation throughout computations. The topological operations are robust and less susceptible to rounding errors than the other mentioned methods. It provides a good treatment of the boundary elements using the virtual elements technique.

## 2.3 AMR: ADAPTIVE MESH REFINEMENT

The concept of mesh adaptation roots back to the 1970s [Allgower 78] with the desire to advance in numerical simulations and establish good qualitative results. Since then intensive effort was employed to develop adaptive mesh refinements that would enable on one hand accurate numerical solutions and on the other hand faster computations [Lo 90, Craig 87, Adjerdid 86, Berger 84]. Error estimators were proposed as indicators for mesh refinement with respect to a global or a local degree of accuracy. In that regard, successful isotropic mesh adaptation methods were developed

[Arney 90, Berger 84, Bieterman 86, Chew 89]. An isotropically adapted mesh is generated by constructing regular equilateral elements while varying the mesh size between regions of the domain according to error estimates. However this method tends to concentrate too many elements at the locations of high solution gradient especially when the solution exhibits anisotropic features i.e. its variations in one direction are more significant than in the other directions. To overcome this computational limitation for applications with directional features such as shock waves, shear and boundary layers in fluid dynamics, researchers developed an adaptation technique, the anisotropic mesh adaptation, that is capable of providing a full control on the element size, shape and orientation. The latter method employs elements with high aspect ratios, hence allowing a better representation of solution variations.

Anisotropic mesh adaptation was first proposed in the late 1980s [Peraire 87, Selmin 92, Löhner 89, Zienkiewicz 94] with the moving front technique in the vision of generating elements with directional features. Despite the success of this method, its first application to 3D problems showed limited stretching capabilities. Over the same period, the author in [Mavriplis 90] proposed a heuristic approach for building Delaunay triangulations that generates high aspect ratio elements. He suggested modifying the definition of the distance on the planes of the mesh elements' circumscribed disk/sphere.

The approach of generating highly stretched elements in a locally mapped space using metric tensors was first introduced for viscous flows and applied on Delaunay meshes [Mavriplis 90, George 91, Castro-Díaz 97]. The challenges posed by such a method involve defining the sizes and directions of stretching without generating obtuse elements. To avoid such elements, the minimum-maximum principle was suggested [Barth 92, Marcum 95]. However one drawback of this approach is due to the high sensitivity of the connectivity to the vertices' distributions. A proposed alternative solution to the above difficulties was to perform local re-meshing.

Significant research effort has been devoted in the last few years to devise powerful anisotropic mesh adaptation techniques with real applications. We distinguish four major error estimates for anisotropic adaptation: the hessian based relying on the solution's hessian information to evaluate the linear interpolation error [Tam 00, Pain 01, Bottasso 04, Gruau 05, Frey 05, Boussetta 06], the a posteriori estimates approximating the discretization error using a theoretical analysis [Kunert 00, Formaggia 03, Picasso 03, Formaggia 04, Micheletti 06, Picasso 06, Hecht 14], the a priori error estimates [Formaggia 01, Huang 05] and the goal oriented estimates that provide mathematical framework for assessing the quality of some functionals [Venditti 03, Micheletti 08a, Alauzet 10b, Peter 12, Yano 12]. All these methods have been well consolidated leading to a standardization of the adaptation process. First, an error analysis enables the localization of the solution's high gradient. This step is followed by the construction of a directional metric tensor that is provided to the mesh generator. In turn, the latter produces an appropriate mesh that is optimized to match the provided metric tensor.

Anisotropic mesh adaptation has proved to be a powerful asset for improving the efficiency of finite element/volume methods. It presents advantages over the isotropic refinement techniques by reducing storage requirements and computational times. It enables the capture of scale heterogeneities that can appear in numerous physical problems including those having boundary layers, shock waves, edge singularities and moving interfaces.

In this work, we intend to develop a metric based mesh adaptation that is capable of well capturing the anisotropy of a physical phenomenon.

### 2.3.1 Techniques of mesh adaptation

There are three main methods for mesh adaptation:

- **The h-adaptation method:** involves local mesh modifications such as edge flipping, local mesh refinement in the locations of high solution's gradient and local mesh coarsening where the solution exhibits slow variations. The method tends to equi-distribute the numerical error over the discretized domain. It provides the mesh generator with a mesh size map that is produced based on an error estimation on the current mesh. In this work, we will be using this technique to adapt the mesh.
- **The r-adaptation method:** relocates the nodes of a mesh without affecting its topology nor the connectivity and the number of its nodes. The nodes are moved to regions where the solution undergoes rapid changes. This method is well suited for problems with moving boundaries. However it is very expensive and has a slow convergence rate.
- **The p-adaptation method:** involves the enrichment of local elements' degrees of interpolation while conserving the topology of the mesh. The main drawback of this method resides in the difficulty of employing it with already existing solvers especially when it comes to solving incompressible flows. In fact, when solving incompressible problems with a mixed formulation, the elements used must verify the *Brezzi-Babuska* compatibility condition. Furthermore it is hard to determine mixed and compatible elements for each degree of interpolation.

A combination between the h-adaptation and the p-adaptation is sometimes employed. The method is called the hp-adaptation and aims at getting a higher precision in the most efficient way. The choice between refinement or local change in the degree of interpolation is yet not trivial and has been the subject of many research studies.

## 2.4 METRIC-BASED ANISOTROPIC MESH ADAPTATION

### 2.4.1 State of the art

As stated earlier, the objective inhere is to build a metric tensor that prescribes optimal mesh sizes and elements' orientations. The procedure involves evaluating an error over each edge of the domain using the projection of the reconstructed gradient of the solution on this edge. The developed algorithm strives to improve the quality of the aforementioned error estimates by attempting to reduce and equi-distribute the error over the edges of the mesh. The approach is based on reducing the interpolation error of a chosen scalar or vector or a combination of several fields. Using a symmetric positive definite tensor called '*metric*', a mapping from the Euclidean to a Riemannian space is employed. An error equi-distribution in the latter space permits the computation of stretching factors and elements' orientations. Therefore leading to the generation of an anisotropic and well adapted mesh with high aspect ratio elements in the former space. The process is repeated until no significant change in the mesh is detected.

Before presenting the derivation of the developed adaptation technique, let us provide some preliminary definitions and the theoretical framework for the rest of the chapter.

### 2.4.2 Preliminary definitions and theoretical framework

This subsection presents the basic definitions that are useful for the development of our mesh adaptation process. The notions of Riemannian space, metric tensor, and unit mesh are recalled.

Let  $\Omega \subset \mathbb{R}^d$  be a physical domain discretized using a finite element mesh  $\Omega_h = \bigcup_{K \in \mathcal{K}} K$ , with  $K$  being a  $d$ -simplex (segment, triangle, tetrahedron,  $\dots$ ). We denote by:

$$\mathbf{X} = \{X^i \in \mathbb{R}^d, i = 1, \dots, N\}$$

the set of nodes in the mesh.

**Definition 4.** We call a **metric**  $\mathcal{M} \subset \mathbb{R}^d$  a  $d \times d$  symmetric positive definite matrix.

**Proposition 5.** A metric  $\mathcal{M}$  can be diagonalized into:

$$\mathcal{M} = \mathcal{R} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{bmatrix} \mathcal{R}^t$$

where  $\mathcal{R}$  is a rotation matrix whose columns are the eigenvectors of  $\mathcal{M}$  and  $(\lambda_i)_{i=1, \dots, d} \geq 0$  are the corresponding eigenvalues.

We define next the *metric space* framework with the associated notions of edge

length, element volume, and unit balls. For that purpose, let us recall the definitions of these notions in the *Euclidean* canonical space.

In the *Euclidean* space, lengths of vectors are defined by means of the canonical Euclidean scalar product (with respect to the identity matrix  $\mathcal{I}_d$ ):

$$\begin{aligned} (\cdot, \cdot)_{\mathcal{I}_d} : \mathbb{R}^d \mathbb{R}^d &\mapsto \mathbb{R}^+ \\ (u, v) &\mapsto (u, v)_{\mathcal{I}_d} = u^t \mathcal{I}_d v. \end{aligned}$$

In this space, the distance between two points  $X^i$  and  $X^j$  is determined by:

$$\mathcal{D}_{\mathcal{I}_d}(X^i, X^j) = \left( \overrightarrow{X^i X^j}^t \mathcal{I}_d \overrightarrow{X^i X^j} \right)^{\frac{1}{2}} = \left( \sum_{c=1}^d \overrightarrow{X^i X^j}_c^2 \right)^{\frac{1}{2}}$$

and the volume of an element  $K$  is  $|K|_{\mathcal{I}_d}$ . We recall also the definition of a unit ball in the Euclidean space:

**Definition 6.** A **unit ball** in the Euclidean space  $\mathcal{B}_{\mathcal{I}_d}$  is the set of vectors whose lengths are less than or equal to 1 with respect to  $\mathcal{I}_d$  i.e.

$$\mathcal{B}_{\mathcal{I}_d} = \left\{ x \in \mathbb{R}^d, \text{ s.t. } x^t \mathcal{I}_d x \leq 1 \right\}$$

The scalar product of 2 vectors can also be defined in other geometrical spaces called *metric spaces* whereby the scalar product induced by a symmetric positive definite matrix is defined by:

$$\begin{aligned} (\cdot, \cdot)_{\mathcal{M}} : \mathbb{R}^d \mathbb{R}^d &\mapsto \mathbb{R}^+ \\ (u, v) &\mapsto (u, v)_{\mathcal{M}} = u^t \mathcal{M} v. \end{aligned}$$

Hence the distance between two points  $X^i$  and  $X^j$  is determined by:

$$\mathcal{D}_{\mathcal{M}}(X^i, X^j) = \left( \overrightarrow{X^i X^j}^t \mathcal{M} \overrightarrow{X^i X^j} \right)^{\frac{1}{2}},$$

and the volume of an element  $K$  is  $|K|_{\mathcal{M}} = \sqrt{\det(\mathcal{M})} |K|_{\mathcal{I}_d}$ . Therefore the definition of a unit ball in the metric space follows:

**Definition 7.** A **unit ball** in the metric space  $\mathcal{B}_{\mathcal{M}}$  is the set of vectors whose lengths are less than or equal to 1 with respect to  $\mathcal{M}$  i.e.

$$\mathcal{B}_{\mathcal{M}} = \left\{ x \in \mathbb{R}^d, \text{ s.t. } x^t \mathcal{M} x \leq 1 \right\}$$

The unit ball is thus delimited by an ellipsoid with axis the eigenvectors  $\mathcal{B}_i$  and sizes  $h_i^2 \lambda_i = 1$  i.e.  $h_i = \frac{1}{\lambda_i}^{\frac{1}{2}}$  along these directions, as shown in figure 2.3. Consequently, a linear mapping between the unit ball in the Euclidean canonical space and the metric space can be deduced and is depicted in figure 2.3.

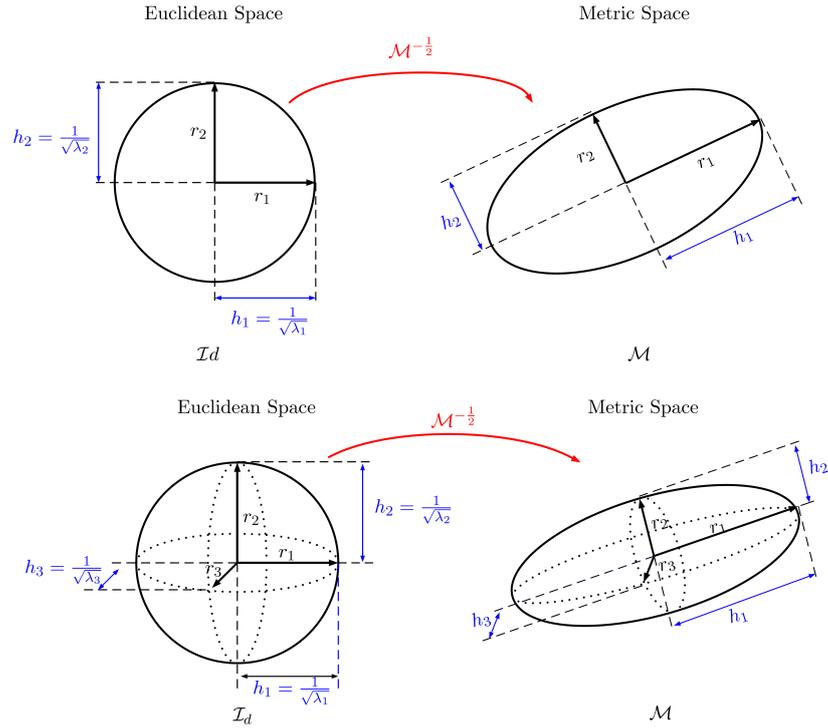


Figure 2.3 – Unit ball and linear mapping between the metric space and the canonical space in two and three dimensional spaces.

When the metric is constant for all the nodes in the mesh, the notion of lengths will be defined in the same way for all the edges of the mesh. We can as well define a metric per element in the mesh. It is also possible to have for each node  $X^i$  a metric that defines the concepts of scalar product related to that specific node taking into account all the previously mentioned definitions. We can now define a **Riemannian metric space** as a real smooth manifold where we can define metric tensors  $\mathcal{M}$  so that to each node  $X^i$  in the discrete mesh is associated a metric with its induced inner product  $(\cdot, \cdot)_{\mathcal{M}_{X^i}}$ :

$$\forall X^i \in \mathbf{X}, \exists \mathcal{M}_{X^i} \text{ s.t.}, (\cdot, \cdot)_{\mathcal{M}_{X^i}} : \mathbb{R}^d \mathbb{R}^d \mapsto \mathbb{R}^+$$

$$\overrightarrow{(X^i X^j)}, \overrightarrow{(X^i X^j)}_{\mathcal{M}_{X^i}} = \overrightarrow{X^i X^j} \mathcal{M}_{X^i} \overrightarrow{X^i X^j}$$

Now that we have introduced the Riemannian metric framework with its associated definition of lengths, we move on to explain how can metric fields be used for mesh adaptation.

### 2.4.3 Metric based mesh adaptation

The keystone idea of anisotropic, metric based, mesh adaptation is to generate a **unit mesh** (with unit edge lengths and regular elements) in a prescribed Riemannian space as stated in [George 91]. Since it is almost impossible to achieve unity for all the edges, we intend to construct a mesh that is optimal (very close to unity) up to a certain tolerance.

As we have mentioned earlier, the main feature of anisotropic mesh adaptation techniques is their control on the orientation of the elements and their sizes. This is exactly the information that is brought by metric tensors. The metric map will be provided to the mesh generator assigning the mesh sizes and the directions along which the latter are prescribed. Indeed the mesh sizes are determined from the eigenvalues of the metric  $h_i = \frac{1}{\lambda_i}^{\frac{1}{2}}$  and the directions are their corresponding eigenvectors.

**Remark 8.** *A metric with the same eigenvalues:*

$$\mathcal{M} = \mathcal{R} \Lambda \mathcal{R}^t = \mathcal{R} \begin{bmatrix} \frac{1}{h(x)^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{h(x)^2} \end{bmatrix} \mathcal{R}$$

*is a special case of metric tensors and leads to isotropic mesh adaptation.*

During the remeshing process, given a metric  $\mathcal{M}$  that is defined on the elements of the mesh, the mesh generator tries to create conformal elements satisfying the properties prescribed by the metric field. The resulting mesh will be anisotropic and well adapted in the Euclidean space. Note that all the computations are now performed in the underlying Riemannian space induced by the metric.

As discussed in the previous section, the process of mesh adaptation goes through a series of mesh optimization along which elements are removed and being replaced. Thus the metric field is interpolated from one mesh topology to another one leading to a slight diffusion of the results. Therefore since the elements are much more volatile than the mesh nodes, defining fields on a continuous basis eases their reconstruction, interpolation or extrapolation.

#### 2.4.3.1 Metric construction at the elements' level

A unit metric field can be associated with any unstructured mesh. Given a non-degenerate element  $K$ , we can define a metric  $\mathcal{M}_K$  such that  $K$  is a unit element in the Riemannian space associated with  $\mathcal{M}_K$ . Indeed, applying an affine transformation  $T_K$  from the physical space  $(\mathbb{R}^d, \mathcal{I}^d)$  to the unit ball of the Euclidean metric space, mapping element  $K$  to a reference unit element  $\hat{K}$ , we can determine  $\mathcal{M}$  as follows:

$$\begin{aligned} K &\xrightarrow{T_K} \hat{K} \\ \mathbf{X} &\xrightarrow{T_K} \hat{\mathbf{X}} + \mathbb{A}_K(\mathbf{X} - \hat{\mathbf{X}}). \end{aligned}$$

This linear mapping would transform each edge in  $K$  into a unit edge in  $\hat{K}$  i.e.

$$1 = |\mathbb{A}_K \mathbf{X}_{ij}| = ({}^t \mathbb{A}_K \mathbb{A}_K \mathbf{X}_{ij}, \mathbf{X}_{ij})^{\frac{1}{2}}.$$

Subsequently, the metric tensor on the element  $K$  can be identified as:

$$\mathcal{M}_K = {}^t \mathbb{A}_K \mathbb{A}_K.$$

$\mathcal{M}_K$  is thus a constant tensor that is uniquely defined on the element  $K$ . Repeating the same construction process for all the elements in the mesh, we obtain a piecewise constant tensor field with element to element discontinuities.

### 2.4.3.2 Difficulties faced by element-wise metric construction

From a practical standpoint for mesh adaptation, it is easier to store the metric at the nodes rather than at the elements. Theoretically speaking, a good representation of the Riemannian space can be obtained by constructing a continuous metric field on the entire computational domain which enables a continuous measurement of length variations at each point and direction [Coupez 09]. Nevertheless, in order to obtain a continuous metric, the discontinuous element tensor field must be transferred to the mesh nodes. For that purpose, an interpolation stage should be performed on the underlying Riemannian space. Ambiguities arise regarding the extrapolation of the information at each element level to any point of the domain.

Indeed when the metric field is defined at the element level of the mesh, one needs to recover the discontinuities from one element to the other by interpolating the field on the nodes. Several interpolation schemes were proposed in the literature. Nevertheless these techniques are usually non-commutative i.e. the resulting metric on the whole domain highly depends on the order of interpolation when several metrics are involved [Loseille 09].

### 2.4.3.3 Overcoming these difficulties

In this thesis, we adopt a different approach for metric construction. It consists in building the metric field directly at the nodes of the mesh. It is derived from the data available at the nodes of the mesh without direct resort to element information nor considering any underlying interpolation. This construction is done using a statistical concept: the length distribution tensor. Inspired from a standard idea used in computation rheology to account for fiber orientation [Advani 87], we introduced a second order tensor that reconstructs the values at the nodes by gathering information from the surrounding edges. Note that the length distribution tensor can be defined for any space dimension. The proposed metric will be derived from the inverse of this tensor that is defined on each node of the mesh from the surrounding edges' contributions. Another contribution is focused on how to modify the edge length in this framework.

### 2.4.4 Metric construction at the nodes' level

As a starting point, we intend to build a natural metric field that is continuous on the domain. As stated earlier, the construction of a unit mesh with anisotropic features

consists in generating elements with unit edges in the prescribed Riemannian space. The same target is adopted for building nodal metrics. Let  $\Gamma(i) = \{\mathbf{X}^{ij}, X^j \in X\}$  be the set of edges in the mesh connected to node  $X^i$ . The aim is to define a metric on  $X^i$  such that the length of each edge  $\mathbf{X}^{ij}$  connected to  $X^i$  equals one with respect to the metric field  $\mathcal{M}^i$ . An averaging process is used to construct a unique tensor at  $X^i$  that gathers all the data defined on the edges sharing this node. Hence, we are seeking a field  $\mathcal{M}^i$  defined on  $X^i$  such that the lengths of all edges in  $\Gamma(i)$  are one. Consider a linear transformation changing  $\mathbf{X}^{ij}$  into a unit length vector:

$$1 = |\mathbb{A}_{ij}\mathbf{X}_{ij}| = ({}^t\mathbb{A}_{ij}\mathbb{A}_{ij}\mathbf{X}_{ij}, \mathbf{X}_{ij})^{\frac{1}{2}}, \quad \forall j \in \Gamma(i) \quad (2.1)$$

Taking  $\mathcal{M}^i = {}^t\mathbb{A}_{ij}\mathbb{A}_{ij}$ , we get:

$$(\mathcal{M}^i\mathbf{X}_{ij}, \mathbf{X}_{ij}) = 1, \quad \forall j \in \Gamma(i) \quad (2.2)$$

Then summing up equation (2.2) over  $j$  we get:

$$\sum_{j \in \Gamma(i)} (\mathcal{M}^i\mathbf{X}_{ij}, \mathbf{X}_{ij}) = \sum_{j \in \Gamma(i)} 1 = |\Gamma(i)|. \quad (2.3)$$

Moreover, using the Einstein notation for tensor scalar product  $A : B = A_{ij}B_{ij}$ , equation (2.3) can be equivalently written as:

$$\mathcal{M}^i : \left( \sum_{j \in \Gamma(i)} \mathbf{X}_{ij} \otimes \mathbf{X}_{ij} \right) = |\Gamma(i)| \quad (2.4)$$

This problem of finding the unknown natural metric  $\mathcal{M}^i$  solution of (2.3) is in general over-determined. In practice it is very hard to construct edges of exact unit length. Hence, this problem can be relaxed into an optimization problem over the space of squared symmetric tensors  $\mathbb{R}_{\text{sym}}^{d \times d}$ :

$$\mathcal{M}^i = \underset{\mathcal{M}^i \in \mathbb{R}_{\text{sym}}^{d \times d}}{\text{argmin}} \left( \sum_{j \in \Gamma(i)} \left( \|\mathbf{X}^i - \mathbf{X}^j\|_{\mathcal{M}^i}^2 - 1 \right)^2 \right), \quad (2.5)$$

where  $\mathcal{M}^i$  should be a symmetric positive definite tensor. However solution  $\mathcal{M}^i$  to this problem may fail to be positive definite (for example when the neighbors of  $X^i$  are aligned on a hyperbola). To deal with this constraint, we resort to the construction of orientation tensors inspired from the works done in rheology [Advani 87]. Before we proceed into the derivation of the metric tensor, let us explore the source of inspiration for the construction of distribution tensors.

#### 2.4.4.1 Analogy with rheology

The mechanical properties of composite materials of a polymer, a metal or a ceramic depend highly on the fiber orientation. The latter is calculated using a mathematical model based on orientation tensors. The simulation of mold filling flow and the prediction of fiber orientation were of great interest for these applications. Thanks to the distribution tensor one can predict the composite's stiffness and anisotropic fiber suspensions due to the flow induced fiber alignment in its direction. The method has its merits in accurately predicting the properties of fiber orientation state. The only prerequisite information is the properties defined on the uniaxial fiber composite which are provided by the finite element computations. Let  $p_j$  be a sample of directions. The second and fourth order distribution tensors on a node  $X^i$  read respectively as:

$${}^2\mathbb{X}^i = \frac{\sum_{j \in \Gamma(i)} \omega_j p_j \otimes p_j}{\sum_{j \in \Gamma(i)} \omega_j}, \quad (2.6)$$

and

$${}^4\mathbb{X}^i = \frac{\sum_{j \in \Gamma(i)} \omega_j p_j \otimes p_j \otimes p_j \otimes p_j}{\sum_{j \in \Gamma(i)} \omega_j}, \quad (2.7)$$

where  $\sum_{j \in \Gamma(i)}$  represents the sum over possible fiber orientations around  $X^i$ ,  $\omega_j$  is a probability distribution function defined on the surface of the unit sphere and describing the orientation of the fiber and  $\otimes$  represents the outer product operator.

#### 2.4.4.2 Construction of the length distribution tensor

From this statistical concept, we retain the orientation features reflected by the second order directional tensor in order to construct the length distribution tensors at the nodes of the mesh. So for every node  $X^i$  in the mesh and every edge orientation  $p_j$ , we define the length distribution function as  $\mathcal{L}(X^i, p_j) \in \mathbb{R}^+$  such that every point  $X^j$  connected to  $X^i$  and in the direction  $p_j$  can be expressed as:

$$X^j = X^i + \mathcal{L}(X^i, p_j) p_j.$$

Without loss of generality, we choose  $X^j$  such that  $|X^{ij}| = 1$ . Proceeding by analogy with (2.6), we define the second order length distribution tensor at node  $X^i$  as:

$${}^2\mathbb{X}^i = \frac{\sum_{j \in \Gamma(i)} \mathcal{L}(X^i, p_j) p_j \otimes p_j}{\sum_{j \in \Gamma(i)} \mathcal{L}(X^i, p_j)}. \quad (2.8)$$

Since  $\mathbf{X}^{ij}$  is in the direction of  $p_j$  then equation (2.8) becomes:

$${}^2\mathbb{X}^i = \frac{\sum_{j \in \Gamma(i)} \mathcal{L}(X^i, \mathbf{X}^{ij}) \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}}{\sum_{j \in \Gamma(i)} \mathcal{L}(X^i, \mathbf{X}^{ij})}.$$

Note that, from the definition of node  $X^j$  and the fact that the length size of the edge  $\mathbf{X}^{ij}$  is equal to the interpolation of the length distribution function at  $X^j$ , we have  $\mathcal{L}(X^i, \mathbf{X}^{ij}) = 1$ , and  $\sum_j \mathcal{L}(X^i, \mathbf{X}^{ij}) = \sum_{j \in \Gamma(i)} 1 = |\Gamma(i)|$ . Consequently,

$${}^2\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}. \quad (2.9)$$

Using the same analysis, we can derive the fourth order length distribution tensor:

$${}^4\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \otimes \mathbf{X}^{ij}. \quad (2.10)$$

**Remark 9.**  ${}^2\mathbb{X}^i$  and  ${}^4\mathbb{X}^i$  are symmetric positive definite tensors, and thus invertible, if at least  $d$  edges are not collinear.

**Remark 10.** The choice of equally distributing all the nodes connected to  $X^i$  comes from the choice of unit length distribution function values at the points  $X^j$ .

This length distribution tensor will be the skeleton for the metric tensor construction.

### 2.4.4.3 Metric construction from the length distribution tensor

In this subsection we will introduce a natural solution to the optimization problem and show in which sense this solution presents a good approximation of the natural solution of equation (2.3) following the lines in [Coupez 11].

**Proposition 11.** Let  ${}^2\mathbb{X}^i$  be the second order symmetric positive definite length distribution tensor at node  $X^i$ . Then the metric

$$\mathbb{M}^i = \frac{1}{d} \left( \mathbb{X}^i \right)^{-1} \quad (2.11)$$

is a good approximation of the natural metric transforming the edges in the mesh into unit edges.

The proposition and its proof were provided in [Coupez 11], we retrace them hereafter. We start by showing that the tensor defined by  $\mathbb{M}^i = \frac{1}{d} \left( \mathbb{X}^i \right)^{-1}$  is indeed a metric at the node  $X^i$ . We provide in the next proposition an analytical example in 2D that proves via simple algebraic computations the validity and robustness of our metric construction technique.

**Proposition 12.** Consider  $\underline{u}, \underline{v} \in \mathbb{R}^2$  two non collinear vectors. Then  $\mathcal{M} = (\underline{u} \otimes \underline{u} + \underline{v} \otimes \underline{v})^{-1}$  is a unit metric for  $\underline{u}$  and  $\underline{v}$  with  $(\mathcal{M}\underline{u}, \underline{u}) = 1$  and  $(\mathcal{M}\underline{v}, \underline{v}) = 1$ .

*Proof.* The length distribution tensor associated with  $\underline{u} = (u_1, u_2)^t$  and  $\underline{v} = (v_1, v_2)^t$  is given by:

$$\mathbb{X} = \frac{1}{2}(\underline{u} \otimes \underline{u} + \underline{v} \otimes \underline{v}) = \frac{1}{2} \begin{pmatrix} u_1^2 + v_1^2 & u_1u_2 + v_1v_2 \\ u_1u_2 + v_1v_2 & u_2^2 + v_2^2 \end{pmatrix}$$

We aim at proving that the tensor

$$\mathcal{M} = \frac{2}{2}({}^2\mathbb{X})^{-1} = ({}^2\mathbb{X})^{-1}$$

is a metric. So we need to show that  $\mathcal{M}$  is symmetric positive definite and that it satisfies the unity property.

$\mathcal{M}$  is symmetric positive definite

$$\mathcal{M} = \frac{1}{(u_1v_2 - u_2v_1)^2} \begin{pmatrix} u_2^2 + v_2^2 & -(u_1u_2 + v_1v_2) \\ -(u_1u_2 + v_1v_2) & u_1^2 + v_1^2 \end{pmatrix}$$

- Since  $\mathcal{M}_{12} = \mathcal{M}_{21}$  then  $\mathcal{M}$  is symmetric.
- Let  $\underline{x} = (x_1, x_2)^t$  be a vector in  $\mathbb{R}^2$  then

$$\begin{aligned} \underline{x}^t \mathcal{M} \underline{x} &= \frac{1}{(u_1v_2 - u_2v_1)^2} (x_1 \ x_2) \begin{pmatrix} u_2^2 + v_2^2 & -(u_1u_2 + v_1v_2) \\ -(u_1u_2 + v_1v_2) & u_1^2 + v_1^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\ &= \frac{1}{(u_1v_2 - u_2v_1)^2} \left( \begin{pmatrix} x_1(u_2^2 + v_2^2) - x_2(u_1u_2 + v_1v_2) \\ -x_1(u_1u_2 + v_1v_2) + x_2(u_1^2 + v_1^2) \end{pmatrix}^t \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \right) \\ &= \frac{1}{(u_1v_2 - u_2v_1)^2} \\ &\quad \times (x_1^2(u_2^2 + v_2^2) - x_1x_2(u_1u_2 + v_1v_2) - x_1x_2(u_1u_2 + v_1v_2) + x_2^2(u_1^2 + v_1^2)) \end{aligned}$$

Rearranging the terms, we get:

$$\begin{aligned} \underline{x}^t \mathcal{M} \underline{x} &= \frac{1}{(u_1v_2 - u_2v_1)^2} ((x_1^2u_2^2 - 2x_1x_2u_1u_2 + x_2^2u_1^2) + (x_1^2v_2^2 - 2x_1x_2v_1v_2 + x_2^2v_1^2)) \\ &= \frac{1}{\underbrace{(u_1v_2 - u_2v_1)^2}_{\geq 0}} \left( \underbrace{(x_1u_2 - x_2u_1)^2}_{\geq 0} + \underbrace{(x_1v_2 - x_2v_1)^2}_{\geq 0} \right) \end{aligned}$$

Then  $\mathcal{M}$  is positive definite.

$\mathcal{M}$  is a unit metric.

We will show that  $\mathcal{M}(\underline{u}, \underline{u}) = 1$ , the proof that  $\mathcal{M}(\underline{v}, \underline{v}) = 1$  is similar.

$$\begin{aligned} \mathcal{M}(\underline{u}, \underline{u}) &= \frac{1}{(u_1v_2 - u_2v_1)^2} \begin{pmatrix} u_2^2 + v_2^2 & -(u_1u_2 + v_1v_2) \\ -(u_1u_2 + v_1v_2) & u_1^2 + v_1^2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\ &= \frac{(u_1v_2 - u_2v_1)^2}{(u_1v_2 - u_2v_1)^2} \\ &= 1 \end{aligned}$$

Therefore  $\mathcal{M} = (\underline{u} \otimes \underline{u} + \underline{v} \otimes \underline{v})^{-1}$  is a unit metric.  $\square$

The next proposition generalizes the conjecture in proposition 12 adopted from [Coupez 11] into the  $d$ -dimensional space.

**Proposition 13.** Consider  $\{\underline{v}_j\}_{1 \leq j \leq d}$  to be a set of  $d$  linearly independent vectors in  $\mathbb{R}^d$ .

Then  $\mathcal{M} = (\sum_j^d \underline{v}_j \otimes \underline{v}_j)^{-1}$  is a unit metric for  $\underline{v}_j$  with  $(\mathcal{M}\underline{v}_j, \underline{v}_j) = 1 \forall j \in [0, d]$ .

*Proof.* The proof follows using the same reasoning carried out in the proof of proposition 12.  $\square$

**Proposition 14.** Let  ${}^2\mathbb{X}^i$  and  ${}^4\mathbb{X}^i$  be respectively the second and fourth order length distribution tensors at a node  $X^i$ . Then the metric  $\mathcal{M}^i$  is a well defined solution of the optimization problem

$$\mathcal{M}^i = \underset{\mathcal{M}^i \in \mathbb{R}_{sym}^{d \times d}}{\operatorname{argmin}} \left( \sum_{j \in \Gamma(i)} \left( (\mathcal{M}\mathbf{X}^{ij}, \mathbf{X}^{ij}) - 1 \right)^2 \right) \quad (2.12)$$

if it is a solution of:

$${}^4\mathbb{X}\mathcal{M} = {}^2\mathbb{X}. \quad (2.13)$$

*Proof.* Denote by  ${}^4\mathbb{X}_{klmn}$  the  $(klmn)$  component of the fourth order length distribution tensor:

$${}^4\mathbb{X}_{klmn} = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} X_k^{ij} X_l^{ij} X_m^{ij} X_n^{ij}$$

Let  $\mathcal{M}^i$  be a solution of equation (2.13). Using the above notation, we compute the component  $(mn)$  of  $({}^4\mathbb{X}\mathcal{M})$ :

$$({}^4\mathbb{X}^i \mathcal{M}^i)_{mn} = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \left( \underbrace{\sum_{kl} \mathcal{M}_{kl}^i X_k^{ij} X_l^{ij}}_{(\mathcal{M}^i \mathbf{X}^{ij}, \mathbf{X}^{ij})} \right) X_m^{ij} X_n^{ij}$$

From equation (2.13), one gets:

$$\begin{aligned}
({}^4\mathbb{X}^i \mathcal{M}^i)_{mn} &= {}^2\mathbb{X}_{mn}^i \\
\Leftrightarrow \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \left( \sum_{kl} (\mathcal{M}_{kl}^i \mathbf{x}_k^{ij} \mathbf{x}_l^{ij}) \right) \mathbf{x}_m^{ij} \mathbf{x}_n^{ij} &= \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{x}_m^{ij} \mathbf{x}_n^{ij} \\
\Leftrightarrow \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \left( \left( \sum_{kl} (\mathcal{M}_{kl}^i \mathbf{x}_k^{ij} \mathbf{x}_l^{ij}) \right) \mathbf{x}_m^{ij} \mathbf{x}_n^{ij} - \mathbf{x}_m^{ij} \mathbf{x}_n^{ij} \right) &= 0 \\
\Leftrightarrow \sum_{j \in \Gamma(i)} \left( \left( \sum_{kl} (\mathcal{M}_{kl}^i \mathbf{x}_k^{ij} \mathbf{x}_l^{ij}) \right) - 1 \right) \mathbf{x}_m^{ij} \mathbf{x}_n^{ij} &= 0 \tag{2.14}
\end{aligned}$$

On the other hand applying the standard differentiation of (2.12) and noting that at the minimum the derivative is zero, we get:

$$0 = \sum_{j \in \Gamma(i)} \left( \mathcal{M}_{kl}^i \mathbf{x}_k^{ij} \mathbf{x}_l^{ij} - 1 \right) \mathbf{x}_m^{ij} \mathbf{x}_n^{ij} \quad \text{for } 1 \leq m, n \leq d \tag{2.15}$$

which is verified by the metric  $\mathcal{M}^i$  as shown in equation (2.14). This finishes the proof of the proposition.  $\square$

To complete the proof of proposition 11 we still need to verify that the metric  $\left( \mathcal{M} = \frac{|\Gamma(i)|}{d} ({}^2\mathbb{X})^{-1} \right)$  also satisfies condition (2.13) and is consequently a solution of the optimization (2.12). This is done in the next proposition.

**Proposition 15.** *Let  $\mathcal{M}^i$  be a metric defined by:*

$$\mathcal{M} = \frac{|\Gamma(i)|}{d} ({}^2\mathbb{X})^{-1}$$

then  $\mathcal{M}^i$  is a solution of:

$${}^4\mathbb{X} : \mathcal{M} = {}^2\mathbb{X},$$

under the assumption that:

$${}^4\mathbb{X} \sim {}^2\mathbb{X} \otimes {}^2\mathbb{X} \tag{2.16}$$

*Proof.* From a numerical standpoint it is hard to construct the fourth order length distribution tensor. Its inversion is even more complicated as it requires the consideration of more edges than the second order tensor. Instead, we use a closure quadratic approximation of the fourth order tensor [Advani 90] in the view of computing the solution of (2.13):

$${}^4\mathbb{X} \sim {}^2\mathbb{X} \otimes {}^2\mathbb{X} \tag{2.17}$$

Substituting equation (2.17) in (2.13), we obtain:

$$({}^2\mathbb{X} \otimes {}^2\mathbb{X}) : \mathcal{M} = {}^2\mathbb{X} \tag{2.18}$$

Rewriting the left hand side of (2.18) using the Einstein operator, implies:

$$({}^2\mathbb{X} \otimes^2 \mathbb{X}) : \mathcal{M} = (\mathbb{X} : \mathcal{M})^2 \mathbb{X} \quad (2.19)$$

We note that for a symmetric positive definite tensor  $A$  of order  $d$  we have  $A : A^{-1} = d$ . Proceeding by analogy between (2.18) and (2.19) yields:

$${}^2\mathbb{X} : \mathcal{M} = 1 \quad (2.20)$$

and thus we obtain the construction of a symmetric positive definite tensor

$$\mathcal{M} = \frac{1}{d} ({}^2\mathbb{X})^{-1} \quad (2.21)$$

Indeed the equality holds when replacing  $\mathcal{M}$  in (2.19) and rearranging the terms:

$$\frac{1}{d} \left( {}^2\mathbb{X} : ({}^2\mathbb{X})^{-1} \right)^2 \mathbb{X} = {}^2 \mathbb{X}$$

□

Applying this analysis on every node  $X^i$  in the mesh and identifying  ${}^2\mathbb{X}$  by the length distribution tensor defined at this node:

$${}^2\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij},$$

the metric associated with  $X^i$

$$\mathcal{M}^i = \frac{1}{d} ({}^2\mathbb{X})^{-1} \quad (2.22)$$

will be the solution of (2.13) and in turn will be a solution to the optimization problem (2.5). In other words,  $\mathcal{M}^i$  defines in a least square sense a unit metric at node  $X^i$ .

So far we have set up the Riemannian space framework and recalled the essential notions related to them along with the dual mapping between the Riemannian and the Euclidean spaces. A definition of unit metric was provided and the aim of metric based mesh adaptation for constructing unit meshes was stated. In the next section we intend to derive metric fields defined at the nodes of the mesh such that all the edges have lengths close to unity with respect to the metrics. In this section we have emphasized how metric fields can be derived in a natural way to prescribe mesh sizes and orientations for adapting the mesh. The length distribution tensor was introduced reflecting the distribution of edges' lengths and their orientation over the mesh. The key motivation behind the use of orientation tensor fields to describe edges' repartition and connections resides in the computational ease of constructing and manipulating them. The definition of unit metric was relaxed via a least square approximation. A sufficient condition guaranteeing that the constructed metric is a solution of the least squares problem was provided. Finally an approximate solution metric satisfying the sufficient condition was derived as a function of the length distribution tensor.

In the following section we address the question of how to repartition the anisotropy and prescribe the mesh sizes in a way to allow a better capture of a certain field of interest (*e.g.* an analytical function, a problem variable, a characteristic field,  $\dots$ ).

## 2.5 ERROR ESTIMATION FOR ANISOTROPIC MESH ADAPTATION

When working with stationary problems, the objective of mesh adaptation is to construct a mesh that is optimal for computing the stationary solution with a good accuracy. The goal of anisotropic mesh adaptation is to ensure that the estimated discrepancy between the exact and the approximate solution is minimal. For that purpose, the mesh density shall be increased locally at the regions of high approximation error so that to provide the fidelity necessary to accurately predict the solution.

In this section we are interested in building a metric tensor that modifies the mesh sizes in all directions in a way to provide a good capture of a physical entity. The work has lead to several publications [Coupez 11, Coupez 13b]. We present all the details and proofs of the derivation. The idea relies on devising an error estimator that predicts the regions of high solution variations. A metric field is deduced from this refining/coarsening indicator giving a better representation of the mesh at the locations of interest.

### 2.5.1 Overview on error estimation techniques

Efforts have been ongoing to explore error estimation techniques in an a priori and an a posteriori sense. The idea was first proposed in [Babuška 78]. Recent developments based on anisotropic interpolation estimates have proved to be successful for generating meshes with high aspect ratio [Zienkiewicz 92, Frey 05, Gruau 05, Dedé 08, Bourgault 09].

The a priori estimators require the knowledge of the exact solution and provide qualitative information about the problem at hand. They proved to be powerful for convergence studies of numerical methods [Formaggia 01, Loseille 09]. On the other hand, the a posteriori error estimators rely on the approximated solution of the problem. We distinguish four widespread branches of a posteriori error estimators: the residual based, the hierarchical, the recovery based and the goal oriented methods.

The residual based error estimators [Demkowicz 84, Bank 85], as suggested by their name, are related to the residual of the discrete solution. Although they are easy to implement, they lack precision due to successive approximations in their derivation. The hierarchical error estimators [Bank 93, Eijkhout 91] involve considering a high order solution as a good representation of the exact solution and perform all computations with respect to it. Since these methods require two solutions for different orders they render high computational costs. Moreover, the recovery based error estimators [Zienkiewicz 87, Zienkiewicz 92, Micheletti 06] intend to bound the  $H_1$ -semi norm of the discretization error  $\|\nabla u - \nabla u_h\|_{L^2}$  via substituting

$\nabla u$  with an appropriate approximation. However the gradient recovery approaches described in [Zienkiewicz 87, Zienkiewicz 92] induce a considerable computational cost. The last well known method of error estimation is the goal-oriented technique [Becker 96, Oden 99, Formaggia 03, Venditti 03, Alauzet 12]. Significant effort has been devoted for developing goal-oriented error estimations that provide a mathematical framework for assessing the quality of some functional with particular relevance. These methods consist in exploiting the dual (or adjoint) problem whose source term is the quantity of interest. It is worth mentioning that different choices of functional generate different adapted meshes and only the solution features that are relevant to this functional will be controlled.

### 2.5.2 Global optimization problem

As stated earlier, in this section, we develop a simple gradient recovery technique allowing the computation of an edge-based error estimator. Using the error equi-distribution principle and under the constraint of a fixed number of nodes in the mesh, a set of stretching factors associated with the edges of the domain are determined. A simple modification of the length distribution tensor to take into account these stretching factors will result in a new metric leading to an anisotropic and well-adapted mesh.

We consider a scalar field  $u \in \mathbf{C}^2(\overline{\Omega}) = \mathcal{V}$  and  $\mathcal{V}_h$  a simple  $P^1$  finite element approximation space:

$$\mathcal{V}_h = \left\{ w_h \in \mathbf{C}^0(\overline{\Omega}), w_h|_K \in P^1(K), K \in \mathcal{K} \right\}$$

where  $\Omega = \bigcup_{K \in \mathcal{K}} K$  and  $K$  is a simplex (segment, triangle, tetrahedron, ...).

We define

$$\mathbf{X} = \left\{ X^i \in \mathbb{R}^d, i = 1, \dots, N \right\}$$

as the set of nodes in the mesh. We denote also by  $U^i$  the nodal value of  $u$  at  $X^i$  and we let  $\Pi_h$  be the Lagrange interpolation operator from  $\mathcal{V}$  to  $\mathcal{V}_h$  such that:

$$\Pi_h u(X^i) = u(X^i) = U^i, \forall i = 1, \dots, N.$$

As shown in figure 2.4, the set of nodes connected to node  $X^i$  is denoted by

$$\Gamma(i) = \left\{ j \text{ s.t. } \exists K \in \mathcal{K}, X^i, X^j \text{ are nodes of } K \right\}.$$

In the context of mesh adaptation the aim is to prescribe mesh sizes and directions to the nodes of the mesh in order to control the approximation error. In general the approximated solution is not interpolating i.e. it does not necessarily coincide with the exact solution at the mesh nodes. Consequently it is very difficult to quantify this error as it is very dependent on the problem at hand. We recall that our objective is to derive a mesh adaptation technique that can be automatically applied to predict any

phenomenon without apriori knowledge of the model that is being solved. To avoid the difficulties encountered when working with the approximation error, we resort to the *Lemma of Cea* which states that the approximation error is bounded by the interpolation error [Cea 64, Ciarlet 91]:

$$\|u - u_h\|_{H^1} \leq C \|u - \Pi_h u\|_{H^1}, \quad (2.23)$$

where  $C$  is a constant related to the space dimension.

The lemma was first proposed for elliptic problems. A similar inequality for the  $L_\infty$  norm was derived for Euler equations [Koobus 05]. Lately an apriori analysis was proposed by Belme ([Belme 11] chapter 6) for the Poisson equation and the Euler problem. We intend in chapter 3 to extend this analysis to the convection-diffusion-reaction problem and the Navier-Stokes equations.

Starting from the assumption that for metric based adaptation methods, in order to master the global approximation error it is sufficient to control the interpolation error, the target can be summarized into finding the mesh, made up of at most  $N$  nodes, that minimizes the linear interpolation error in the  $L^p$  norm.

The mesh adaptation process can thus be reformulated as a constrained optimization problem whereby one intends to minimize the global interpolation error over the domain of interest for a fixed number of nodes  $N$ . We seek a metric field  $\widetilde{\mathcal{M}}$  defined at the nodes of the non-adapted mesh and at the origin of an optimal well adapted one  $\mathcal{H}(\widetilde{\mathcal{M}})$ :

$$\widetilde{\mathcal{M}} = \underset{\mathcal{M} \in \mathbb{R}_{\text{sym}}^{d \times d}}{\text{argmin}} \|u - \Pi_h u\|_{L^p(\Omega_h)} \quad \text{such that} \quad \sum_i 1 = N, \quad (2.24)$$

where  $i$  is the index number of the nodes in  $\mathcal{H}(\widetilde{\mathcal{M}})$ . In this framework  $\Pi_h$  is closely related to the unit mesh  $\mathcal{H}(\widetilde{\mathcal{M}})$  of the discretized domain.

### 2.5.3 Gradient and strong continuity along the edges

In what follows, we denote the interpolated solution by  $u_h = \Pi_h u$ . For the sake of simplicity, we introduce the following notations for a generic node  $X^i$ :

$$\mathbf{X}^{ij} = X^j - X^i \quad \text{and} \quad U^{ij} = U^j - U^i \quad \forall j \in \Gamma(i).$$

The gradient of  $u_h$  is a piecewise constant vector field discontinuous from element to element. However its projection onto the edges is continuous and depends only on the nodal values of  $u$  at the extremities of the edges. This is proved by considering a Taylor series expansion about the node  $X^i$ :

$$U^j = U^i + \nabla u_h|_K \cdot \mathbf{X}^{ij}.$$

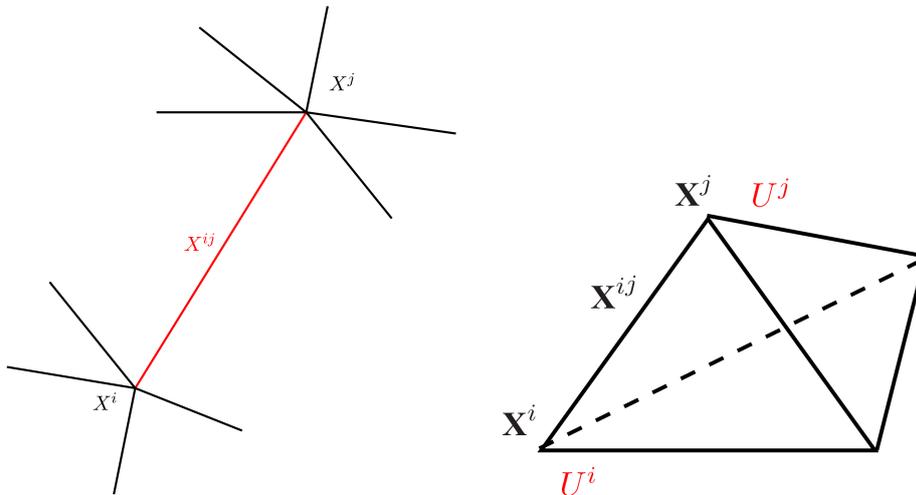


Figure 2.4 – Length  $\mathbf{X}^{ij}$  of the edge joining nodes  $X^i$  and  $X^j$ .

Rearranging the terms, we get:

$$\nabla u_h|_K \cdot \mathbf{X}^{ij} = U^{ij}. \quad (2.25)$$

Since this equality is true for all the elements sharing the edge  $X^i X^j$  then the gradient projection of the interpolated field along  $X^i X^j$  can be expressed as the difference between the scalar field values at the extremities of the edge.

$$\nabla u_h \cdot \mathbf{X}^{ij} = U^{ij}. \quad (2.26)$$

From the continuity property we can deduce that the interpolation error along the edges is of second order for  $P_1$  elements as will be demonstrated in the following proposition [Coupez 11, Coupez 13b].

**Proposition 16.** *The error in the projected gradients of the exact and interpolated solutions is bounded by the hessian of the solution.*

$$\left| \underbrace{\nabla u_h \cdot \mathbf{X}^{ij}}_{U^{ij}} - \nabla u(\mathbf{X}^i) \cdot \mathbf{X}^{ij} \right| \leq \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|, \quad (2.27)$$

where  $\mathbb{H}(u)(\mathbf{Y}) = \nabla^{(2)}u(\mathbf{Y})$  is the associated Hessian of  $u$  evaluated at a generic point  $\mathbf{Y}$ . Recall that taking  $u \in \mathbb{C}^2(\bar{\Omega})$  we obtain  $\nabla u \in \mathbb{C}^1(\bar{\Omega})$ .

*Proof.* The projection of a smooth scalar field  $u$  along an edge  $\mathbf{X}^{ij}$  still shows high regularity. Indeed since  $u|_{\mathbf{X}^{ij}} \in \mathbb{C}^2(\mathbf{X}^{ij})$  then  $\nabla u|_{\mathbf{X}^{ij}} \in \mathbb{C}^1(\mathbf{X}^{ij})$ . Let  $x \in [X^i, X^j]$ :

$$u(x) = u(X^i + sX^i X^j) \quad s \in [0, 1].$$

The mean value theorem implies that:

$$\exists A^{ij} \in [X^i, X^j], \text{ such that } U^j = U^i + \nabla u(A^{ij})\mathbf{X}^{ij}.$$

Equivalently,

$$\exists A^{ij} \in [X^i, X^j], \text{ such that } U^{ij} = \nabla u(A^{ij})\mathbf{X}^{ij}. \quad (2.28)$$

As the gradient is constant along an edge, then combining equations (2.26) and (2.28) yields:

$$\nabla u_h \mathbf{X}^{ij} = \nabla u(A^{ij})\mathbf{X}^{ij}.$$

Moreover, since  $A^{ij} \in [X^i, X^j]$ , it can be written as:

$$A^{ij} = X^i + s_{ij}\mathbf{X}^{ij}.$$

A subsequent use of the mean value theorem to the  $C^1$  function  $\nabla u|_{\mathbf{X}^{ij}}$  on  $[X^i, A^{ij}]$  yields:

$$\exists B^{ij} \in [X^i, A^{ij}], \text{ such that } \left( \nabla u(A^{ij}) - \nabla u(X^i) \right) \cdot \mathbf{X}^{ij} = s_{ij} \mathbb{H}(u)(B^{ij})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.29)$$

Then combining (2.28) and (2.29) we get:

$$|U^{ij} - \nabla u(\mathbf{X}^i)| \cdot \mathbf{X}^{ij} = |s_{ij} \mathbb{H}(u)(B^{ij})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \quad (2.30)$$

Now, using equation (2.26), equation (2.30) reads as:

$$|\nabla u_h - \nabla u(\mathbf{X}^i)| \cdot \mathbf{X}^{ij} = |s_{ij} \mathbb{H}(u)(B^{ij})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \quad (2.31)$$

Therefore, noting that  $s_{ij} \in [0, 1]$ , the statement of the proposition follows:

$$\left| \underbrace{\nabla u_h \cdot \mathbf{X}^{ij}}_{U^{ij}} - \nabla u(\mathbf{X}^i) \cdot \mathbf{X}^{ij} \right| \leq \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|,$$

□

Most of the anisotropic metric-based mesh adaptation techniques present in the literature intend to recover the unknown hessian of the solution at the mesh vertices in view of building the optimal metric field. The authors in [Agouzal 99] derived sufficient conditions for approximating the discrete hessian  $\nabla_{\mathcal{R}_h}^2 u_h$ . From this approximated hessian a metric  $\mathcal{M}_{\nabla_{\mathcal{R}_h}^2}$  is constructed resulting in a mesh that is quasi-uniform with respect to it and that will be also quasi-uniform with respect to a metric that is obtained from the exact hessian  $\mathbb{H}(u)(x)$ . However when the initial mesh used for recovery is far from the optimal one, the discrete hessian is not necessarily a good representation of the continuous one. To overcome this difficulty, the process of generating the mesh and re-building the hessian is iteratively repeated until getting a quasi-optimal mesh.

Other methods were proposed in the literature and aim at recovering the solution's

hessian from the piecewise linear approximate solution defined at the vertices of the mesh. We will recall briefly the three most commonly used ones:

### A recovery based on the projection methods

The Zienkiewicz-Zhu (ZZ) projection method [Zienkiewicz 92] is used on  $P_1$  finite element spaces. The gradient is constant on the elements and discontinuous from one element to the other. In order to recover the gradient at the nodes  $X^i$  of the mesh, a patch consisting of the elements sharing the node  $X^i$  is constructed. Then a linear function is defined fitting in a least square sense the gradient values at the centers of mass of the elements in the patch. The reconstruction is repeated twice in order to recover the discrete hessian of the solution:

$$\nabla_{\mathcal{R}_h} u_h(X^i) = \Pi_{ZZ}(\nabla(\Pi_{ZZ}(\nabla u_h))), \quad (2.32)$$

where  $\Pi_{ZZ}$  is the Zienkiewicz-Zhu (ZZ) projector on the  $P_1$  finite element space.

The  $L^2$  projection operator is derived from the Clément operator [Claisse 11, Alauzet 10c]. The  $L^2$  projection method reconstructs the gradient by means of a volume-weighted average. It is done in two steps: first a gradient recovery step at node  $X^i$  whereby the recovered gradient is given by:

$$\nabla_{\mathcal{R}_h} u_h(X^i) = \frac{\sum_{K_j \in \mathcal{B}(X^i)} |K_j| \nabla u_h|_{K_j}}{\sum_{K_j \in \mathcal{B}(X^i)} |K_j|} = \frac{\sum_{K_j \in \mathcal{B}(X^i)} |K_j| \sum_{P \in K_j} u_h(P) \nabla w_{K_j}^P}{\sum_{K_j \in \mathcal{B}(X^i)} |K_j|}, \quad (2.33)$$

with  $P$  being a node of element  $K_j$ ,  $w_{K_j}^P$  its barycentric coordinates and  $\mathcal{B}(X^i)$  is the patch of elements sharing node  $X^i$ . Then a subsequent recovery is applied to each component of the recovered gradient in order to get a hessian reconstruction:

$$\nabla_{\mathcal{R}_h}^2 u_h(X^i) = \frac{\sum_{K_j \in \mathcal{B}(X^i)} |K_j| \nabla^2 u_h|_{K_j}}{\sum_{K_j \in \mathcal{B}(X^i)} |K_j|} = \frac{\sum_{K_j \in \mathcal{B}(X^i)} |K_j| \left( \sum_{P \in K_j} \Pi_h(u_h)(P) \right) \left( \nabla w_{K_j}^P \right)^t}{\sum_{K_j \in \mathcal{B}(X^i)} |K_j|}. \quad (2.34)$$

However, both projection techniques do not remove high frequency errors introduced by small non-uniformities in the mesh. Another drawback of these techniques comes from the accumulation of diffusion through the repetitive averaging process.

### A recovery based on the variational methods [Alauzet 10c]

The recovery based on the variational methods relies on the fact that for  $P_1$  scalar field  $u_h$ , the gradient is constant by element. Hence, the recovery of the hessian is done by using a weak formulation and the Green formula. It starts by building a higher

order approximation of the solution  $u^*$  and considering it as a good representation of  $u$ . Each component of the reconstructed hessian is recovered by:

$$\frac{\partial^2_{\mathcal{R}_h} u_h}{\partial x_i \partial x_j}(P) = \frac{\sum_{K_j \in \mathcal{B}(P)} \left( \frac{\partial u_h}{\partial x_j} \right) |_K \int_K \frac{\partial \phi_K}{\partial x_j}}{\frac{\mathcal{B}(P)}{4}}, \quad (2.35)$$

where  $\phi_P \in \mathcal{V}_h$  is the piecewise linear  $P_1$  finite element function associated with the vertex  $P$ , and  $\mathcal{V}_h$  is the approximation space associated with the  $P_1$  finite element method.

### A recovery based on a least square approach [Picasso 11]

The aim of the recovery method based on a least square approach is to determine the recovered hessian  $\nabla^2_{\mathcal{R}_h} u_h$  at a node  $X^i$ . We define  $\mathcal{B}(X^i)$  as the patch of elements sharing node  $X^i$ . We consider a node  $X^j \in \mathcal{B}(X^i)$ . A Taylor development of  $\nabla^2_{\mathcal{R}_h} u_h$  at  $X^j$  about  $X^i$  truncated at second order yields:

$$\frac{1}{2} X^i X^j \nabla^2_{\mathcal{R}_h} u_h(X^i) X^i X^j = u_h(X^j) - u_h(X^i) - X^i X^j \nabla_{\mathcal{R}_h} u_h(X^i) \quad (2.36)$$

This can be rewritten as a system:

$$AH = B,$$

where  $H$  is a vector made up of the components of the hessian matrix.

The system is solved using a least square approximation. Notice that this approach involves the resolution of a system at each node in the mesh and each remeshing step and hence affects the computational time of the simulation.

During adaptive iterations, the number of topological optimizations of the mesh grows and the hessian recovery becomes more and more complex. The actual complexity is in fact problem dependent.

The major advantage of the anisotropic mesh adaptation technique that we propose is that it avoids the hessian reconstruction. Indeed using the bounding relation (2.27) we can express the projected hessian of the solution ( $\mathbb{H}(u)(\mathbf{Y})\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}$ ) in terms of the projected gradient which is reconstructed using the solution values at the edge extremities and the length distribution tensor.

We recall that  $u|_{\mathbf{X}^{ij}} \in \mathcal{C}^2(\mathbf{X}^{ij})$  then  $\nabla u|_{\mathbf{X}^{ij}} \in \mathcal{C}^1(\mathbf{X}^{ij})$ . A Taylor series development of the gradient of  $u$  leads to:

$$\nabla u(X^j) = \nabla u(X^i) + \mathbb{H}(u)(X^i)\mathbf{X}^{ij}. \quad (2.37)$$

Then, projecting onto  $\mathbf{X}^{ij}$ , we get:

$$\left(\nabla u(X^j) - \nabla u(X^i)\right) \cdot \mathbf{X}^{ij} = \mathbb{H}(u)(X^i) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \quad (2.38)$$

We denote by  $\mathbf{g}^i = \nabla u(X^i)$  and  $\mathbf{g}^j = \nabla u(X^j)$  the gradient of  $u$  evaluated at the node  $X^i$  and  $X^j$ , respectively. We denote as well by  $\mathbf{g}^{ij} = \mathbf{g}^j - \mathbf{g}^i$  the change of the gradient along the edge  $\mathbf{X}^{ij}$ . Then

$$\mathbf{g}^{ij} \cdot \mathbf{X}^{ij} = \mathbb{H}(u)(X^i) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.39)$$

For  $P_1$  numerical methods, we use this projection as an evaluation of second order interpolation errors along the edge:

$$e_{ij} = |u - u_h|_{\mathbf{X}}^{ij} = |\mathbf{g}^{ij} \cdot \mathbf{X}^{ij}|. \quad (2.40)$$

However this equation cannot be evaluated exactly as it requires the knowledge of  $\nabla u$  and its continuity at the nodes of the mesh.

In cases where the exact solution of the PDE  $u$  is unknown, the only information readily available on its gradient comes from the finite element approximation. This means that for a  $P_1$  finite element method we do not have access to the point-wise information but only to the element-wise one. For this reason we resort to a recovery procedure.

#### 2.5.4 Least square gradient recovery

The gradient of the piecewise linear function  $u_h$  is well-defined at the elements' interiors. A smoothing should be applied to construct nodal gradient values. A number of gradient recovery techniques can be found in the literature. We have presented (eqs. (2.32) and (2.33)) the gradient recovery techniques that are based on projection methods, nevertheless these methods necessitate a complex implementation to achieve super-convergence properties.

Polynomial preserving techniques [Zhang 05, Zhang 04] can also be used for gradient recovery. They consist of fitting high order polynomials, in the least-squares sense, to the known approximate solution at the nodes and subsequently differentiating them. More precisely, for a node  $X^i$  assuming that we want to construct a quadratic fitting, at least 5 neighboring nodes need to be used. The polynomial is derived by a least squares fitting to the values of the approximate solution at these nodes. The problem reduces to a least squares one. If the latter is not full rank, additional nodes should be considered and the fitting procedure is repeated. The recovery procedure might become more and more computationally complex especially when working with anisotropic meshes.

The gradient reconstruction that we propose is based on a least squares approximation of  $\nabla u_h$  along the edges of the mesh. Using the length distribution tensor, a continuous gradient will be defined directly at the node of the mesh and depending only on the solution's interpolation values. For each node  $X^i$ , we seek  $\mathbf{G}^i \in \mathbb{R}^d$  the

proposed gradient reconstruction satisfying:

$$\mathbf{G}^i = \operatorname{argmin}_{\mathbf{G} \in \mathbb{R}^d} \sum_{j \in \Gamma(i)} \left| (\mathbf{G} - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2 = \operatorname{argmin}_{\mathbf{G} \in \mathbb{R}^d} \sum_{j \in \Gamma(i)} \left| \mathbf{G} \mathbf{X}^{ij} - U^{ij} \right|^2. \quad (2.41)$$

The minimum can be obtained by setting the derivative of (2.41) to zero i.e.

$$\begin{aligned} \sum_{j \in \Gamma(i)} \left( \mathbf{G}^i \mathbf{X}^{ij} - U^{ij} \right) \mathbf{X}^{ij} &= 0 \\ \iff \mathbf{G}^i \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} &= \sum_{j \in \Gamma(i)} U^{ij} \mathbf{X}^{ij} \end{aligned} \quad (2.42)$$

We highlight the appearance of the length distribution tensor. Introducing the notation

$$\mathbf{U}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} U^{ij} \mathbf{X}^{ij},$$

one gets:

$$\mathbf{G}^i = \left( \mathbb{X}^i \right)^{-1} \mathbf{U}^i. \quad (2.43)$$

Note that the gradient recovery technique that we propose is directly computed on the nodes of the mesh and the only requirement for its implementation is the knowledge of the approximate solution at the nodes.

**Remark 17.** We recall that  $\mathbb{X}^i$  is a positive definite tensor if the set of edges connected to  $X^i$ ,  $\Gamma(i)$ , contains at least  $d$  non-collinear vectors. Therefore the reconstructed gradient  $\mathbf{G}^i$  is defined under this assumption. Note that this condition is satisfied if at least one non degenerate element exists at the node  $X^i$ .

**Remark 18.**  $\mathbf{G}^i$  can also be expressed in terms of the natural metric defined at node  $X^i$ :

$$\mathbf{G}^i = d\mathcal{M}^i \mathbf{U}^i. \quad (2.44)$$

Therefore, the approximated error is evaluated by substituting  $\mathbf{g}$  by  $\mathbf{G}$  in (2.40):

$$e_{ij} = \left| \mathbf{G}^{ij} \cdot \mathbf{X}^{ij} \right|. \quad (2.45)$$

## 2.5.5 A posteriori error analysis

We want to verify that the choice of the edge-based error estimation defined by equation (2.45) preserves the second order:  $\left| \left( \mathbf{G}^i - \mathbf{g}^i \right) \cdot \mathbf{X}^{ij} \right| \sim (\mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij})$  where  $\mathbf{G}^i$  is the recovery gradient given by (2.43) and  $\mathbf{g}^i$  is the exact value of the gradient at node  $X^i$ . The proof will be done in the sequel.

In the next proposition, we provide error estimates in the neighborhood of a node  $X^i$ .

**Proposition 19.** Let  $u \in \mathbf{C}^2(\overline{\Omega})$ . Then for  $\mathbf{G}^i$  defined as in (2.43), we have the following projected error bounds:

$$\left( \sum_{j \in \Gamma(i)} \left| (\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij} \right|^2 \right)^{\frac{1}{2}} \leq 2 \left( \sum_{j \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|^2 \right)^{\frac{1}{2}}. \quad (2.46)$$

*Proof.* Since  $\mathbf{G}^i$  is defined by:

$$\mathbf{G}^i = \operatorname{argmin}_{\mathbf{G} \in \mathbb{R}^d} \sum_{j \in \Gamma(i)} \left| (\mathbf{G} - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2,$$

then

$$\sum_{j \in \Gamma(i)} \left| (\mathbf{G}^i - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2 \leq \sum_{j \in \Gamma(i)} \left| (\nabla u - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2. \quad (2.47)$$

The error between the reconstructed gradient and the true gradient of the approximate solution can be split into two components:

$$\left| (\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij} \right| = \left| (\mathbf{G}^i - \nabla u_h + \nabla u_h - \nabla u) \cdot \mathbf{X}^{ij} \right|$$

Applying this decomposition, together with inequality (2.47) yields:

$$\begin{aligned} \sum_{j \in \Gamma(i)} \left| (\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij} \right|^2 &= \sum_{j \in \Gamma(i)} \left| (\mathbf{G}^i - \nabla u_h + \nabla u_h - \nabla u) \cdot \mathbf{X}^{ij} \right|^2 \\ &\stackrel{\text{Triangle inequality}}{\leq} \sum_{j \in \Gamma(i)} \left( \left| (\mathbf{G}^i - \nabla u_h) \cdot \mathbf{X}^{ij} \right| + \left| (\nabla u - \nabla u_h) \cdot \mathbf{X}^{ij} \right| \right)^2 \\ &\leq \sum_{j \in \Gamma(i)} 2 \left( \left| (\mathbf{G}^i - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2 + \left| (\nabla u - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2 \right) \\ &\stackrel{\text{from (2.47)}}{\leq} 4 \sum_{j \in \Gamma(i)} \left| (\nabla u - \nabla u_h) \cdot \mathbf{X}^{ij} \right|^2 \\ &\stackrel{\text{from (2.27)}}{\leq} 4 \sum_{j \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|, \end{aligned} \quad (2.48)$$

leading to the statement of the proposition.  $\square$

In proposition 19 we have came up with an error estimate over the edges sharing node  $X^i$ . In the following proposition we aim at providing error estimates along a single edge  $X^i X^j$ .

**Proposition 20.** Let  $u \in \mathbf{C}^2(\overline{\Omega})$ . Then for  $\mathbf{G}^i$  defined on a node  $X^i$  by (2.43) and every node  $X^j$  connected to  $X^i$ , we have the following projected error bounds:

$$a - \left| (\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij} \right| \leq 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}| \right)^{\frac{1}{2}}.$$

b -

$$|(\mathbf{G}^i - \nabla u_h) \cdot \mathbf{X}^{ij}| \leq 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}| \right)^{\frac{1}{2}} + \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|.$$

c -

$$|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| \leq 2 \left( \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}| \right)^{\frac{1}{2}} + \left( \sum_{k \in \Gamma(j)} \max_{\mathbf{Y} \in [X^j, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{jk} \cdot \mathbf{X}^{jk}| \right)^{\frac{1}{2}} + \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}| \right).$$

*Proof.* Proof of a:

It follows from proposition 19. Indeed:

$$|(\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij}| \leq \left( \sum_{k \in \Gamma(i)} |(\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ik}|^2 \right)^{\frac{1}{2}} \leq 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}|^2 \right)^{\frac{1}{2}}. \quad (2.49)$$

Proof of b:

The error between the reconstructed gradient and the true gradient of the approximate solution can be split into two components:

$$\left| (\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij} \right| = \left| (\mathbf{G}^i - \nabla u_h + \nabla u_h - \nabla u) \cdot \mathbf{X}^{ij} \right|.$$

Applying this decomposition, together with inequality (2.49) yields:

$$\begin{aligned} \left| (\mathbf{G}^i - \nabla u) \cdot \mathbf{X}^{ij} \right| &\stackrel{\text{Triangle inequality}}{\leq} \left| (\mathbf{G}^i - \nabla u_h) \cdot \mathbf{X}^{ij} \right| + \left| (\nabla u_h - \nabla u) \cdot \mathbf{X}^{ij} \right| \\ &\stackrel{\text{from (2.49)}}{\leq} 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}|^2 \right)^{\frac{1}{2}} + \left| (\nabla u_h - \nabla u) \cdot \mathbf{X}^{ij} \right| \\ &\stackrel{\text{from (2.27)}}{\leq} 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}|^2 \right)^{\frac{1}{2}} + \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}| \end{aligned} \quad (2.50)$$

Proof of c:

Recall from equation (2.26) that  $\nabla u_h \cdot \mathbf{X}^{ij} = U^{ij}$ , then the above inequality can be

rewritten as:

$$\left| \mathbf{G}^i \cdot \mathbf{X}^{ij} - U^{ij} \right| \leq 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}|^2 \right)^{\frac{1}{2}} + \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \quad (2.51)$$

Again using the splitting trick, one can rewrite  $|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|$  as:

$$|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| = |\mathbf{G}^j \cdot \mathbf{X}^{ij} - \mathbf{G}^i \cdot \mathbf{X}^{ij}| = |\mathbf{G}^j \cdot \mathbf{X}^{ij} - U^{ij} + U^{ij} - \mathbf{G}^i \cdot \mathbf{X}^{ij}|. \quad (2.52)$$

The triangle inequality on (2.52) implies:

$$|\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| \leq |\mathbf{G}^j \cdot \mathbf{X}^{ij} - U^{ij}| + |U^{ij} - \mathbf{G}^i \cdot \mathbf{X}^{ij}| \quad (2.53)$$

Using inequality (2.51) on  $|\mathbf{G}^j \cdot \mathbf{X}^{ij} - U^{ij}|$  and  $|U^{ij} - \mathbf{G}^i \cdot \mathbf{X}^{ij}|$  leads to:

$$\begin{aligned} |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| &\leq 2 \left( \sum_{k \in \Gamma(j)} \max_{\mathbf{Y} \in [X^j, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{jk} \cdot \mathbf{X}^{jk}|^2 \right)^{\frac{1}{2}} \\ &\quad + 2 \left( \sum_{k \in \Gamma(i)} \max_{\mathbf{Y} \in [X^i, X^k]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}|^2 \right)^{\frac{1}{2}} \\ &\quad + \max_{\mathbf{Y} \in [X^i, X^j]} |\mathbb{H}(u)(\mathbf{Y}) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \end{aligned} \quad (2.54)$$

Finally all the estimates of the proposition follow.  $\square$

From the last inequality of proposition 20 we infer that  $|\mathbf{G}^{ij} \mathbf{X}^{ij}|$  is an estimate of the projected Hessian of the scalar field  $u$  along the edge  $\mathbf{X}^{ij}$ . We notice that the projected reconstructed gradient along edge  $\mathbf{X}^{ij}$  is equally affected by two contributions one from the patch associated with node  $X^i$  and the other from the patch associated with node  $X^j$ .

### 2.5.6 Edge-based error estimation

Based on the previous error analysis on the reconstruction gradient, we use

$$e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| \quad (2.55)$$

as an estimate of the error along the edge  $\mathbf{X}^{ij}$ .

**Proposition 21.** *Let  $u$  be a quadratic function:*

$$u(x) = u(X^i) + \nabla u(X^i)(x - X^i) + \frac{1}{2}(x - X^i)^t \mathbb{H}(u)(x - X^i) \quad x \in [X^i, X^j]. \quad (2.56)$$

*Then the edge based error respects the following equality:*

$$e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| = |\mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \quad (2.57)$$

*Proof.* Differentiating the quadratic function with respect to  $x$ , one gets:

$$\nabla u(x) = \nabla u(X^i) + \mathbb{H}(u)(x - X^i). \quad (2.58)$$

Taking  $x = X^j$ , and using the recovered gradient as a representation of the gradient at the nodes of the mesh yields:

$$\mathbf{G}^j = \mathbf{G}^i + \mathbb{H}(u)\mathbf{X}^{ij} \quad (2.59)$$

Rearranging the terms and projecting onto  $\mathbf{X}^{ij}$ , we can express the edge based error as:

$$e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}| = |\mathbb{H}(u)\mathbf{X}^{ij} \cdot \mathbf{X}^{ij}| \quad (2.60)$$

□

Note that for a regular function  $u$  we can always consider its quadratic form (from a Taylor development) so that the proposition remains verified.

**Remark 22.** *Future work will focus on generalizing the interpolation error estimates into approximation error ones.*

We recall that the objective of the anisotropic metric based mesh adaptation is to build up edges of unit length in the Riemannian metric space. Stretching factors will be defined on the edges to transform them into unit lengths. The motivation in this section is to perform an edge-based error estimation over an initial mesh  $\mathcal{H}_h$  that will be used to construct a metric tensor prescribing an optimal mesh  $\widetilde{\mathcal{H}}_h$ . From now on, the symbol tilde will be used to refer to quantities on the optimal mesh.

We are interested now in studying how the edge-based error changes as a result of modifying the length of edge  $\mathbf{X}^{ij}$  by a stretching factor  $s_{ij} \in \mathbb{R}^+$ . The stretching of an edge in its own direction is a linear mapping defined by:

$$\begin{aligned} \mathcal{H}_h &\mapsto \widetilde{\mathcal{H}}_h \\ \mathbf{X}^{ij} &\mapsto \widetilde{\mathbf{X}}^{ij} = s_{ij}\mathbf{X}^{ij}. \end{aligned} \quad (2.61)$$

Clearly, since  $s_{ij}$  is in  $\mathbb{R}^+$  the transformation can be a stretching or a shrinking as depicted in figure 2.5. For simplicity we will refer to both cases as stretching the edge.

**Proposition 23.** *Denote by  $s_{ij} \in \mathbb{R}^+$  the stretching factor that transforms edge  $\mathbf{X}^{ij}$  in its own direction according to mapping (2.61), and let  $\widetilde{e}_{ij} = |\widetilde{\mathbf{G}}^{ij} \cdot \widetilde{\mathbf{X}}^{ij}|$ . Then this error satisfies the following inequality:*

$$\widetilde{e}_{ij} \leq s_{ij}^2 e_{ij}. \quad (2.62)$$

*In other words, as a result of stretching the edge by a factor  $s_{ij}$ , the error changes quadratically with respect to  $s_{ij}$ .*

*Proof.* Without loss of generality, we focus on the case of a quadratic function  $u$ . The analysis can be extended to regular non quadratic functions by considering the

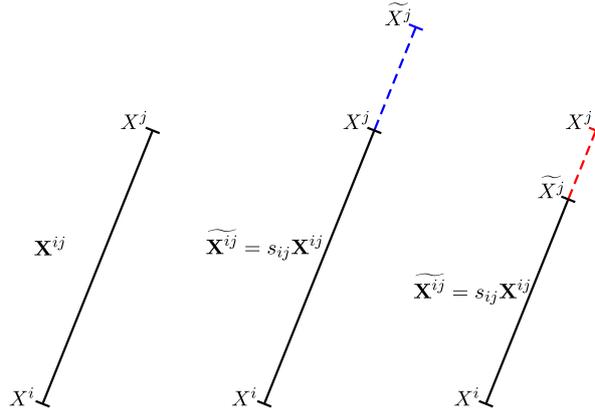


Figure 2.5 – Stretching or shrinking of a spatial edge  $\mathbf{X}^{ij}$  as a result of a scaling  $s_{ij}$ .

quadratic form of  $u$ . Let  $x$  be a point on  $\mathbf{X}^{ij}$ , then

$$u(x) = U^i + s \nabla u(X^i) \cdot \mathbf{X}^{ij} + \frac{1}{2} s^2 \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.63)$$

Evaluating at the point  $\widetilde{X}^j = X^i + s_{ij} \mathbf{X}^{ij}$  and rearranging the terms, one gets:

$$\widetilde{U}^{ij} = s_{ij} \nabla u(X^i) \cdot \mathbf{X}^{ij} + \frac{1}{2} s_{ij}^2 \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.64)$$

But since  $\widetilde{U}^{ij} = s_{ij} \nabla u_h \cdot \mathbf{X}^{ij}$ , then:

$$|s_{ij} \nabla u_h \cdot \mathbf{X}^{ij} - s_{ij} \nabla u(X^i) \cdot \mathbf{X}^{ij}| = \frac{1}{2} s_{ij}^2 |\mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}|. \quad (2.65)$$

On the other hand, for  $u_h$ , the interpolation of  $u$  on the given mesh, we have:

$$u_h(\widetilde{X}^j) = U^i + s_{ij} \nabla u_h \cdot \mathbf{X}^{ij} \quad (2.66)$$

Combining (2.65) and (2.66) we obtain an estimate on the error along the stretched edge  $\widetilde{\mathbf{X}}^{ij}$ ,

$$\begin{aligned} \tilde{e}_{ij} &= \left| u(\widetilde{X}^j) \cdot \widetilde{\mathbf{X}}^{ij} - u_h(\widetilde{X}^j) \cdot \widetilde{\mathbf{X}}^{ij} \right| \\ &= \left| U^i + s_{ij} \nabla u(X^i) \cdot \mathbf{X}^{ij} + \frac{1}{2} s_{ij}^2 \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} - (U^i + s_{ij} \nabla u_h \cdot \mathbf{X}^{ij}) \right| \\ &\leq \left| s_{ij} \nabla u_h \cdot \mathbf{X}^{ij} - s_{ij} \nabla u(X^i) \cdot \mathbf{X}^{ij} \right| + \frac{1}{2} s_{ij}^2 \left| \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \right| \\ &\leq \frac{1}{2} s_{ij}^2 \left| \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \right| + \frac{1}{2} s_{ij}^2 \left| \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \right| \\ &\leq s_{ij}^2 \left| \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \right| \end{aligned} \quad (2.67)$$

Therefore,

$$\tilde{e}_{ij} = |\widetilde{\mathbf{G}}^{ij} \cdot s_{ij} \mathbf{X}^{ij}| \leq s_{ij}^2 e_{ij}, \quad (2.68)$$

where

$$\widetilde{\mathbf{G}}^{ij} = \widetilde{\mathbf{G}}^j - \mathbf{G}^i.$$

□

Whence we have proved the quadratic behavior of the error estimator with respect to the stretching factors.

Before proceeding into the construction of the metric tensor that controls the errors along the edges, let us examine how the control on the error along the edges induces a control of the error over the elements in the mesh.

### 2.5.7 Optimal control on the Lp norm of the interpolation error

In this section we address the question of how the edge error estimates can be brought back to element error estimates by the use of appropriate integration rules. But before we conduct the study of element based error estimation in terms of the edge based estimation, let us recall some useful integration rules.

#### Recall: Second order exact integration rules

Let  $p \in \mathcal{C}^2(K)$  be a quadratic function defined on an element  $K$ . Let  $\mathbf{X}^{ij}$  be an edge of this element. Then the following second order Gauss quadrature integration rules are exact:

$$\int_K p(x) dK = \frac{1}{3} |K| \sum_{\mathbf{X}^{ij} \in K, i \neq j} p\left(\frac{\mathbf{X}^i + \mathbf{X}^j}{2}\right), \quad \text{in (2D)}, \quad (2.69)$$

and

$$\int_K p(x) dK = |K| \left( -\frac{1}{20} \sum_{\mathbf{X}^i \in K} p(\mathbf{X}^i) + \frac{1}{5} \sum_{\mathbf{X}^{ij} \in K, i \neq j} p\left(\frac{\mathbf{X}^i + \mathbf{X}^j}{2}\right) \right), \quad \text{in (3D)}. \quad (2.70)$$

Using these estimates, we develop the following proposition on the computation of the  $L_1$  norm of the interpolation error over the elements of the mesh.

**Proposition 24.** *Let  $u \in \mathcal{C}^2(\overline{\Omega})$  be the quadratic form of a regular function. The Taylor series development of  $u$  at a point  $x$  on the edge  $\mathbf{X}^{ij}$  reads as:*

$$u(x) = u(\mathbf{X}^i) + \alpha \nabla u(\mathbf{X}^i) \mathbf{X}^{ij} + \frac{1}{2} \alpha^2 \mathbf{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}, \quad \alpha \in [0, 1]. \quad (2.71)$$

The  $L_1$  norm of the interpolation error over the discretized domain is then given by:

$$\|u - \Pi_h u\|_1 = \frac{1}{24} \sum_{K \in \mathcal{H}} |K| \sum_{\mathbf{X}^{ij} \in K, i \neq j} e_{ij}, \quad \text{in (2D)}, \quad (2.72)$$

and

$$\|u - \Pi_h u\|_1 = \frac{1}{40} \sum_{K \in \mathcal{H}} |K| \sum_{\mathbf{X}^{ij} \in K, i \neq j} e_{ij}, \quad \text{in (3D)}. \quad (2.73)$$

*Proof.* Let  $u_h \in \mathbf{C}^1(\bar{\Omega})$  be the linear interpolation of  $u$  over  $\mathcal{H}$ . We have for  $x \in \mathbf{X}^{ij}$ :

$$u_h(x) = u(X^i) + \alpha \nabla u_h(X^i) \mathbf{X}^{ij}. \quad (2.74)$$

Since  $\nabla u_h \in \mathbf{C}^0(\bar{\Omega})$ , then  $\nabla u_h(X^i) = \nabla u_h|_{\mathbf{X}^{ij}}$ .

For the sake of notation simplicity, we will use  $\nabla u_h \cdot \mathbf{X}^{ij}$  to denote  $\nabla u_h|_{\mathbf{X}^{ij}} \cdot \mathbf{X}^{ij}$ .

The error at a point  $x$  of the edge  $\mathbf{X}^{ij}$  is given by:

$$\begin{aligned} e(x(\alpha)) &= (u(x(\alpha)) - u_h(x(\alpha))) \\ &= U^i + \alpha \nabla u(X^i) \mathbf{X}^{ij} + \frac{1}{2} \alpha^2 \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} - \left( U^i + \alpha \nabla u_h(X^i) \mathbf{X}^{ij} \right) \\ &= \alpha \left( \nabla u(X^i) \mathbf{X}^{ij} - \nabla u_h(X^i) \mathbf{X}^{ij} \right) + \frac{1}{2} \alpha^2 \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \end{aligned} \quad (2.75)$$

Taking  $x = X^j$  leads to:

$$e(X^j) = \left( u(X^j) - u_h(X^j) \right) = \left( \nabla u(X^i) \mathbf{X}^{ij} - \nabla u_h(X^i) \mathbf{X}^{ij} \right) + \frac{1}{2} \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.76)$$

But  $u(X^j) = u_h(X^j)$  as  $u_h$  is the linear interpolation of  $u$ , thus:

$$0 = \left( \nabla u(X^i) \mathbf{X}^{ij} - \nabla u_h(X^i) \mathbf{X}^{ij} \right) + \frac{1}{2} \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.77)$$

It follows that:

$$\left( \nabla u(X^i) \mathbf{X}^{ij} - \nabla u_h(X^i) \mathbf{X}^{ij} \right) = -\frac{1}{2} \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \quad (2.78)$$

Substituting equation (2.78) in (2.75) yields:

$$\begin{aligned} e(x(\alpha)) &= -\alpha \frac{1}{2} \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} + \frac{1}{2} \alpha^2 \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \\ &= \frac{\alpha}{2} (\alpha - 1) \mathbb{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \end{aligned} \quad (2.79)$$

In order to compute the  $L_1$  norm of the interpolation error over an element  $K$ , we use the second-order quadrature rules presented in equations (2.69) and (2.70).

We highlight the fact that the interpolation error is null at the vertices of the element, therefore the element interpolation error is given by:

$$\int_K e(x) dK = |K| \sum_{\mathbf{X}^{ij} \in K, i \neq j} \omega e \left( \frac{X^i + X^j}{2} \right) \quad (2.80)$$

where  $\omega = \frac{1}{3}$  in (2D) and  $\omega = \frac{1}{5}$  in (3D).

This element-wise error can then be expressed as:

$$\int_K e(x) dK = |K| \sum_{\mathbf{x}^{ij} \in K, i \neq j} \omega e \left( x \left( \frac{1}{2} \right) \right).$$

From equation (2.79), it follows:

$$\int_K e(x) dK = |K| \sum_{\mathbf{x}^{ij} \in K, i \neq j} \omega \left| \frac{1}{4} \left( -\frac{1}{2} \right) \right| \left| \mathbf{H}(u) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \right|.$$

Using equation (2.57) implies:

$$\int_K e(x) dK = \frac{1}{8} \omega |K| \sum_{\mathbf{x}^{ij} \in K, i \neq j} \left| \mathbf{G}^{ij} \cdot \mathbf{X}^{ij} \right|. \quad (2.81)$$

Summing over the elements  $K \in \mathcal{H}$ , we obtain an estimation for the  $L_1$  norm of the interpolation error:

$$\|u - \Pi_h u\|_1 = \sum_{K \in \mathcal{H}} |K| \int_K e(x) dK = \frac{1}{8} \omega \sum_{K \in \mathcal{H}} |K| \sum_{\mathbf{x}^{ij} \in K, i \neq j} \left| \mathbf{G}^{ij} \cdot \mathbf{X}^{ij} \right|, \quad (2.82)$$

where  $\omega = \frac{1}{3}$  in (2D) and  $\omega = \frac{1}{5}$  in (3D).

Therefore the statement of the proposition follows.  $\square$

In the previous proposition, we conducted a theoretical study of the equivalence between the  $L_1$  interpolation error and the proposed edge based error estimation. We intend to present next a generalization to the  $L_p$  norm.

We want to provide a theoretical validation of the equivalence between the proposed edge based error estimator and the interpolation error in  $L_p$  norm for a quadratic function. This is done, in the sequel, by expressing the element error analysis by an edge based error analysis and using the appropriate integration rules.

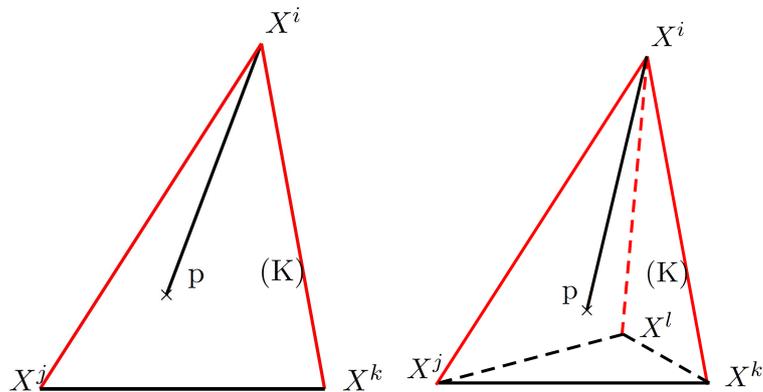


Figure 2.6 – Arbitrary point  $P$  inside an element connected to the node  $X^i$ .

**Proposition 25.** Consider an element  $K \in \mathcal{H}$  and let  $P$  be a point in  $K$  as presented in figure

2.6. In what follows,  $P$  will be taken as a Gauss quadrature point. Then we have for the two dimensional case:

$$\begin{aligned} X^i P &= v \mathbf{X}^i \mathbf{X}^j + w \mathbf{X}^i \mathbf{X}^k \Leftrightarrow P = (1 - v - w) X^i + v X^j + w X^k, \\ \text{with } v &= \frac{x_p - x_i}{x_j - x_i} \text{ and } w = \frac{y_p - y_i - v(y_j - y_i)}{y_k - y_i} \end{aligned}$$

being its barycentric coordinates, and for the three dimensional case:

$$X^i P = \alpha \mathbf{X}^i \mathbf{X}^j + \beta \mathbf{X}^i \mathbf{X}^k + \gamma \mathbf{X}^i \mathbf{X}^l,$$

with  $\alpha, \beta, \gamma$  being its barycentric coordinates.

The interpolation error at  $P$  can be evaluated in 2D by:

$$u(P) - \Pi_h u(P) = \frac{v}{2}(v-1)e_{ij} + \frac{w}{2}(w-1)e_{ik} + vwG^{ij}\mathbf{X}^{\mathbf{ik}},$$

and in 3D by:

$$u(P) - \Pi_h u(P) = \frac{\alpha}{2}(\alpha-1)e_{ij} + \frac{\beta}{2}(\beta-1)e_{ik} + \frac{\gamma}{2}(\gamma-1)e_{il} + \alpha\beta G^{ij}\mathbf{X}^{\mathbf{ik}} + \alpha\gamma G^{ij}\mathbf{X}^{\mathbf{il}} + \beta\gamma G^{ik}\mathbf{X}^{\mathbf{il}}.$$

*Proof.* Without loss of generality, we will provide the proof for the (2D) case in order to avoid redundancy, the same analysis can be done for (3D).

Using a Taylor series approximation around the point  $X^i$  we get:

$$\begin{aligned} u(P) &= U^i + \nabla u(X^i) \mathbf{X}^i \mathbf{P} + \frac{1}{2} \nabla^2 u(X^i) \mathbf{X}^i \mathbf{P} \cdot \mathbf{X}^i \mathbf{P} \\ &= U^i + \nabla u(X^i) (v \mathbf{X}^{\mathbf{ij}} + w \mathbf{X}^{\mathbf{ik}}) + \frac{1}{2} \nabla^2 u(X^i) (v \mathbf{X}^{\mathbf{ij}} + w \mathbf{X}^{\mathbf{ik}}) \cdot (v \mathbf{X}^{\mathbf{ij}} + w \mathbf{X}^{\mathbf{ik}}) \\ &= U^i + v \nabla u(X^i) \mathbf{X}^{\mathbf{ij}} + w \nabla u(X^i) \mathbf{X}^{\mathbf{ik}} + \frac{1}{2} v^2 \mathbb{H}u(X^i) \mathbf{X}^{\mathbf{ij}} \cdot \mathbf{X}^{\mathbf{ij}} \\ &\quad + \frac{1}{2} w^2 \mathbb{H}u(X^i) \mathbf{X}^{\mathbf{ik}} \cdot \mathbf{X}^{\mathbf{ik}} + vw \mathbb{H}u(X^i) \mathbf{X}^{\mathbf{ij}} \cdot \mathbf{X}^{\mathbf{ik}}. \end{aligned}$$

The linear interpolation of  $u$  at  $P$  is given by:

$$\begin{aligned} \Pi_h u(P) &= (1 - v - w)U^i + vU^j + wU^k \\ &= (1 - v - w)U^i + v \left( U^i + \nabla_h u(X^i) \mathbf{X}^{\mathbf{ij}} \right) + w \left( U^i + \nabla_h u(X^i) \mathbf{X}^{\mathbf{ik}} \right). \end{aligned}$$

Hence, the interpolation error can be expressed by:

$$\begin{aligned} u(P) - \Pi_h u(P) &= v \left( \nabla u(X^i) - \nabla_h u(X^i) \right) \mathbf{X}^{\mathbf{ij}} + w \left( \nabla u(X^i) - \nabla_h u(X^i) \right) \mathbf{X}^{\mathbf{ik}} \\ &\quad + \frac{v^2}{2} \mathbb{H}u(X^i) \mathbf{X}^{\mathbf{ij}} \cdot \mathbf{X}^{\mathbf{ij}} + \frac{w^2}{2} \mathbb{H}u(X^i) \mathbf{X}^{\mathbf{ik}} \cdot \mathbf{X}^{\mathbf{ik}} \\ &\quad + vw \mathbb{H}u(X^i) \mathbf{X}^{\mathbf{ij}} \cdot \mathbf{X}^{\mathbf{ik}}. \end{aligned} \tag{2.83}$$

Evaluating at the point  $P = X^j$  we have  $u(P) - \Pi_h u(P) = 0$ ,  $v = 1$  and  $w = 0$ . This leads to:

$$\begin{aligned} 0 &= \left( \nabla u(X^i) - \nabla_h u(X^i) \right) \mathbf{X}^{ij} + \frac{v^2}{2} \mathbb{H}u(X^i) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} \\ \Leftrightarrow \left( \nabla u(X^i) - \nabla_h u(X^i) \right) \mathbf{X}^{ij} &= -\frac{v^2}{2} \mathbb{H}u(X^i) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij}. \end{aligned} \quad (2.84)$$

Similarly, by taking  $P = X^k$ , we get:

$$\left( \nabla u(X^i) - \nabla_h u(X^i) \right) \mathbf{X}^{ik} = -\frac{w^2}{2} \mathbb{H}u(X^i) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik}. \quad (2.85)$$

Rearranging the terms and using equations (2.84) and (2.85), equation (2.83) reduces to:

$$\begin{aligned} u(P) - \Pi_h u(P) &= \frac{v}{2}(v-1) \mathbb{H}u(X^i) \mathbf{X}^{ij} \cdot \mathbf{X}^{ij} + \frac{w}{2}(w-1) \mathbb{H}u(X^i) \mathbf{X}^{ik} \cdot \mathbf{X}^{ik} + vw \mathbb{H}u(X^i) \mathbf{X}^{ij} \cdot \mathbf{X}^{ik} \\ &= \frac{v}{2}(v-1) e_{ij} + \frac{w}{2}(w-1) e_{ik} + vw G^{ij} \mathbf{X}^{ik}. \end{aligned}$$

□

The interpolation error at any point  $P$  of  $(K)$  can therefore be computed using the proposed edge based error estimator.

**Proposition 26.** *The  $L^p$  norm of the interpolation error over the domain  $\Omega$  is given in (2D) by:*

$$\left( \int_{\Omega} |u - \Pi_h u|^p d\Omega \right)^{\frac{1}{p}} = \left( \sum_{K=1}^{n_K} |K| \sum_{g=1}^{n_G} \omega_K^g \left| \frac{v}{2}(v-1) e_{ij}^K + \frac{w}{2}(w-1) e_{ik}^K + vw G_K^{ij} \mathbf{X}^{ik} \right|^p \right)^{\frac{1}{p}},$$

and in (3D) by:

$$\begin{aligned} \left( \int_{\Omega} |u - \Pi_h u|^p d\Omega \right)^{\frac{1}{p}} &= \left( \sum_{K=1}^{n_K} |K| \sum_{g=1}^{n_G} \omega_K^g \left| \frac{\alpha}{2}(\alpha-1) e_{ij}^K + \frac{\beta}{2}(\beta-1) e_{ik}^K \right. \right. \\ &\quad \left. \left. + \alpha\beta G_K^{ij} \mathbf{X}^{ik} + \alpha\gamma G_K^{ij} \mathbf{X}^{il} + \beta\gamma G_K^{ik} \mathbf{X}^{il} \right|^p \right)^{\frac{1}{p}}, \end{aligned}$$

where  $n_K$  is the number of elements in the mesh,  $|K|$  the volume of the  $K^{\text{th}}$  element,  $n_G$  the number of Gauss points for the  $K^{\text{th}}$  element and  $\omega_K^g$  the  $g^{\text{th}}$  quadrature weight for the  $K^{\text{th}}$  element.

*Proof.* Without loss of generality, we will provide the proof for the (2D) case in order to avoid redundancy, the same analysis can be done for (3D).

$$\begin{aligned}
\left( \int_{\Omega} |u - \Pi_h u|^p d\Omega \right)^{\frac{1}{p}} &= \left( \sum_{K=1}^{n_K} \int_K |u - \Pi_h u|^p dK \right)^{\frac{1}{p}} \\
&= \left( \sum_{K=1}^{n_K} |K| \sum_{g=1}^{n_G} \omega_K^g \left| u(Q_g^K) - \Pi_h u(Q_g^K) \right|^p dK \right)^{\frac{1}{p}} \\
&= \left( \sum_{K=1}^{n_K} |K| \sum_{g=1}^{n_G} \omega_K^g \left| \frac{v}{2}(v-1)e_{ij}^K + \frac{w}{2}(w-1)e_{ik}^K + vw \mathbf{G}_K^{ij} \mathbf{x}^{ik} \right|^p \right)^{\frac{1}{p}},
\end{aligned}$$

with  $Q_g^K$  being the quadrature points of element  $K$ .  $\square$

In this section, we have carried out an analysis on the equivalence between the edge based error estimator and the interpolation error in  $L_p$  norm for a regular function. Based on this result, the constructed mesh driven by the minimization of the edge based error estimation will enable an optimal control on the interpolation error in  $L_p$  norm and hence a good capture of all solution scales.

### 2.5.8 Optimal metric construction

As stated earlier, the mesh optimization algorithm consists of three major steps: a  $P_1$  gradient recovery using the length distribution tensor, followed by an edge-based a posteriori sampling of the interpolation error that will be synthesized to derive a new metric tensor encoding stretching feedback to drive the mesh toward optimality. In the previous sections, we have addressed the first two steps toward anisotropic mesh adaptation, we are now ready to build-up the metric field that generates the anisotropic and well-adapted mesh induced by the error estimator.

We have cast the mesh optimization problem of finding an optimal mesh that minimizes the  $L^p$  norm of the interpolation error of a certain field as a problem of seeking a metric field  $\mathcal{M}$  defined at the nodes of the mesh that drives the mesh toward optimality based on the error sampling estimator.

In the framework of metric based anisotropic mesh adaptation, two possible attempts for building the adapted mesh can be adopted: adapting the mesh while respecting a certain level of accuracy and adapting the mesh while preserving a fixed number of degrees of freedom. When dealing with unsteady phenomena, new features may appear in the solution, to keep up with the level of accuracy, the former approach tends to refine the mesh by adding new degrees of freedom which if not well controlled can cause a drastic increase in mesh complexity, thus impacting negatively the computational time of the simulation. However, from a practical point of view, in order to simulate long-time and large-scale industrial applications it is preferable to choose a number of nodes  $N$  based on the available computational resources and to determine the most accurate possible solution. So the level of accuracy is not set a priori but it is highly desirable to have the best representation of the simulated phenomena.

The method that we propose produces the optimal mesh under the constraint of a fixed number of nodes. It tends to refine the mesh in a hierarchical importance of the solution's gradient. In other words, if new features with high gradients appear in the solution, the mesh will be automatically coarsened in regions with lower gradients and will be refined near the newly emerging features. In that case, if a small number of nodes was fixed by the user, the solution will still be well captured although not with the same level of accuracy.

In this section, we intend to develop a metric based mesh adaptation that is capable of well capturing the anisotropy of physical phenomena by generating extremely stretched and highly directional elements under the constraint of a fixed number of nodes. The metric tensor would prescribe optimal mesh sizes and element orientations. The developed algorithm strives to improve the quality of the aforementioned estimates by attempting to reduce and equi-distribute the error over the edges of the mesh under the constraint of a fixed degree of freedom.

The novelty of the developed technique resides in the combination of an edge-based error estimation with the equi-distribution principle to derive a set of edge stretching factors resulting in an optimal anisotropic mesh adaptation. Unlike the Hessian-based techniques for metric construction, the method that we propose avoids the reconstruction of this tensor and renders a reduction in the computational cost.

The method consists in stretching the edges of a mesh in a way to better capture the anisotropy of the solution. Starting from the natural metric defined earlier and associated with the current mesh, we intend to modify the definition of the length distribution tensor to reflect the stretching of edges in their own direction. Without loss of generality, we consider a node  $X^i$  of the mesh and the edge  $\mathbf{X}^{ij}$  connected to  $X^i$ . The modified edge is defined by:

$$\widetilde{\mathbf{X}}^{ij} = s_{ij} \mathbf{X}^{ij}. \quad (2.86)$$

The length distribution tensor corresponding to the modified edges reads as:

$$\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij} \mathbf{X}^{ij} \otimes s_{ij} \mathbf{X}^{ij},$$

Consequently, the associated metric tensor will be given by:

$$\mathcal{M}^i = \frac{1}{d} (\mathbb{X})^{-1}. \quad (2.87)$$

The issue that remains to be addressed is how to define the stretching factors in a way that the interpolation error in the new resulting optimal mesh will be minimized under the constraint of a fixed number of nodes.

### 2.5.8.1 Metric construction minimizing the total induced error for a fixed number of edges

A metric construction that intends to minimize the total error induced on the edges of a mesh was proposed in [Coupez 11]. The result is summarized in the following theorem:

**Theorem 26.1.** *Let  $A$  be a prescribed number of edges,  $e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|$  be the calculated error along the edge  $\mathbf{X}^{ij}$  of the previous mesh, and  $p \in [1, d]$  be an exponent to be defined.*

*Consequently, for a set of stretching coefficients*

$$S = \{s_{ij} \in \mathbb{R}^+; i = 1, \dots, N; j = 1, \dots, N; \Gamma(i) \cap \Gamma(j) \neq \emptyset\},$$

*the continuous metric field defined at the mesh nodes and that minimizes the interpolation error for a fixed number of nodes is given by:*

$$\mathbb{M}^i = \frac{1}{d} \left( \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1}, \quad (2.88)$$

where,

$$s_{ij} = \left( \frac{\lambda}{e_{ij}} \right)^{\frac{1}{p+2}}, \quad (2.89)$$

and

$$\lambda = \left( \frac{\sum_i \sum_{j \in \Gamma(i)} e_{ij}^{\frac{p}{p+2}}}{A} \right)^{\frac{p+2}{p}}. \quad (2.90)$$

*Proof.* By virtue of the quadratic behavior of the error with respect to the stretching factors, the total error induced on the modified mesh is given by:

$$\varphi(S) = \frac{1}{2} \sum_{i,j} s_{ij}^2 e_{ij}.$$

Note that the factor  $\frac{1}{2}$  is used to avoid counting the contribution from edge  $\mathbf{X}^{ij}$  twice, once as  $\mathbf{X}^{ij}$  and once as  $\mathbf{X}^{ji}$ .

We aim at minimizing the functional  $\varphi(s)$  under the constraint of a fixed number of edges:

$$S = \underset{S, \text{ s.t. } \sum_{i < j} 1}{\operatorname{argmin}} \varphi(s). \quad (2.91)$$

We introduce  $\eta_{ij} = s_{ij}^{-p}$  as being the number of edges created as a result of reducing the edge  $\mathbf{X}^{ij}$  by a factor  $s_{ij}$ . The exponent  $p$  intends to take into account the fact that the edges are created not only in the direction of  $\mathbf{X}^{ij}$  but also implicitly from the crossing of the newly emerged edges. Thus the constraint from the imposed number of edges

is written as:

$$\sum_{ij} s_{ij}^{-p} = A.$$

Then the optimization problem can be reformulated as a Lagrangian:

$$\mathcal{L}(S, \lambda) = \varphi(S) + \frac{\lambda}{p} ((\eta_{ij}) - A), \quad (2.92)$$

where  $\frac{\lambda}{p}$  is the Lagrangian multiplier to account for the constraint on the number of edges.

At minimum, we have:

$$\frac{\partial \mathcal{L}}{\partial S} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial \lambda} = 0.$$

Therefore, for every  $s_{ij}$  we have:

$$\frac{\partial \mathcal{L}}{\partial s_{ij}} = 0 \implies s_{ij} e_{ij} - \lambda s_{ij}^{(p+1)} = 0 \implies s_{ij} = \left( \frac{\lambda}{e_{ij}} \right)^{\frac{1}{p+2}} \quad (2.93)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \implies \sum_{ij} \left( \frac{e_{ij}}{\lambda} \right)^{\frac{p}{p+2}} = A \implies \lambda = \left( \frac{\sum_i \sum_{j \in \Gamma(i)} e_{ij}^{\frac{p}{p+2}}}{A} \right)^{\frac{p+2}{p}}. \quad (2.94)$$

This proves the statement of the theorem.  $\square$

Obviously, a zero estimated error can lead to the generation of an edge with infinite length which cannot be handled by the mesher and in practice can lead to complications for the simulation. On the other hand, a very large error yields a mesh that is very refined at the location of high gradient. This implies either a loss in accuracy over the whole domain or the need to impose an excessive number of degrees of freedom. The latter results in a drastic increase in the computational cost of the simulation. These issues are resolved by using a regularization of the errors along the edges as follows:

$$\begin{cases} e_{ij} = \max \left( |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|, \varepsilon_{\min} |\mathbf{X}^{ij}|^2 \right), \\ e_{ij} = \min \left( |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|, \varepsilon_{\max} |\mathbf{X}^{ij}|^2 \right). \end{cases} \quad (2.95)$$

These choices are justified by the following proposition:

**Proposition 27.** *The mesh associated with the edge error*

$$e_{ij} = \varepsilon |\mathbf{X}^{ij}|^2$$

*is uniform if and only if:*

$$|\mathbf{X}^{ij}| = h$$

*the edge length is constant.*

*Proof.* Proof of  $\Rightarrow$ :

Assume that the mesh associated with edge error estimates  $e_{ij} = \varepsilon |\mathbf{X}^{ij}|^2$  is uniform with edge length  $h$ , then:

$$h = |\widetilde{\mathbf{X}}^{ij}| = s_{ij} |\mathbf{X}^{ij}| \implies s_{ij} = \frac{h}{|\mathbf{X}^{ij}|}.$$

The mesh is uniform when  $s_{ij} = 1$ .

Proof of  $\Leftarrow$ :

Assuming that all the edges are of the same constant length  $|\mathbf{X}^{ij}| = h$ , then

$$\lambda = \left( \frac{\sum_i \sum_{j \in \Gamma(i)} \varepsilon |\mathbf{X}^{ij}|^2}{A} \right)^{\frac{p+2}{p}} = \varepsilon |\mathbf{X}^{ij}|^2.$$

Hence  $s_{ij} = 1$ .

Therefore background meshes with fixed edge lengths can be used assigning the maximum and the minimum edge lengths. The error estimates for these meshes are proportional to the square of the edge length.  $\square$

**Remark 28.** *When the scaling factors for the mesh edges are one then the mesh is optimal. Hence the process of mesh adaptation can be repeated until all the stretching factors are almost one.*

**Remark 29.** *The exponent  $p$  intends to take into account the fact that the nodes are created not only along the edge but also implicitly from the crossing of the newly emerged edges. In general, we consider the number of created edges to be on average  $s_{ij}^{-d}$ .*

In this work we propose a new track for metric construction. It avoids the ambiguous choice of the parameter  $p$  in the previously described metric tensor. The method that we develop minimizes the error estimates of the interpolation error by imposing the equi-distribution principle for a fixed number of nodes. In the next chapter the equi-distribution will be done in space and time hence providing an additional control on the time step sizes.

**The error equi-distribution principle:** It is based on the idea of evenly distributing the error over the mesh. The concept of equi-distribution roots back to the work of [Burchard 74] on determining the best spline approximations with variable knots. The principle demonstrated its importance for multi-dimensional adaptive mesh generation and several works were developed in that line [Dvinsky 91, Cao 02, Chen 04a]. Babuška and Rheinboldt [Babuška 78] conducted an a posteriori error analysis for the finite element methods and showed that a mesh is asymptotically optimal if all the error indicators are equally distributed on the mesh. Other theoretical convergence analysis were also developed in [Kopteva 05, Cao 02, Chen 08].

Our procedure for building the metric tensor relies on considering a target homogeneous error that is equi-distributed over the mesh. In order to maintain a fixed number of degrees of freedom in the mesh, we define the target error as a function of the imposed number of nodes.

In section 2.5.6, we have demonstrated that the interpolation error along an edge  $\mathbf{X}^{ij}$  changes quadratically as the edge length is stretched by a factor  $s_{ij}$ . That is:

$$\tilde{e}_{ij} = s_{ij}^2 e_{ij}. \quad (2.96)$$

For the target error to be equi-distributed in the mesh corresponding to the stretched edges, we should have:

$$\forall \mathbf{X}^{ij} \in \tilde{\mathcal{H}}, \quad \tilde{e}_{ij} = \varepsilon, \quad (2.97)$$

where  $\varepsilon$  is a target error that is uniform and totally balanced over the mesh.

Substituting equation (2.97) into (2.96) we obtain:

$$s_{ij} = \left( \frac{\varepsilon}{e_{ij}} \right)^{\frac{1}{2}}. \quad (2.98)$$

Let  $n_{ij}$  be the number of created nodes in relation with the stretching factor  $s_{ij}$  and in the direction of the edge  $\mathbf{X}^{ij}$ . It is given by:

$$n_{ij} = \left( \frac{\varepsilon}{e_{ij}} \right)^{-\frac{1}{2}} = s_{ij}^{-1}. \quad (2.99)$$

In what follows, we will use the notation  $n_{ij}(\varepsilon)$  to highlight the dependence of  $n_{ij}$  on  $\varepsilon$ .

We are looking for a node distribution tensor  $\mathcal{N}^i$  associated with node  $X^i$  that distributes the number of nodes along the different directions of the edges connected to node  $X^i$ .

For each node  $X^i$  of the mesh, the number of created nodes along  $\mathbf{X}^{ij}$  is given by:

$$\mathcal{N}^i \mathbf{X}^{ij} = n_{ij} \mathbf{X}^{ij}. \quad (2.100)$$

This equation can be constraining so we relax it via the corresponding least squares approximation:

$$\mathcal{N}^i = \operatorname{argmin}_{\mathcal{N}^i \in \mathbb{R}_{\text{sym}}^{d \times d}} \sum_{j \in \Gamma(i)} |\mathcal{N} \mathbf{X}^{ij} - n_{ij} \mathbf{X}^{ij}|^2 \quad (2.101)$$

**Proposition 30.** *The node distribution tensor is defined in a least squares sense by:*

$$\mathcal{N}^i = \frac{1}{d} \left( \mathbb{X}^i \right)^{-1} \left( \sum_{j \in \Gamma(i)} n_{ij} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right) \quad (2.102)$$

*Proof.* Using the standard differentiation of (2.101) and noting that at the minimum, the derivative is zero, we get:

$$\sum_{j \in \Gamma(i)} \mathcal{N} \mathbf{X}^{ij} \mathbf{X}^{ij} = \sum_{j \in \Gamma(i)} n_{ij} \mathbf{X}^{ij} \mathbf{X}^{ij} \quad (2.103)$$

and equivalently,

$$\mathbb{X}^i : \mathcal{N}^i = \left( \sum_{j \in \Gamma(i)} n_{ij} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right) \quad (2.104)$$

Note that for a symmetric positive definite tensor  $A$  of order  $d$  we have  $A : A^{-1} = d$ , we thus obtain the node distribution tensor:

$$\mathcal{N}^i = \frac{1}{d} \left( \mathbb{X}^i \right)^{-1} \left( \sum_{j \in \Gamma(i)} n_{ij} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right) \quad (2.105)$$

□

The metric tensor that equi-distributes a uniform error over the mesh in the view of minimizing the interpolation error estimates under the constraint of a fixed number of nodes is given by the following theorem:

**Theorem 30.1.** *Let  $N$  be a prescribed number of nodes,  $e_{ij} = |\mathbf{G}^{ij} \cdot \mathbf{X}^{ij}|$  be the calculated error along the edge  $\mathbf{X}^{ij}$  of the previous mesh,  $\varepsilon$  the uniform equi-distributed error. Then, for a set of stretching coefficients*

$$\mathcal{S} = \{s_{ij} \in \mathbb{R}^+ ; i = 1, \dots, N ; j = 1, \dots, N ; \Gamma(i) \cap \Gamma(j) \neq \emptyset\},$$

*the continuous metric field defined at the mesh nodes and minimizing the interpolation error for a fixed number of nodes is given by:*

$$\mathbb{M}^i = \frac{1}{d} \left( \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1}, \quad (2.106)$$

where,

$$s_{ij} = \left( \frac{\varepsilon}{e_{ij}} \right)^{\frac{1}{2}}, \quad (2.107)$$

and

$$\varepsilon = \left( \frac{N}{\sum_i n^i(1)} \right)^{-\frac{4}{d}}. \quad (2.108)$$

In here,  $n^i(1)$  denotes the number of nodes created around node  $X^i$  for a global homogenous error of 1 and it is computed by:

$$n^i(1) = \sqrt{\det \left( \frac{1}{d} (\mathbb{X}^i)^{-1} \left( \sum_{j \in \Gamma(i)} \left( \frac{1}{e_{ij}} \right)^{-\frac{1}{2}} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij} \right) \right)} \quad (2.109)$$

*Proof.* The local mesh density i.e. the total number of created nodes per node  $X^i$  in the different directions connected to it is:

$$\begin{aligned} n^i(\varepsilon) &= \sqrt{\det \left( \frac{1}{d} (\mathbb{X}^i)^{-1} \left( \sum_{j \in \Gamma(i)} n_{ij} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij} \right) \right)} \\ &= \sqrt{\det \left( \frac{1}{d} (\mathbb{X}^i)^{-1} \left( \sum_{j \in \Gamma(i)} \left( \frac{\varepsilon}{e_{ij}} \right)^{-\frac{1}{2}} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij} \right) \right)}. \end{aligned} \quad (2.110)$$

Hence, we have:

$$\begin{aligned} n^i(\varepsilon) &= \varepsilon^{-\frac{d}{4}} \sqrt{\det \left( \frac{1}{d} (\mathbb{X}^i)^{-1} \left( \sum_{j \in \Gamma(i)} \left( \frac{1}{e_{ij}} \right)^{-\frac{1}{2}} \mathbf{x}^{ij} \otimes \mathbf{x}^{ij} \right) \right)} \\ \Leftrightarrow n^i(\varepsilon) &= \varepsilon^{-\frac{d}{4}} n^i(1). \end{aligned} \quad (2.111)$$

where  $n^i(1)$  is the number of created nodes for a uniform error equal to 1.

The total number of nodes in the adapted mesh is therefore:

$$\begin{aligned} N &= \sum_i n^i(\varepsilon) \\ N &= \varepsilon^{-\frac{d}{4}} \sum_i n^i(1). \end{aligned} \quad (2.112)$$

Consequently, the global induced uniform error for a given total number of nodes  $N$  can be determined by:

$$\varepsilon = \left( \frac{N}{\sum_i n^i(1)} \right)^{-\frac{4}{d}}. \quad (2.113)$$

We will denote the uniform induced error  $\varepsilon$  by  $\varepsilon(N)$  to reflect its dependence on the prescribed degrees of freedom.

Thus, the corresponding stretching factors under the constraint of a fixed number

of nodes  $N$  are given by:

$$s_{ij} = \left( \frac{\varepsilon(N)}{e_{ij}} \right)^{\frac{1}{2}} = \left( \frac{\sum_i n^i(1)}{N} \right)^{\frac{2}{d}} e_{ij}^{-1/2}. \quad (2.114)$$

We can now deduce the metric tensor associated with stretching the edges  $\mathbf{X}^{ij}$  of the mesh in their own directions by factors  $s_{ij}$ :

$$\mathcal{M}^i = \frac{1}{d} \left( \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1} \quad (2.115)$$

□

We use the same regularization forms (2.95) described above in order to bound the minimum and the maximum error estimates and hence control implicitly the minimum and maximum mesh sizes.

### 2.5.9 Privileged length distribution tensor

In the aim of building the metric tensor that reflects to the best the solution's anisotropy, we reconsider in this section the definition of the length distribution tensor that we constructed based on an averaged orientation property around a node  $X^i$ :

$$\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij},$$

where we average the contribution of the different edges to determine the state of node  $X^i$ . This definition can be modified to give additional weight to the edges in the direction of the solution's gradient, so we associate to each of the edges surrounding  $X^i$  a certain privilege that depends on its orientation hence resulting in a better capture of anisotropy. For that purpose we define for an edge  $\mathbf{X}^{ij}$  a weight  $\omega_{ij}$  as follows:

$$\omega_{ij} = \sin(\theta_{ij}) \quad (2.116)$$

where  $\theta_{ij}$  is the angle formed between the solution's gradient  $\nabla u$ , or the recovered gradient, and the edge  $\mathbf{X}^{ij}$  as shown in figure 2.7. The weights can then be computed by:

$$\omega_{ij} = \frac{\|\nabla u \wedge \mathbf{X}^{ij}\|}{\|\nabla u\| \cdot \|\mathbf{X}^{ij}\|} \quad (2.117)$$

As for most PDE's that we consider, the exact gradient is not known, we compute the weights with respect to the reconstructed gradient.

Using the privileging principle, the modified definition of the length distribution

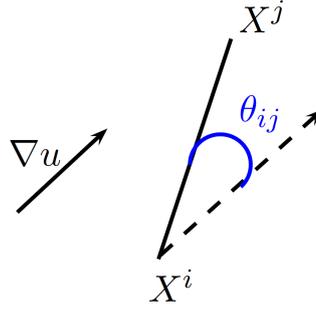


Figure 2.7 – Angle between the solution's gradient and the edge  $\mathbf{X}^{ij}$ .

tensor for computing the optimal metric tensor reads as:

$$\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij}^2 \omega_{ij} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij},$$

It is important to mention that for the case of null gradient, the weights will be set to one and a regularization on the error estimates will be applied as described above in equation (2.95). In this case the edges will be set to the maximum in accordance with the regularization formula. No additional privilege needs to be added. The efficiency of the newly defined length distribution tensor will be confronted in the numerical examples with the averaged distribution one.

**Remark 31.** *A theoretical validation on the optimality of this construction and its extension to ensure second order control on the error estimates will be the subject of future investigations.*

In this section, we have presented an optimal metric generation technique relying on an edge-based error estimation and on the equi-distribution principle. The full derivations and the rationales behind the construction were provided. We started by defining a length distribution tensor followed by a gradient recovery procedure. The reconstructed gradient is then used to compute an edge based interpolation error estimator. The latter is minimized under the constraint of a fixed number of nodes by respecting the equi-distribution principle. A set of stretching factors associated to each edge is then generated, leading to the optimal metric tensor.

The presented method is simple, easy to implement, independent of the problem at hand, and works under the constraint of a fixed number of nodes leading to the optimal (most accurate) mesh for a given computational power. An important characteristic of the method that we propose is that it does not require the knowledge/recovery of the solution Hessian which might negatively affect the computational cost of the simulation. The ability to control the  $L_p$  norm of the interpolation error permits a higher resolution of the different solution's scales, as will be demonstrated by the numerical results.

Since the metric field is defined at the nodes of the mesh, it can be given to any mesh generator that takes as input a metric field for whatever meshing technique (Delaunay, moving front method, local topological optimization,  $\dots$ ). Many of these mesh

generators can be found nowadays in the literature [Coupez 91, George 99, Frey 01, Bottasso 04, Geuzaine 09, Li 05]. In this work, we give the metric tensor to the MTC mesher (described earlier) that would in turn generate an anisotropic and adaptive mesh as will be discussed in the next section.

## 2.6 ANISOTROPIC MESH ADAPTATION THROUGH LOCAL TOPOLOGICAL OPTIMIZATION

We are interested, in this section, in answering the following question: how does a mesh generator respond to a given metric field? We will explain the general idea, the details are provided in [Gruau 05] and will be omitted here for brevity.

We have explained previously how a mesh of a domain is generated from the mesh of its boundary and how local topological optimization are repeated iteratively until the minimal volume criterion is satisfied and a good geometric quality is obtained.

Starting with an initial mesh, the minimal volume should be respected and the quality of the elements should be improved. The mesher goes through a series of topological optimization using the “starring” operator and the element quality is measured in relation with the metric.

In the context of metric based anisotropic mesh adaptation, the quality criterion for an element  $K$  in the mesh is defined by:

$$\mathcal{Q}_K = \min \left( \frac{c_0 |K|_{\mathcal{M}(K)}}{h_{\mathcal{M}(K)}^d}, h_{\mathcal{M}(K)}^d, \frac{1}{h_{\mathcal{M}(K)}^d} \right) \quad (2.118)$$

where,

- $\mathcal{M}(K)$  is the mean element metric:  $\mathcal{M}(K) = \frac{1}{d+1} \sum_{i=0}^d \mathcal{M}^i$ , with  $\mathcal{M}^i$  being the metric tensor provided at node  $X^i$ . We note that this choice can be improved into an affine invariant approach or a logarithmic reconstruction.
- $|K|_{\mathcal{M}} = |K| \sqrt{\mathcal{M}(K)}$  is the new volume of the element  $|K|$ .
- $h_{\mathcal{M}(K)} = \left( \frac{2}{d(d+1)} \sum_{i=0}^d \|\mathbf{X}^{ij}\|_{\mathcal{M}(K)}^2 \right)^{\frac{1}{2}}$  is the average edge length.
- $c_0 = \frac{d!}{\sqrt{d+1}} 2^{\frac{d}{2}}$  is a constant that is chosen so that a quality 1 would be obtained when the element is equilateral in the Riemannian space associated with the metric  $\mathcal{M}(K)$ .

After each topological modification, the metric tensor is interpolated from the old mesh to the new one. More precisely, when new nodes are created (the centroids of a certain patch) the values at these nodes are computed by averaging the values on the

surrounding nodes in the patch.

$$\mathcal{M}(C) = \frac{1}{n} \sum_{i=0}^n \mathcal{M}^i$$

When the new node is the midpoint of a segment joining two existing nodes  $[X^i, X^j]$  then the metric at the midpoint is defined to be the average of the metrics at the extremities.

$$\mathcal{M}(C) = \frac{\mathcal{M}^i + \mathcal{M}^j}{2}$$

This way of interpolation was demonstrated in [Gruau 05] to be fast and efficient. We note that this choice can be improved into an affine invariant approach or a logarithmic reconstruction.

We highlight the utility of defining the metric tensors at the nodes of the mesh for metric interpolation throughout the remeshing process. If the metrics were defined at the element level then an extrapolation to the nodes needs first to be performed followed by projecting back onto the created elements through the “copy-paste” process of the mesh topological improvement.

## 2.7 FIELDS' INTERPOLATION BETWEEN MESHES

Once a new mesh is obtained, the solution fields need to be interpolated from the background mesh  $\mathcal{H}_{\text{Background}}$  to the new mesh  $\mathcal{H}_{\text{new}}$  in order to pursue the computations. A classical interpolation method 2.8 is used to transfer data according to their type:

- $P_1$  variables: Nodal variables computed at the level of the background mesh vertices.
- $P_0$  variables: Element variables defined at the level of the background mesh elements.

### Transfer of $P_1$ data:

The transfer of the discrete  $P_1$  fields defined at the nodes' level is done in two steps:

1. A localization step, whereby a method of *bounding boxes* [Digonnet 01], is used to identify the element  $K = \{X_i^K, i = 1, \dots, d + 1\}$  in the background mesh that contains the new node  $P$ .
2. A linear interpolation of the  $P_1$  variable field from the nodes of the element  $K$  to  $P$  using its barycentric coordinates  $\beta_i$  with respect to  $K$ :

$$\Pi_h u(P) = \sum_{i=1}^{d+1} \beta_i(P) u_h(X_i^K) \quad (2.119)$$

This interpolation scheme is  $P_1$  exact, of second order and independent of the mesh topology.

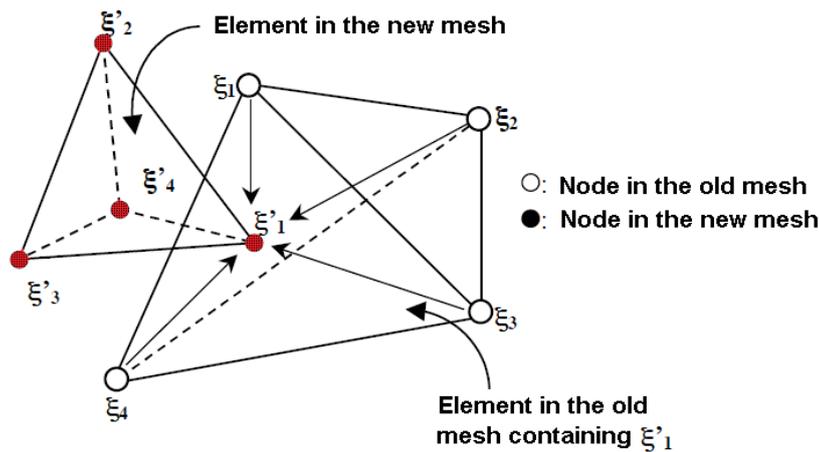


Figure 2.8 –  $P_1$  interpolation from an old mesh to a new one, adopted from [Boussetta 06].

### Transfer of $P_0$ data:

There are two possible ways of transferring  $P_0$  fields from the background to the new mesh. One possibility is to extrapolate the fields from element level onto the nodes then apply a  $P_1$  transfer to the Gauss points of  $\mathcal{H}_{\text{new}}$  elements. The other possibility is to apply a direct transfer whereby the Gauss points of the new elements are localized inside the elements of the background mesh and take their values. The latter technique showed that it makes rough choices especially for coarse elements whereas the former one induces diffusion due to the extrapolation step. However the projection technique suits better the MTC mesher since the latter proceeds by local modification of topologies, i.e. if a certain region of the mesh is not changed, the data will not be transferred, hence avoiding numerical diffusion.

Before we move on to the numerical validation of the proposed metric-based mesh adaptation method, let us recall its key features:

- The adaptation yields a control on the  $L_p$  norm of the interpolation error through a procedure relying on a posteriori estimates without any a priori assumptions on the solution behavior.
- The method does not require the recovery of the solution's Hessian which is a costly step in most metric-based mesh adaptation techniques present in the literature.
- The method produces the optimal mesh for a fixed degree of freedom.
- By construction, it preserves the second order convergence.
- It is automatic, the only parameter that needs to be controlled is the number of degrees of freedom.
- The method tends to refine the mesh in the hierarchical importance of the solution's gradient. In other words, if new features (with high gradients) appear

in the solution, the mesh will be automatically coarsened in regions with lower gradient and will be refined near the newly emerging features. In this case, for a small number of nodes fixed by the user, the solution will still be well captured although not with the same degree of accuracy.

We summarize in algorithm 2 the steps for mesh adaptation based on given function values.

---

**Algorithm 2:** Mesh adaptation algorithm on a given function  $u$ .

---

**Input:** Input mesh  $\mathcal{H}$ , a target number of nodes  $N$  and a function's nodal values  $u^i$ .

**Output:** An optimal anisotropic mesh  $\tilde{\mathcal{H}}$  well adapted to accurately capture the variation of  $u^i$ .

**begin**

**while** (*not converged*) **do**

        • **for** each node  $X^i$  in  $\mathcal{H}$ , **do**

            – Compute the recovery gradient  $G^i$ .

            – Determine the error estimates  $e_{ij}$  on the edges  $\mathbf{X}^{ij}$  connected to  $X^i$ .

            – Compute the stretching factors  $s_{ij}$  on the edges  $\mathbf{X}^{ij}$ :

                \* Method 1: using the formula for the total error minimization.

                \* Method 2: using the error equi-distribution principle.

            – Construct the metric tensor  $\mathcal{M}^i$ .

        • Generate the optimal mesh  $\tilde{\mathcal{H}}$ .

        • Interpolate  $u$  from  $\mathcal{H}$  to  $\tilde{\mathcal{H}}$ .

        • Test the stopping criterion.

---

In practice, a unit mesh cannot be achieved. What we aim at getting is quasi-unit edges [Frey 01] where:

$$\frac{1}{\sqrt{2}} \leq |\mathbf{X}^{ij}|_{\mathcal{M}^i} \leq \sqrt{2}$$

At the limit, we have  $|\mathbf{X}^{ij}|_{\mathcal{M}^i} = \sqrt{2}$ . Together with the regularization form (2.95), we can determine an upper bound for the error estimates below which we can consider that we have reached a unit mesh. Choosing a threshold value  $\varepsilon_{\max}$ , we get a unit mesh when:

$$\forall i, j, \varepsilon_{\max} |\mathbf{X}^{ij}|_{\mathcal{M}^i} \leq (\sqrt{2})^2 \varepsilon_{\max}$$

This stopping criterion is applied when using the method based on the minimization of the total error over the mesh. On the other hand, when the method based on the error equi-distribution principle is applied, we assume that the mesh has converged to the optimal configuration when the computed homogenous error presents a relative

error between two consecutive remeshings less than the threshold value:

$$\frac{\varepsilon(N)^\zeta - \varepsilon(N)^{\zeta-1}}{\varepsilon(N)^\zeta} \leq \epsilon_{\text{TOL}}, \quad (2.120)$$

where  $\zeta$  is the number of repeated remeshing and  $\epsilon_{\text{TOL}}$  is a certain tolerance.

We will show in chapter 5 how the method can be easily extended to account for several fields in a single metric.

## 2.8 NUMERICAL EXPERIMENTS

In this section we illustrate on the performance of the developed edge-based error estimators and anisotropic mesh adaptation technique on two and three dimensional analytical test cases. We also verify the second order convergence of the method and compare it to other results present in the literature. To stay in the context of numerical examples where the analytical solution is unknown, we do not use the actual gradient of the known function for the computation of the error estimates.

### 2.8.1 Convergence tests on a quadratic function

In order to perform convergence tests, we construct unit meshes with growing complexities:

$$\begin{array}{cccccc} \mathcal{H}_N & , & \mathcal{H}_{2N} & , & \mathcal{H}_{4N} & , & \mathcal{H}_{8N} & , & \mathcal{H}_{16N} \\ \downarrow & , & \downarrow & , & \downarrow & , & \downarrow & , & \downarrow \\ N & , & 2N & , & 4N & , & 8N & , & 16N \end{array}$$

In the numerical examples we take  $N = 1,000$ . Then we compute the actual interpolation error on each of these meshes using a Gauss interpolation formula:

$$\|u - \pi_h u\|_{L^p} = \left( \sum_{K=1}^{n_K} \int_K |u - \Pi_h u|^p \right)^{\frac{1}{p}} = \left( \sum_{K=1}^{n_K} \sum_{i=1}^{n_g} \omega_i |u(\mathcal{G}_i) - \Pi_h(\mathcal{G}_i)|^p \right)^{\frac{1}{p}} \quad (2.121)$$

with  $n_K$  being the number of elements in the mesh and  $n_g$  is the number of Gauss points in element  $K$ .

In this example we address two aspects of the developed anisotropic mesh adaptation algorithms. First we are interested in studying and comparing the three proposed methods:

- Mesh adaptation based on the total interpolation error minimization. We denote the mesh obtained with this method as  $\mathcal{H}^I$ .
- Mesh adaptation based on the privileged interpolation error minimization. We denote the mesh obtained with this method as  $\mathcal{H}^{IP}$ .

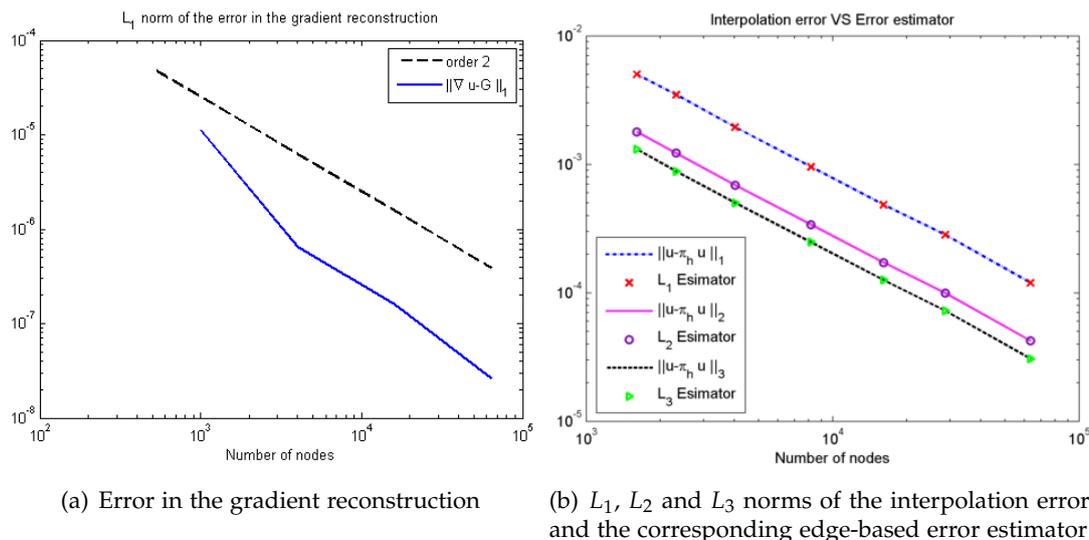


Figure 2.9 – Quadratic function: validation of the gradient reconstruction and the edge-based error estimator.

- Mesh adaptation based on the error equi-distribution. We denote the mesh obtained with this method as  $\mathcal{H}^E$ .
- Mesh adaptation based on the privileged error equi-distribution. We denote the mesh obtained with this method as  $\mathcal{H}^{EP}$ .

The methods are validated in terms of order of convergence, i.e. how accurate is the algorithm, and of efficiency, i.e. how fast does the method reach a unit mesh.

In this test case we first validate the developed gradient reconstruction approach then we evaluate the performance of the derived error estimator and mesh adaptation technique to control the interpolation error in the  $L_p$  norm. For that purpose we consider a quadratic function  $u \in \mathbb{R}$  defined on the domain  $\Omega = [-1.5, 1.5]$  by:

$$u(x, y) = 0.3(x^2 + y^2) \quad (2.122)$$

As the exact gradient of the function is known, we can determine how accurate is the developed gradient recovery technique. To evaluate the accuracy of the proposed reconstruction, we compute the  $L_1$  norm of the error between the exact and the recovered gradients on meshes with increasing complexities.

Figure 2.9(a) presents the variation of the interpolation error's  $L_1$  norm induced by the reconstruction technique on different meshes. We can clearly see how the error changes quadratically as the number of nodes is doubled. We can infer that the proposed recovery technique results in a second order reconstruction of the function's gradient.

We move on now to assess the reliability of the proposed error estimator. To do so, we compare the  $L_p$ , ( $p = 1, 2, 3$ ) norm of the interpolation error to the estimated  $L_p$ , ( $p = 1, 2, 3$ ) norm as discussed in section 2.5.7. Figure 2.9(b) shows a good match-

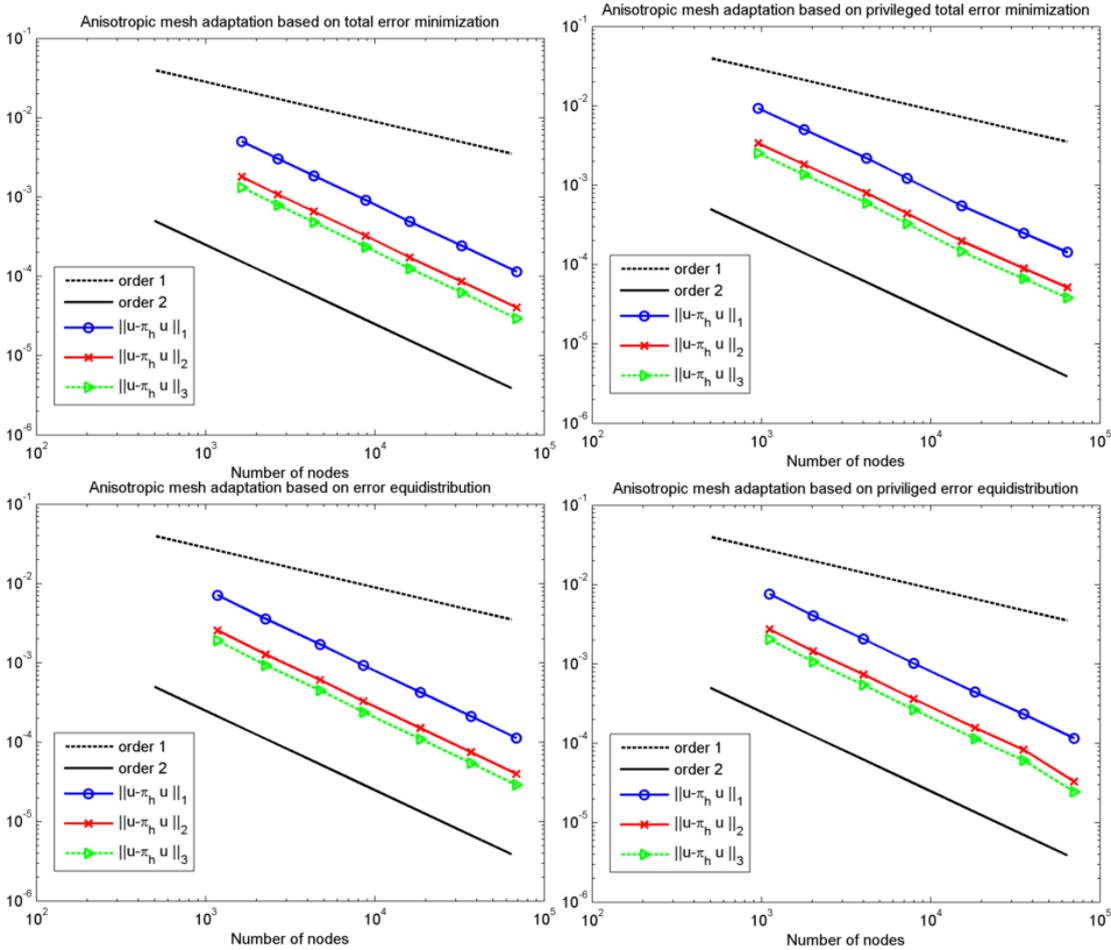


Figure 2.10 – Quadratic function: Mesh convergence for the  $L_1$ ,  $L_2$  and  $L_3$  norms of the interpolation error

ing between the computed interpolation error and the corresponding error estimate. Therefore by limiting the estimated error, we are capable of controlling the  $L_p$  norm of the interpolation error.

Using the developed mesh adaptation algorithms relying on the edge based error estimates, we adapt the mesh and analyze the control of the interpolation error for the  $L_1$ ,  $L_2$  and  $L_3$  norms. We start the computations on a coarse mesh made up of 100 nodes and then refine it according to the proposed adaptive techniques.

Concerning the mesh convergence, for a regular and smooth function as shown in figure 2.10, the theoretical second order is reached whatever the selected norm  $L_p$ ,  $p \in [1, \infty[$ .

## 2.8.2 Numerical validations on functions with steep gradients

Anisotropic mesh adaptation is devised to reproduce the anisotropic features of physical phenomena. However the construction becomes more challenging when sharp angles are present in the adaptation field as the algorithm tends to create elements

with extremely high aspect ratios. In such cases the mesh generator does not produce the optimal unit mesh in one shot, several attempts need to be executed in order to drive the mesh/solution couple toward optimality. So for each number of degrees of freedom the algorithm of adaptation is repeated several times until a unit mesh is obtained, i.e. until the measured interpolation error shows no additional improvement. Note that the number of iterations to reach a unit mesh depends highly on the quality of the initial mesh. The efficiency analysis will be applied on three different functions that vary in stiffness, regularity and scales of variations. In practice, we will consider that we have reached a unit mesh when the relative difference in the interpolation error between two successive meshes is less than 1%.

### 2.8.2.1 Mountains and valleys

Before we proceed into the convergence and efficiency analysis on functions with steep gradients and high anisotropic features, let us give an example showing the sharp anisotropic features of the proposed algorithms. We consider the mountains and valleys function taken from [Borouchaki 01]. A circular domain of radius 3 centered at the origin of the space is considered. We intend to adapt the mesh on the analytical solution defined by:

$$f(x, y) = \exp^{-3((x+0.3)^2+(y+0.2)^2)} - \exp^{-3((x-0.3)^2+(y-0.4)^2)} + \frac{1}{2} \tanh(10 \sin(x^2 - y^2) \cos(x^2 + y^2)) \quad (2.123)$$

This test case is a complicated one as the computational domain's boundary is curved. The initial mesh is isotropic and made up of 1,000 nodes. We iteratively adapt the mesh using the algorithm based on the error equi-distribution. As this example is just for illustrative purposes we do not intend to compare the performance of the different proposed mesh adaptation algorithms. This will be done in the following examples. We present in figure 2.11 four consecutive stages of anisotropic refinement and a zoom on the last stage. It is important to note that all the represented meshes contain roughly the same number of elements set to 7,000 at the beginning of the computations. The zoom shows exactly how anisotropic the mesh can be around obstacles/sharp gradients thanks to this method. We can see how the nodes in the mesh are automatically removed from regions of low function's variations and others are added near the steep gradients. The good orientation of the generated elements allows a good representation of the curvatures. Note that due to the error equi-distribution over the edges, the anisotropic meshes tend to preserve the function's symmetric profile by equally partitioning the nodes over the regions of steep features. Figure 2.12 shows the anisotropic mesh obtained in the reference in eight consecutive mesh adaptations. We showed the mesh at the first, second, and last phase of modification as presented in [Borouchaki 01]. These meshes are made up respectively of 37,070, 67,533, 107,469, and 47,419 nodes whereas the meshes generated by our adaptive algorithm are varying

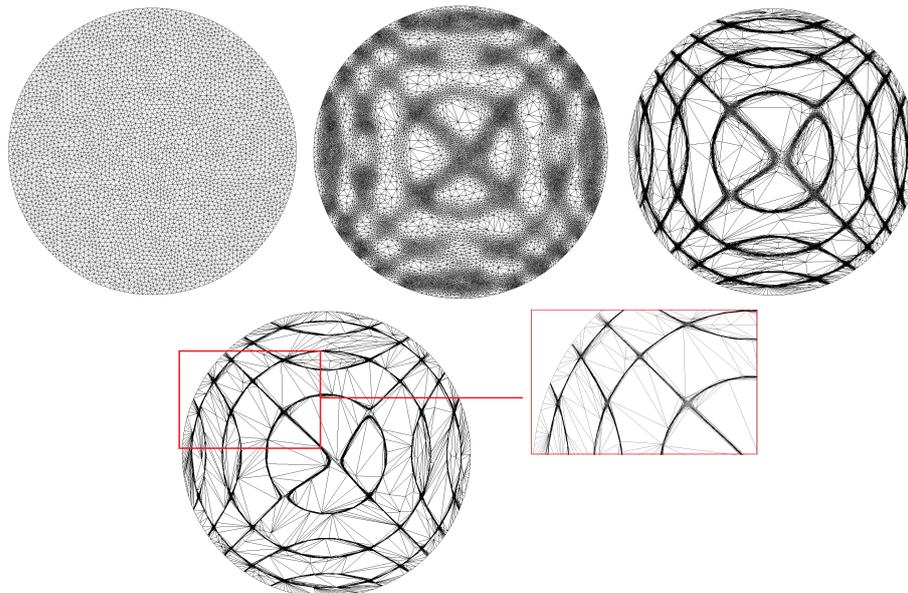


Figure 2.11 – Mountains and valleys: Iterative anisotropic refinement obtained within 4 consecutive iteration of the proposed mesh adaptation algorithm.

around the same degree of freedom ( 7,000). The advantages of iteratively adapting the mesh is clear as the sharpness of the representation is evident in the last iterations, where the anisotropy is more pronounced, whereas the first iteration fails. As shown in figure 2.13, the function computed on an isotropic non-adapted mesh presents an “orange skin”, its surface is not smooth and reflects a poor quality representation whereas the one computed on the generated anisotropic mesh has a high quality, it presents a smooth surface and the mountains’ sharp descents are accurately captured.

### 2.8.2.2 MINES logo

In the view of reflecting how steep can the mesh become, we take the logo of the Ecole des Mines and apply the anisotropic mesh adaptation algorithm for 5 consecutive iterations. The example was performed in 3D with around 30,000 nodes. We present in figure 2.14 the 3D logo together with a cross section along the  $y = 0$  plane showing the capability of the proposed technique to capture and reproduce very sharp angles with a high resolution. One can notice how the elements are extremely stretched and well oriented near the sharp edges.

### 2.8.2.3 Steep hyperbolic tangent function

The objective of this numerical example is to illustrate how the developed metric construction techniques are well defined to permit a good capture of a function’s steep layers. The analytical function that we consider is taken from [Coupez 11]. It is defined

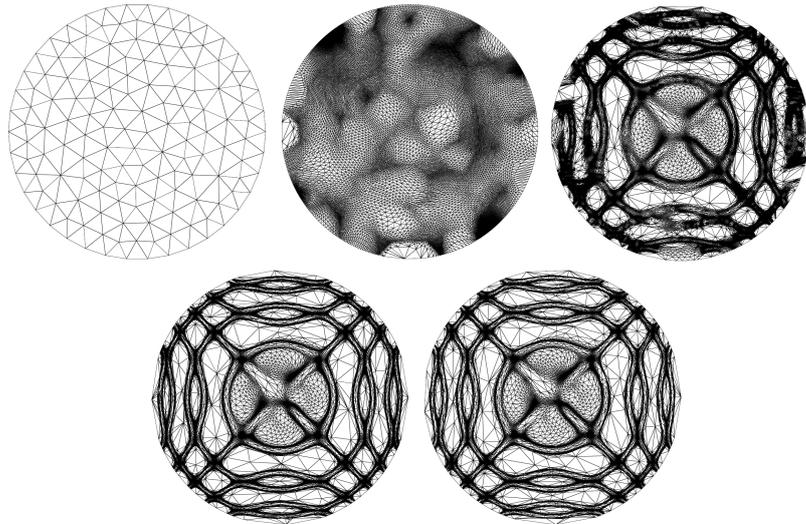


Figure 2.12 – Mountains and valleys: Iterative anisotropic refinement obtained within 8 consecutive iterations in the reference [Borouchaki 01].

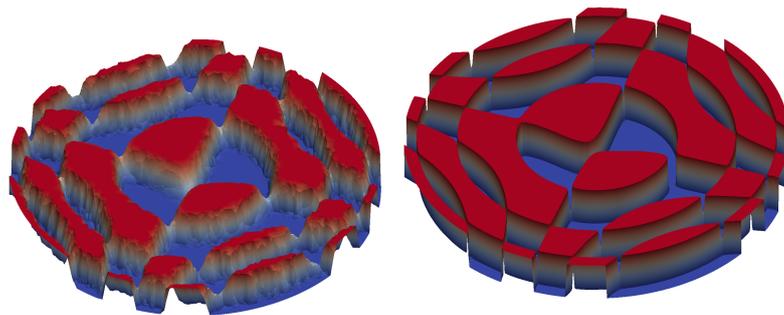


Figure 2.13 – Mountains and valleys: first iteration (left) and last one (right) in the mesh adaptation process.

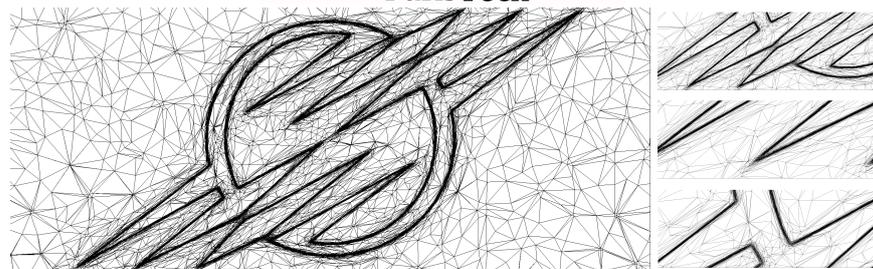


Figure 2.14 – Anisotropic mesh adaptation around the logo of the Ecole des Mines and zoomed snapshots near sharp angles.

on the domain  $[0, 1]^d$  by:

$$\begin{aligned}
 a(\mathbf{x}) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X}\|\right)\right) \\
 b(\mathbf{x}) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X} - \mathbf{C}\|\right)\right) \\
 \mathbf{C} &= \begin{pmatrix} 1 \\ \cdot \\ 1 \end{pmatrix} \\
 u(\mathbf{x}) &= a \circ a(\mathbf{x}) + b \circ b(\mathbf{x})
 \end{aligned} \tag{2.124}$$

The parameter  $E$  varies from 1 to 32; the larger  $E$  is, the sharper the function's gradient is, favoring anisotropic meshing. The edge-based error estimation based on the control of the interpolation error is evaluated for the  $L_1$ ,  $L_2$  and  $L_\infty$  norms. For sake of comparisons on a steep function, we set  $E = 16$  and start the computations on an isotropic mesh made up of 100 nodes. We depict in figure 2.15 the analytical function obtained on an anisotropically adapted mesh (of around 7,000 nodes) with the privileged error equi-distribution technique. Despite the simplicity of this function,

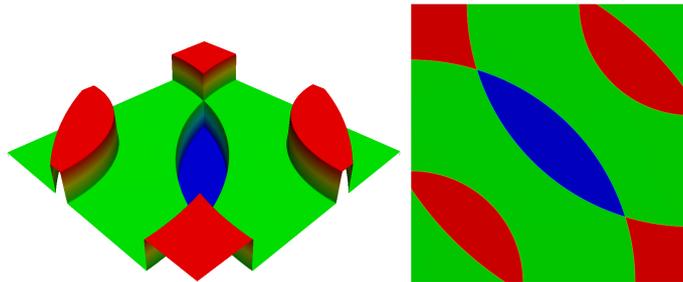


Figure 2.15 – Surface plot (left), and isovalues (right) of the steep radial function described by equation (2.124) on a mesh made up of 7,000 nodes.

it develops complex features that manifest in the interference between two radial gradients forcing isotropic meshing at the intersecting regions. Figure 2.16 shows the adapted meshes ( $\mathcal{H}^I$ ,  $\mathcal{H}^{IP}$ ,  $\mathcal{H}^E$  and  $\mathcal{H}^{EP}$ ) after 7 iteration of the adaptation algorithm for mesh complexities of 7,000(left) and 20,000(right) nodes. We can identify from the mesh plots that the adaptations based on the privileged orientation scheme are capable of converging faster toward the optimal mesh whereas the algorithms based on the averaged length distribution tensor are still driving the mesh toward optimality and did not reach a good representation yet. We highlight that the method based on the error equi-distribution converges even faster than the one derived from the total error minimization algorithm. However for a higher number of degrees of freedom, 7 iterations were enough to converge to optimality using any of the developed methods. We can detect, on the converged meshes, how the elements are well oriented and stretched along the tangential direction to allow a steep capture of the function's circular gradient. The nodes are being automatically redistributed with a higher density in the vicinity of sharp gradients.

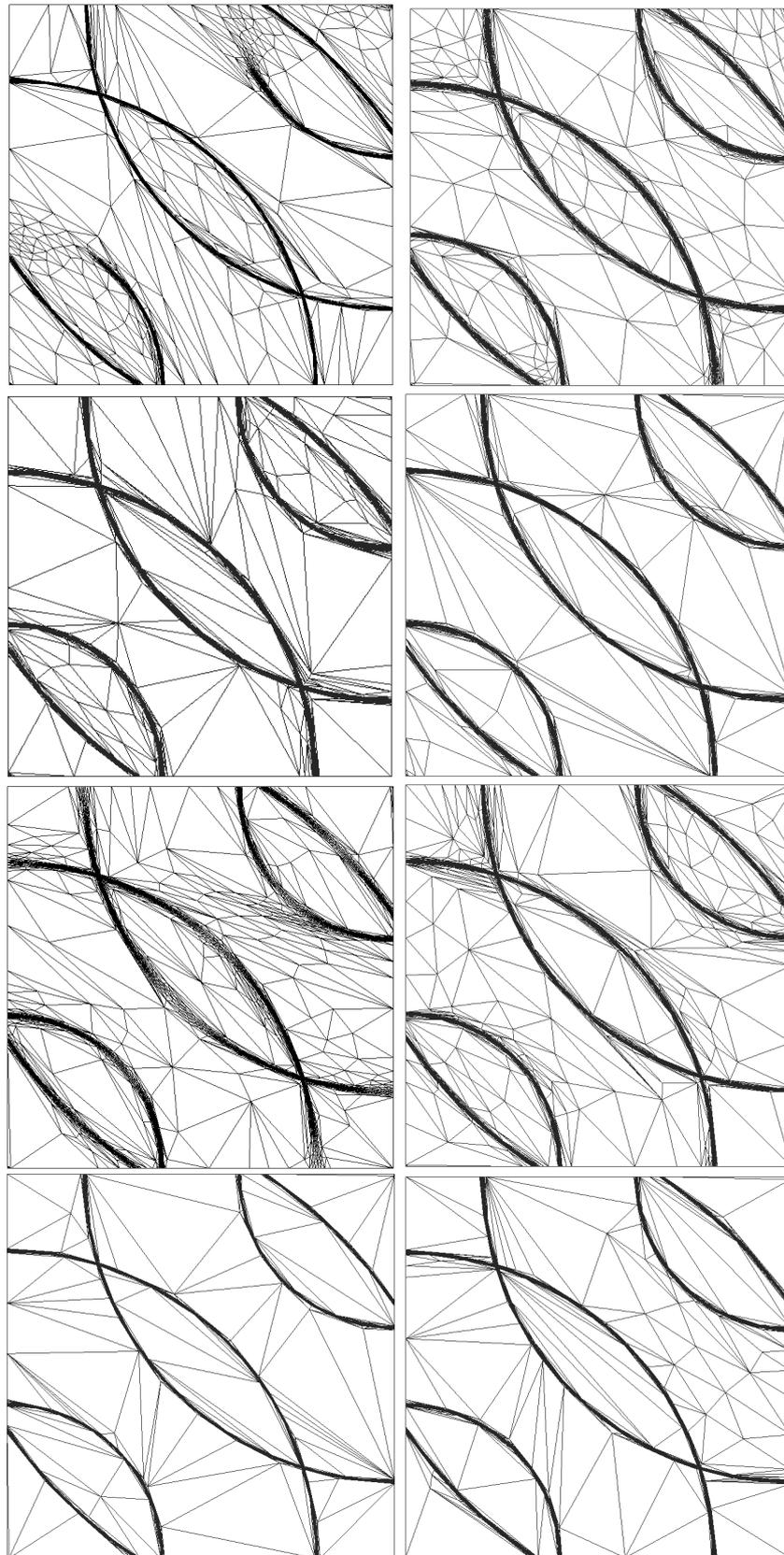


Figure 2.16 – Steep radial function: anisotropic meshes ( $\mathcal{H}^I$ ,  $\mathcal{H}^{IP}$ ,  $\mathcal{H}^E$ , and  $\mathcal{H}^{EP}$  respectively from top to bottom) obtained with around 7,000(left) and 20,000(right) nodes after 7 successive adaptations.

| Adaptation         | $\ u - \pi_h u\ _1$ | $\ u - \pi_h u\ _2$ |
|--------------------|---------------------|---------------------|
| $\mathcal{H}^I$    | $1.15e^{-3}$        | $8.26e^{-3}$        |
| $\mathcal{H}^{IP}$ | $3.97e^{-4}$        | $1.26e^{-3}$        |
| $\mathcal{H}^E$    | $8.72e^{-4}$        | $1.92e^{-3}$        |
| $\mathcal{H}^{EP}$ | $2.38e^{-4}$        | $5.11e^{-4}$        |

Table 2.1 – Steep radial function: statistics on the interpolation errors for the different meshes generated with 7,000 nodes after 7 successive adaptations.

| # Nodes | $\mathcal{H}^I$ | $\mathcal{H}^{IP}$ | $\mathcal{H}^E$ | $\mathcal{H}^{EP}$ |
|---------|-----------------|--------------------|-----------------|--------------------|
| 1000    | 21              | 15                 | 22              | 17                 |
| 2000    | 17              | 8                  | 16              | 8                  |
| 4000    | 16              | 8                  | 15              | 7                  |
| 8,000   | 14              | 7                  | 13              | 7                  |
| 16,000  | 6               | 6                  | 7               | 4                  |
| 32,000  | 5               | 5                  | 5               | 4                  |
| 64,000  | 5               | 5                  | 5               | 4                  |

Table 2.2 – Steep radial function: statistics on the number of mesh adaptations needed to drive the mesh toward unity.

We summarize in table 2.1 the history of the  $L_1$  and  $L_2$  norms of the interpolation error on the anisotropically adapted meshes using 7,000 nodes and after 7 iterations of the adaptation algorithm. We can notice that the privileged adaptation minimizes better the interpolation error. This result is in correlation with the mesh plots in figure 2.16 as it renders a faster convergence toward the optimal mesh. In order to evaluate the mesh convergence rates of the proposed algorithms in presence of sharp gradients, we have conducted convergence studies for the  $L_1$  and  $L_2$  norms of the interpolation error. For that purpose, starting with the same initial mesh of 100 nodes, we have applied each of the edge-based anisotropic mesh adaptation algorithms iteratively until the convergence of the couple solution/mesh. Figure 2.17 summarizes the convergence order estimates. We can observe that a poor convergence is obtained on isotropically refined meshes. On the contrary, as expected from the theoretical analysis, the different proposed techniques of anisotropic mesh adaptation allow an asymptotic recovery of the second order convergence. While 8,000 nodes were needed for the mesh to converge in the  $L_2$  norm, a faster capture was established in the  $L_1$  norm with only 4,000 nodes. This finding is in accordance with the fact that the  $L_1$  norm requires less regularity than the other norms. We provide in table 2.2 a synthesis on the number of mesh adaptations needed to drive the mesh toward unity. We highlight the fewer number of meshing iterations as the imposed number of nodes is increased; in other words, for a sufficient number of degrees of freedom, a fast convergence is achieved. In this test case, we have validated the accuracy and efficiency of the developed anisotropic mesh adaptation algorithms. The proposed a posteriori edge-based error estimation proved to be a robust indicator of a solution's anisotropy.

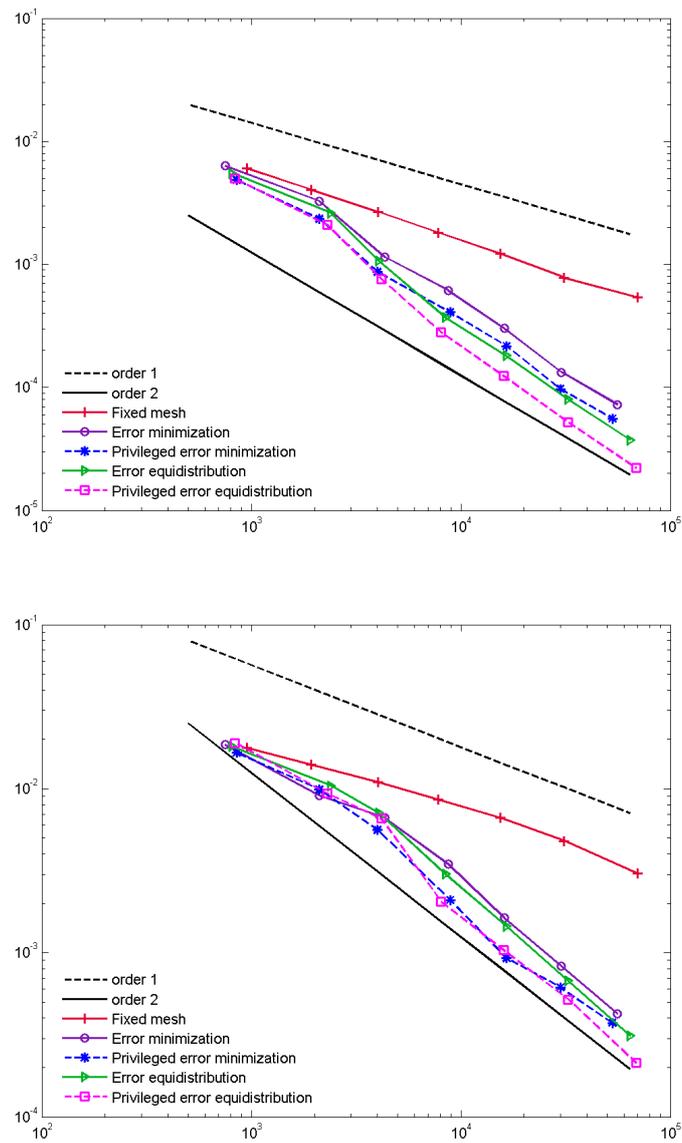


Figure 2.17 – Steep radial function: convergence analysis on the  $L_1$  and  $L_2$  norms of the interpolation error of the analytical function defined by (2.124).

We have pointed out the good results generated by both techniques of metric construction: the one based on the total error minimization and the one involving an error equi-distribution. The two methods achieved an asymptotic second order convergence and provided good efficiency in driving the mesh toward unity when a sufficient number of nodes is employed. We note also that faster and better minimization were obtained by the metric based on the equi-distribution principle.

To avoid result redundancy, from this point on, we will assess only this algorithm as it will be the one used for generating anisotropic meshes throughout the rest of this thesis.

### 2.8.3 Numerical validations on functions with multiscale variations

We intend in this example to demonstrate the capability of the developed anisotropic mesh adaptation algorithm to capture the different scales present in a field of interest. For that purpose, we consider a (2D) function taken from [Loseille 09] that exhibits variations at small and large scales with respective amplitudes 0.01 and 1. The analytical function is defined on the domain  $[-1, 1] \times [-1, 1]$  by:

$$u(x, y) = \begin{cases} \sin(50xy) & \text{if } \frac{\pi}{50} \leq xy < 2\frac{\pi}{50} \\ 0.01 \sin(50xy) & \text{otherwise.} \end{cases} \quad (2.125)$$

A plot of isovalues, a surface profile and a cross section along the  $x = y$  axis obtained with the privileged anisotropic mesh adaptation with 7,000 nodes are shown in figure 2.18. We can clearly identify the good capture of the small wiggles as well as the large-scale sinusoidal wave. Unlike the metric construction based on the error equi-distribution and the average length distribution tensor where the nodes are localized around the steep gradient, higher weights were associated to the small scales when using the privileged length distribution tensor. A second order convergence is reached using an averaged or a privileged length distribution tensor. However, in accordance with the reference, around 10,000 nodes were needed to capture that order. We summarize the convergence history in figure 2.19. We can distinguish 3 parts in the graph. A phase of second order convergence, delimited by a mesh complexity of around 2,000 nodes, where the steep gradients are well captured whereas the small scale sinusoidal oscillations are not detected. A phase of slower convergence rate comes after where the mesh detects the fine-scales but the number of degrees of freedom, between 2,000 and 10,000, is not sufficient to represent them and hence a significant error is highlighted on the mesh. Finally, for a higher number of nodes, a good capture of the solution's fluctuations is established hence regaining the second order convergence. We depict in figure 2.20 the meshes generated by the metric construction based on the equi-distribution principle and made up of around 9,000 nodes. It shows the good orientation and distribution of the mesh elements that present stretched edges parallel to the sinusoidal gradient and small mesh sizes in its orthogonal direction. Note the high concentration of the elements in the localized zones around the steep layer when

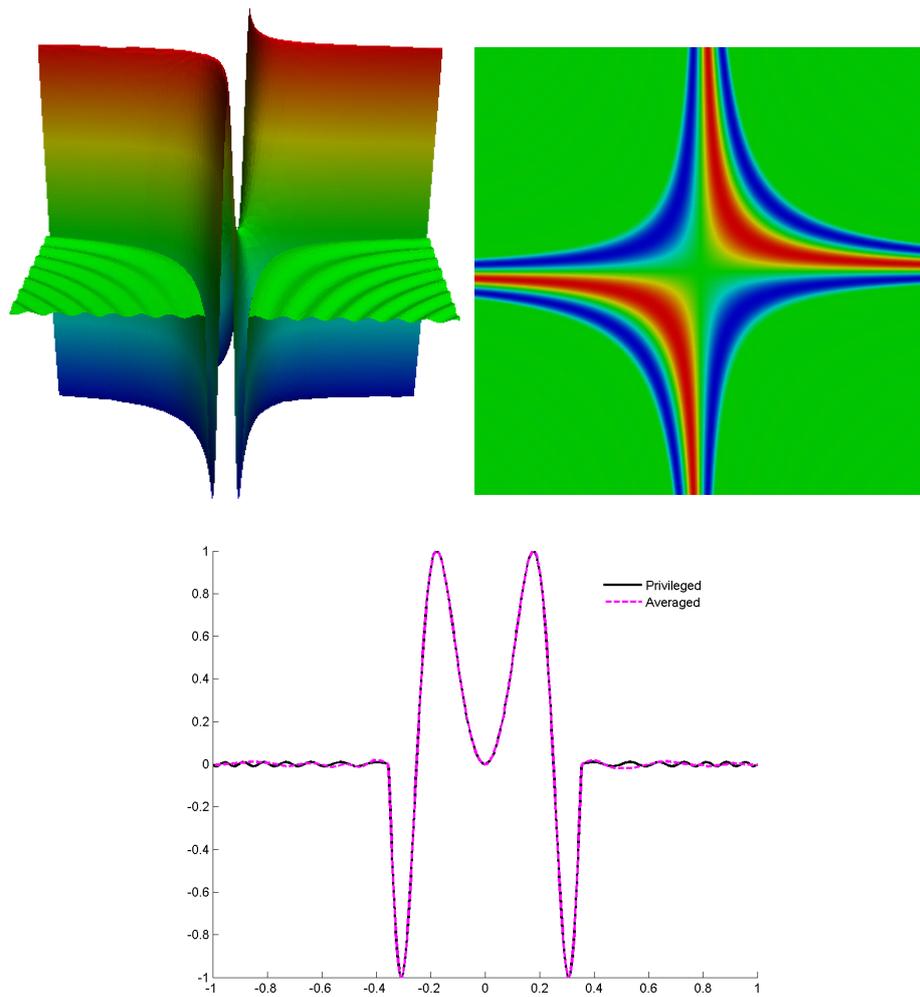


Figure 2.18 – Multiscale function: surface plot (top-left), function isovalues (top-right) and a cross section along the  $x = y$  axis (bottom).

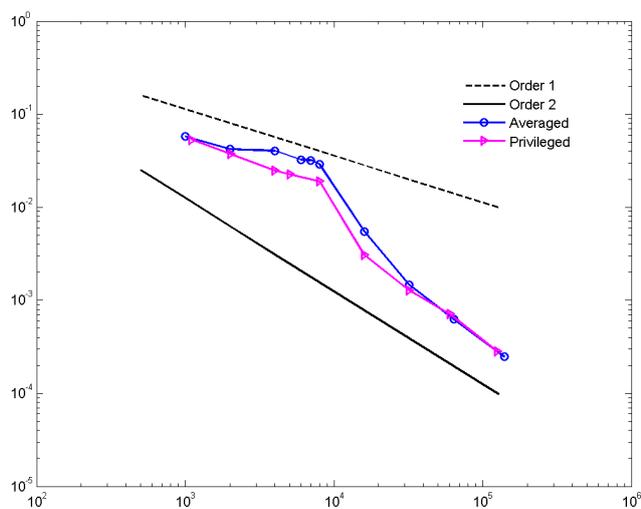


Figure 2.19 – Multiscale function: Convergence analysis on the  $L_1$  norm of the interpolation error.

using the averaged length distribution tensor.

While a slight difference can be presumed from the previous examples between the two versions of the proposed mesh adaptation, a correlation between the sensitivity to small fluctuations and the weighted length distribution tensor is evident from this numerical test case. The choice between the two algorithms depends on the desired sensitivity to small-scales. So if strong gradients are to be privileged with respect to small ones, the averaged length distribution tensor works very well. On the other hand, if one is interested in capturing all scales of a phenomena then the privileged metric construction is encouraged. Note that both methods have almost the same performance when the number of nodes in the mesh is sufficiently high. It

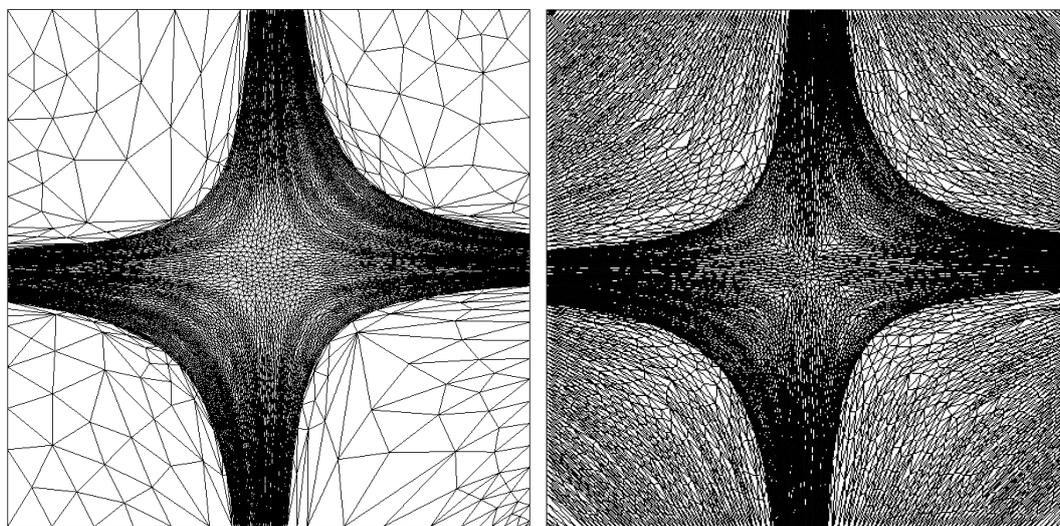


Figure 2.20 – Multiscale function: meshes obtained with 9,000 nodes using the averaged (left) and the privileged (right) length distribution tensor for metric construction.

is important to mention that the computational cost associated with the computation of the scaling factors for the privileged length distribution tensor is negligible highlighting once again its potential in improving the accuracy/efficiency aspect of the computations.

## 2.9 CONCLUSION

In this chapter, we have pursued the work in [Coupez 11] to develop a new route for building anisotropic meshes. Starting with the fundamental equivalence between optimal and unit meshes, we aimed at building unit meshes. However, doing so on the canonical Euclidean space yields an isotropic mesh that is fine everywhere in the domain and thus implies a considerable computational cost. The idea therefore was to build this unit mesh in the Riemannian metric space. The resulting mesh would be anisotropic and well adapted in the canonical space. We have constructed the metric tensor that defines the Riemannian metric space at each node  $X^i$  so that to prescribe

along each edge direction connected to  $X^i$  a scaling factor. This stretching factor was determined in terms of the error induced along the edges. A new error estimator was developed and it was expressed as the projected function's gradient along the edges. Nevertheless, since the method was constructed to be applied to any field of interest, in particular to the solution of PDEs, and since for most physical problems, the exact solution is not known a priori nor does its gradient, we have derived a second order recovery approach for this field. Moreover, we have proposed two possible approaches for computing the stretching factors: the first one intends to minimize the total error over the edges of the computational domain and the second equi-distributes the error over these edges. An extension to provide control on the  $L^p$  norm of the interpolation error was established. Furthermore, we have enriched the anisotropy of the mesh by introducing more weights to the edges in the direction of the variable's gradient. The theoretical investigation of this idea will be the subject of future contributions. A key feature of the proposed adaptation method is its ability to enhance the accuracy while preserving a fixed number of nodes which is a great advantage for industrial applications. We have validated the accuracy and efficiency of the developed error estimator and anisotropic mesh adaptation technique on several analytical functions presenting smooth variations, sharp discontinuities, and multiscale features. The objective in the upcoming chapter is to apply the developed adaptation tool in the context of computational fluid dynamic simulations.

## 2.10 RÉSUMÉ FRANÇAIS

Dans ce chapitre, nous avons poursuivi le travail commencé dans [Coupez 11] dans le but de développer une nouvelle méthode pour la construction de maillages anisotropes. Partant du paradigme qu'un maillage anisotrope est équivalent à un maillage uniforme unité avec différentes définitions de distance, nous avons cherché à construire des maillages unités dans des espaces métriques Riemanniens. Le maillage résultant serait anisotrope et bien adapté dans l'espace canonique. Nous avons alors construit le tenseur métrique associé à l'espace métrique sur chaque nœud du maillage de sorte à ce qu'à chaque issue de ce nœud, un facteur d'étirement soit imposé. Ce facteur a été déterminé en fonction de l'erreur induite le long des arrêtes connectées au nœud. Un nouvel estimateur d'erreur a été proposé et a été exprimé en fonction du gradient du paramètre de remaillage projeté sur les arrêtes. Néanmoins, étant donné que la méthode a été construite de manière à s'appliquer sur tous les domaines et pour tout phénomène physique, et comme pour la plupart des problèmes physiques, la solution exacte, et donc son gradient, n'est pas connue, une méthode de reconstruction du second ordre du gradient a été proposée. De plus, nous avons proposé deux approches pour le calcul des facteurs d'étirement. La première consiste à minimiser l'erreur totale induite sur les arrêtes du maillage et la deuxième vise à équi-distribuer l'erreur sur ces arrêtes. Une extension pour contrôler la norme  $L^p$  de l'erreur d'interpolation a été proposée et démontrée. Nous avons aussi proposé d'introduire des poids sur

les arrêtes favorisant l'étirement de ces derniers dans le sens orthogonal au fort gradient du champ étudié. Une caractéristique essentielle de la méthode développée est sa capacité d'optimiser le maillage tout en conservant un nombre fixe de nœuds ce qui constitue un grand avantage pour les applications industrielles. Dans la dernière partie de ce chapitre, nous avons validé la précision et l'efficacité de l'estimateur d'erreur proposé et l'adaptation de maillage anisotrope développée sur plusieurs fonctions analytiques présentant de faibles et fortes variations ainsi que des évolutions multi-échelles. L'objectif dans le chapitre suivant est d'appliquer l'outil d'adaptation dans le cadre de simulations CFD diverses.

# COUPLING THE ANISOTROPIC MESH ADAPTATION WITH CFD PROBLEMS

“Progress lies not in enhancing what is, but in advancing toward what will be.”

Gibran

## CONTENTS

---

|       |                                                                              |     |
|-------|------------------------------------------------------------------------------|-----|
| 3.1   | A PRIORI ERROR ANALYSIS . . . . .                                            | 145 |
| 3.1.1 | convection-diffusion-reaction equation . . . . .                             | 145 |
| 3.1.2 | Incompressible Navier-Stokes equations . . . . .                             | 147 |
| 3.2   | TUNING WITH STABILIZED FINITE ELEMENT METHODS . . . . .                      | 148 |
| 3.2.1 | Illustrative example . . . . .                                               | 151 |
| 3.2.2 | Validation of the characteristic lengths choice . . . . .                    | 152 |
| 3.2.3 | Navier-Stokes equations with dominant viscous terms . . . . .                | 152 |
| 3.2.4 | Convection-Diffusion-Reaction equation with dominant viscous terms . . . . . | 158 |
| 3.3   | APPLICATIONS TO CONVECTION-DIFFUSION PROBLEMS . . . . .                      | 160 |
| 3.3.1 | Numerical experiments on the steady convection-diffusion problem . . . . .   | 161 |
| 3.4   | APPLICATIONS TO INCOMPRESSIBLE FLOW PROBLEMS . . . . .                       | 169 |
| 3.4.1 | Driven flow cavity problem (2D) . . . . .                                    | 170 |
| 3.4.2 | Driven flow cavity problem (3D) . . . . .                                    | 176 |
| 3.4.3 | Application to a 3D coupled heat transfer and fluid flows problem . . . . .  | 180 |
| 3.5   | CONCLUSION . . . . .                                                         | 183 |
| 3.6   | RÉSUMÉ FRANÇAIS . . . . .                                                    | 183 |

---

**D**ESPITE the growing computational power, anisotropic mesh adaptation seems mandatory to reduce the complexity of numerical simulations especially when complex 3D industrial applications are involved. Indeed, it is highly desirable to obtain the most accurate solution with the least computational time and cost. In the previous chapter, we have developed a metric based anisotropic mesh adaptation technique. Its accuracy, efficiency and robustness were validated on analytical test cases. The method relies on an a posteriori edge-based error estimator without any a priori assumptions on the solution’s behavior or the problem at hand. Therefore it can be

directly applied, without modification, to any field of interest. In this chapter, we are interested in applying and assessing the proposed algorithm in the context of computational fluid dynamics and heat transfer simulations. In chapter 5, we illustrate on how the algorithm can be generalized to naturally account for several fields and drive the mesh toward optimality, hence robustly capturing anisotropic features developed by the fields of interest. In the metric based anisotropic mesh adaptation framework introduced previously, the idea was to construct a mesh that minimizes the interpolation error of a given function. The objective now is to replace the given function by the solution of a PDE. From the discrete solution of the problem and the length distribution tensor, we reconstruct the gradient of this field then we evaluate the estimated error along the edges of the current mesh. Based on the equi-distribution principle a metric tensor is constructed and given to the mesh generator leading to the optimal mesh minimizing the interpolation error of the solution. Once the new mesh is obtained, the problem variables will be interpolated on the new mesh to proceed with computations. This process is repeated until the convergence of the mesh-solution couple. On the other hand, the existing stabilized finite element methods can be leveraged and require only minor modifications to accommodate with the anisotropy of the mesh. In this chapter, we reconsider in section 1 the Navier-Stokes and the convection-diffusion-reaction problems and perform an error analysis pointing out the relation between the approximation and the interpolation errors. We move on next to illustrate on the tuning of the stabilization methods in order to account for highly stretched elements. In the following section, we provide numerical examples reflecting the performance, accuracy and efficiency of the developed anisotropic mesh adaptation in simulating heat transfer and fluid flow problems.

### 3.1 A PRIORI ERROR ANALYSIS

We aim in this section at providing an a priori finite element analysis on the convection-diffusion-reaction problem and the Navier-Stokes equations. We follow the lines in [Belme 11, Courty 06] to derive the error estimates. In the reference [Belme 11, Courty 06], the analysis was limited to the Poisson's equation.

#### 3.1.1 convection-diffusion-reaction equation

We start by considering the convection-diffusion-reaction equation with a velocity field  $\mathbf{v}$ , a diffusion factor  $\kappa$  and a reaction term  $\sigma$ :

$$\begin{cases} \partial_t u + \mathbf{v} \cdot \nabla u - \nabla \cdot (\kappa \nabla u) + \sigma u = f, & \text{in } \Omega, \\ u(., 0) = 0 & \text{in } \Omega, \\ u = 0, & \text{on } \Gamma, \end{cases} \quad (3.1)$$

whose weak formulation writes as:

$$\left( \frac{\partial u}{\partial t}, w \right) + (\mathbf{v} \cdot \nabla u, w) - (\nabla \cdot (\kappa \nabla u), w) + (\sigma u, w) = (f, w) \quad \forall w \in H_0^1(\Omega) \quad (3.2)$$

The approximation error can be decomposed into two parts:

$$u_h - u = \underbrace{u_h - \Pi_h u}_{\text{Implicit error}} + \underbrace{\Pi_h u - u}_{\text{Interpolation error}} \quad (3.3)$$

The implicit error belongs to the discrete approximation space  $\mathcal{V}_h = H_0^1 \subset H^1(\Omega)$  of piecewise linear functions defined on the elements of the mesh  $\mathcal{H}_h$ .

Define  $\bar{\Pi}_h$  to be a projection operator from  $\mathcal{V}$  onto  $\mathcal{V}_h$ . As suggested in [Belme 11], since  $u \in \mathcal{V}$  is not necessarily smooth, one can choose the Clément interpolation operator. Taking  $w \in \mathcal{V}$ , its projection will be  $\bar{\Pi}_h w \in \mathcal{V}_h$ . Therefore, the weak formulation reads as:

$$\begin{aligned} \left( \frac{\partial u_h}{\partial t}, \bar{\Pi}_h w \right) + (\mathbf{v} \cdot \nabla u_h, \bar{\Pi}_h w) - (\nabla \cdot (\kappa \nabla u_h), \bar{\Pi}_h w) + (\sigma u_h, \bar{\Pi}_h w) \\ = (f, \bar{\Pi}_h w) \quad \forall w \in H_0^1(\Omega) \end{aligned} \quad (3.4)$$

The adjoint of the projection operator implies:

$$\begin{aligned} \left( \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} \right) u_h, w \right) + (\bar{\Pi}_h^* (\mathbf{v} \cdot \nabla) u_h, w) - (\bar{\Pi}_h^* (\nabla \cdot (\kappa \nabla)) u_h, w) \\ + (\bar{\Pi}_h^* (\sigma \mathbb{I}) u_h, w) = (\bar{\Pi}_h^* ((\mathbb{I}) f), w) \quad \forall w \in H_0^1(\Omega) \end{aligned} \quad (3.5)$$

where,  $\mathbb{I}$  denotes the identity tensor. Considering the dual space of  $\mathcal{V}$ ,  $\mathcal{V}'$ , equation 3.5

yields:

$$\bar{\Pi}_h^* \left( \frac{\partial}{\partial t} \right) u_h + \bar{\Pi}_h^* (\mathbf{v} \cdot \nabla) u_h - \bar{\Pi}_h^* (\nabla \cdot (\kappa \nabla)) u_h + \bar{\Pi}_h^* (\sigma \mathbb{I}) u_h = \bar{\Pi}_h^* ((\mathbb{I}) f) \quad \in \mathcal{V}' \quad (3.6)$$

and equivalently:

$$\begin{aligned} & \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} \right) u_h - \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} \right) \bar{\Pi}_h u - \bar{\Pi}_h^* (\nabla \cdot (\kappa \nabla)) u_h + \bar{\Pi}_h^* (\nabla \cdot (\kappa \nabla)) \bar{\Pi}_h u + \bar{\Pi}_h^* (\mathbf{v} \cdot \nabla) u_h \\ & \quad - \bar{\Pi}_h^* (\mathbf{v} \cdot \nabla) \bar{\Pi}_h u + \bar{\Pi}_h^* (\sigma \mathbb{I}) u_h - \bar{\Pi}_h^* (\sigma \mathbb{I}) \bar{\Pi}_h u \\ & = (\bar{\Pi}_h^* ((\mathbb{I}) f) - \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} \right) \bar{\Pi}_h u + \bar{\Pi}_h^* (\nabla \cdot (\kappa \nabla)) \bar{\Pi}_h u - \bar{\Pi}_h^* (\mathbf{v} \cdot \nabla) \bar{\Pi}_h u - \bar{\Pi}_h^* (\sigma \mathbb{I}) \bar{\Pi}_h u) \quad \text{in } \mathcal{V}' \end{aligned} \quad (3.7)$$

Rearranging the terms, equation (3.7) reads in  $\mathcal{V}'$  as:

$$\begin{aligned} & \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} - \nabla \cdot (\kappa \nabla) + (\mathbf{v} \cdot \nabla + \sigma \mathbb{I}) \right) (u_h - \bar{\Pi}_h u) \\ & = \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} - \nabla \cdot (\kappa \nabla) + (\mathbf{v} \cdot \nabla + \sigma \mathbb{I}) \right) (u - \bar{\Pi}_h u) \end{aligned} \quad (3.8)$$

with  $\bar{\Pi}_h^* \left( \frac{\partial}{\partial t} - \nabla \cdot (\kappa \nabla) + (\mathbf{v} \cdot \nabla + \sigma \mathbb{I}) \right)$  invertible on adhoc spaces according to [Courty 05, Bikhentaev 12]. Consequently, we obtain:

$$(u_h - \bar{\Pi}_h u) = \left( \begin{array}{c} \bar{\Pi}_h^* \left( \frac{\partial}{\partial t} - \nabla \cdot (\kappa \nabla) + (\mathbf{v} \cdot \nabla + \sigma \mathbb{I}) \right)^{-1} \bar{\Pi}_h^* \\ \cdot \left( \frac{\partial}{\partial t} - \nabla \cdot (\kappa \nabla) + (\mathbf{v} \cdot \nabla + \sigma \mathbb{I}) \right) (u - \bar{\Pi}_h u) \end{array} \right) \quad (3.9)$$

and in variational form:

$$\begin{aligned} & \left( \frac{\partial (u_h - \bar{\Pi}_h u)}{\partial t}, \bar{\Pi}_h w \right) - (\nabla \cdot (\kappa \nabla (u_h - \bar{\Pi}_h u)), \bar{\Pi}_h w) + (\mathbf{v} \cdot \nabla (u_h - \bar{\Pi}_h u), \bar{\Pi}_h w) \\ & + (\sigma (u_h - \bar{\Pi}_h u), \bar{\Pi}_h w) = \left( \frac{\partial (u - \bar{\Pi}_h u)}{\partial t}, \bar{\Pi}_h w \right) - (\nabla \cdot (\kappa \nabla (u - \bar{\Pi}_h u)), \bar{\Pi}_h w) \\ & \quad + (\mathbf{v} \cdot \nabla (u - \bar{\Pi}_h u), \bar{\Pi}_h w) + (\sigma (u - \bar{\Pi}_h u), \bar{\Pi}_h w) \quad \forall w \in \mathcal{V} \end{aligned} \quad (3.10)$$

When  $u$  and  $w$  are smooth, it can be shown using a density argument that the projection operator can be taken to be the classical interpolation operator [Courty 05, Courty 06]. This concludes the a priori analysis on the relation between the implicit and thus the approximation error and the interpolation error.

### 3.1.2 Incompressible Navier-Stokes equations

We reconsider now the incompressible Navier-Stokes equations and follow the same analysis to relate the implicit error to the interpolation one. Without loss of generality, we assume zero-Dirichlet boundary condition.

$$\begin{aligned} \rho(\partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) - 2\nabla \cdot \mu \boldsymbol{\varepsilon}(\mathbf{v}) + \nabla p &= \mathbf{f} \quad \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{v} &= 0 \quad \text{in } \Omega \times (0, T) \end{aligned} \quad (3.11)$$

where  $\mathbf{v}$  and  $p$  are the desired velocity and pressure fields. Inhere  $\mu$  is the dynamic viscosity,  $\boldsymbol{\varepsilon}(\mathbf{v})$  the strain rate tensor,  $\rho$  the fluid density and  $\mathbf{f}$  a given vector of external forces acting on the flow. As presented in chapter 1, the weak form of (3.11) consists in finding  $(\mathbf{v}, p) \in (\mathcal{V}, \mathcal{Q})$  such that:

$$\begin{cases} \left( \rho \frac{\partial \mathbf{v}}{\partial t}, \mathbf{w} \right) + (\rho \mathbf{v} \cdot \nabla \mathbf{v}, \mathbf{w}) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}) : \boldsymbol{\varepsilon}(\mathbf{w})) - (p, \nabla \cdot \mathbf{w}) = (f, \mathbf{w}) & \forall \mathbf{w} \in \mathcal{V}^0 \\ (\nabla \cdot \mathbf{v}, q) = 0 & \forall q \in \mathcal{Q} \end{cases} \quad (3.12)$$

An implicit time stepping scheme is first applied resulting in a non-linear system of equations each time-step. The nonlinear term is then linearized using a Newton-Raphson linearization and keeping terms only to first order at the  $i^{\text{th}}$  iteration:

$$(\mathbf{v} \cdot \nabla \mathbf{v}, \mathbf{w})^i \approx (\mathbf{v}^i \cdot \nabla \mathbf{v}^{i-1}, \mathbf{w}) + (\mathbf{v}^{i-1} \cdot \nabla \mathbf{v}^i, \mathbf{w}) - (\mathbf{v}^{i-1} \cdot \nabla \mathbf{v}^{i-1}, \mathbf{w}). \quad (3.13)$$

For any  $\mathbf{v} \in \mathcal{V}$ , the projection operator  $\bar{\Pi}_h^v$  associates  $\bar{\Pi}_h^v \mathbf{w} \in \mathcal{V}$  and for any  $p \in \mathcal{Q}$ , the projection operator  $\bar{\Pi}_h^p$  associates  $\bar{\Pi}_h^p q \in \mathcal{Q}$ . Hence, the variational formulation using a finite element discretization reads as:

$\forall q \in \mathcal{Q}$  and  $\forall \mathbf{w} \in \mathcal{V}^0$ ,

$$\begin{cases} \left( \rho \frac{\partial \mathbf{v}_h}{\partial t}, \bar{\Pi}_h^v \mathbf{w} \right) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \bar{\Pi}_h^v \mathbf{w}) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\bar{\Pi}_h^v \mathbf{w})) - (p_h, \nabla \cdot \bar{\Pi}_h^v \mathbf{w}) = (f, \bar{\Pi}_h^v \mathbf{w}) \\ (\nabla \cdot \mathbf{v}_0, \bar{\Pi}_h^p q) = 0. \end{cases} \quad (3.14)$$

Using the adjoint of the projection operator and performing the Newton-Raphson linearization on the nonlinear term imply:

$\forall q \in \mathcal{Q}$  and  $\forall \mathbf{w} \in \mathcal{V}^0$ ,

$$\begin{cases} \left( \bar{\Pi}_h^{v,*} \left( \rho \frac{\partial}{\partial t} \right) \mathbf{v}_h, \mathbf{w} \right) + (\bar{\Pi}_h^{v,*} (\rho \mathbb{I} \cdot \nabla \mathbf{v}^{i-1}) \mathbf{v}_h^i, \mathbf{w}) + (\bar{\Pi}_h^{v,*} (\rho \mathbf{v}_h^{i-1} \mathbb{I} \cdot \nabla) \mathbf{v}_h^i, \mathbf{w}) \\ - \left( \bar{\Pi}_h^{v,*} (\mathbb{I} \mathbf{v}_h^{i-1} \cdot \nabla \mathbf{v}_h^{i-1}, \mathbf{w}) \right) + (\bar{\Pi}_h^{v,*} (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\mathbf{w})) - (\bar{\Pi}_h^{v,*} (\mathbb{I}) p_h, \nabla \cdot \mathbf{w})) \\ = (\bar{\Pi}_h^{v,*} (\mathbb{I}) f, \mathbf{w}) \\ (\bar{\Pi}_h^{v,*} (\nabla \cdot) \mathbf{v}_h, q) = 0 \end{cases} \quad (3.15)$$

Proceeding in the same way as for the convection-diffusion-reaction problem, one gets in the dual spaces  $\mathcal{V}'$  and  $\mathcal{Q}'$ :

$$\begin{cases} \overline{\Pi}_h^{v,*} \left( \rho \frac{\partial}{\partial t} + \rho \mathbb{I} \cdot \nabla \mathbf{v}^{i-1} \rho \mathbf{v}^{i-1} \mathbb{I} \cdot \nabla + 2\mu \boldsymbol{\varepsilon} \right) (\mathbf{v}_h - \overline{\Pi}_h^v \mathbf{v}) - \overline{\Pi}_h^{v,*} (\mathbb{I})(p_h - \overline{\Pi}_h^v p) \\ = \overline{\Pi}_h^{v,*} \left( \rho \frac{\partial}{\partial t} + \rho \mathbb{I} \cdot \nabla \mathbf{v}^{i-1} \rho \mathbf{v}^{i-1} \mathbb{I} \cdot \nabla + 2\mu \boldsymbol{\varepsilon} \right) (\mathbf{v} - \overline{\Pi}_h^v \mathbf{v}) - \overline{\Pi}_h^{v,*} (\mathbb{I})(p - \overline{\Pi}_h^v p) \\ (\overline{\Pi}_h^{v,*} (\nabla \cdot) \mathbf{v}_h, q) = (\overline{\Pi}_h^{v,*} (\nabla \cdot) \mathbf{v}, q) \end{cases} \quad (3.16)$$

This yields the desired relation between the implicit error and the interpolation error.

The a priori analysis was focused on the weak formulation of the flow problem. A Newton-Raphson linearization was performed and a backward Euler discretization was applied in time. These assumptions were also considered in the numerical simulation. The control of the error provided by the anisotropic mesh adaptation is driven by the interpolation error and consequently will result in the control of the discretization error.

### 3.2 TUNING WITH STABILIZED FINITE ELEMENT METHODS

When coupling stabilized finite element methods with anisotropic meshing, a better choice of the element's characteristic length  $h_K$  can be made. It will be shown that the VMS approach for flow problems and the SUPG method for heat transfer phenomena are among the best methods to couple with anisotropic finite element methods. However, a tuning of the stabilization coefficients is needed in both the convective and diffusive terms to take into account highly stretched elements with an anisotropic ratio of order  $O(1 : 1000)$ . Indeed, as it will be demonstrated in the numerical experiments, the direction of the velocity provides the element size needed in the convection terms while a bubble condensation technique gives rise to a proper coefficient for the diffusion term.

In this work we adopted the following definitions for the stabilizing parameters  $\tau_K$  and  $\tau_C$  for the resolution of fluid flows as proposed in [Tezduyar 00, Shakib 91] and in [Codina 00a] respectively:

$$\tau_K = \left( \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2 \|\mathbf{v}_h\|_K}{h_K} \right)^2 + \left( \frac{4\nu}{h_K^2} \right)^2 \right)^{-1/2}, \tau_C = \left( \left( \frac{\mu}{\rho} \right)^2 + \left( \frac{c_2 \|\mathbf{v}_h\|_K}{c_1 h_K} \right)^2 \right)^{1/2} \quad (3.17)$$

where  $\nu$  and  $\rho$  are respectively the kinematic viscosity and density of the fluid, and  $c_1$  and  $c_2$  are two constants independent from  $h_K$  [Codina 00a] and  $h_K$  is the characteristic length of element  $K$ .

We also use the parameters proposed in [Shakib 91, Codina 00a] for the stabilization

of convection-diffusion-reaction problems:

$$\tau_K = \left( \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2\|\mathbf{v}_h\|_K}{h_K} \right)^2 + 9 \left( \frac{4\kappa}{h_K^2} \right)^2 + \sigma^2 \right)^{-1/2} \quad (3.18)$$

Note that the calculation of  $h_K$  is the main subject of this chapter. This parameter is involved in the stabilization terms for each element  $K$  as shown by equations (3.17) and (3.18). It has been shown in the literature that good results can be obtained when using the minimum edge length of  $K$  [Mittal 00], or the element's diameter [Micheletti 04].

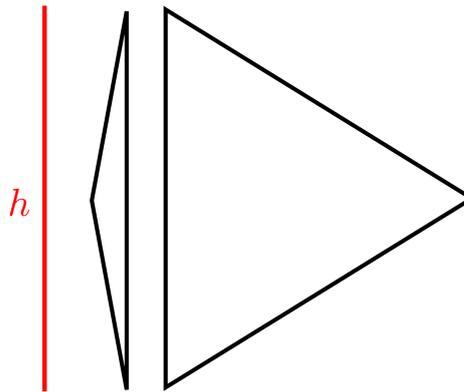


Figure 3.1 – Characteristic length for isotropic and anisotropic elements based on classical formulas.

Nevertheless, when strongly anisotropic meshes with highly stretched elements are involved, the definition of  $h_K$  is still a problem with critical interest as it plays an important role in the setting of stabilizing parameters [Harari 92, Codina 00a]. Indeed, when a classical characteristic length is used, an undesired behavior can be observed where one edge of the element  $K$  remains unchanged while the others are refined. This idea is reflected in figure 3.1. Clearly the two triangles are different, one is anisotropic while the other is isotropic, but the same characteristic length is associated to them.

One possible choice, proposed in [Micheletti 02] is to determine this factor from a relation between the direction of the convective field and the orientation of the anisotropic element  $K$  given by the eigenvalues and eigenvectors of the Riemannian metric tensor. In [Förster 09] the authors examined deeply the effect of different element length definitions on distorted meshes. In [Cangiani 05] anisotropic error estimates for the residual free bubble (RFB) method were developed to derive a new choice of the stabilizing parameters suitable for anisotropic partitions.

In this work, we modify the definition of the stabilizing parameters (3.17) to account for a modified characteristic length and improve the numerical solution's quality:

$$\tau_K = \left( \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2\|\mathbf{v}_h\|_K}{h_K^c} \right)^2 + \left( \frac{4\nu}{(h_K^d)^2} \right)^2 \right)^{-1/2}, \quad \tau_c = \left( \left( \frac{\mu}{\rho} \right)^2 + \left( \frac{c_2\|\mathbf{v}_h\|_K}{c_1 h_K^c} \right)^2 \right)^{1/2} \quad (3.19)$$

As stated in chapter 1, the stabilization parameter switches the tuning between a

diffusion dominated regime where the term  $\left(\frac{4\nu}{(h_K^d)^2}\right)$  affects the most the stabilization and a convection dominated regime where the term  $\left(\frac{2\|\mathbf{v}_h\|_K}{h_K^c}\right)$  plays the most significant role in stabilizing the numerical solution.

On the other hand, for the convection-diffusion-reaction equation we define:

$$\tau_K = \left( \left( \left( \frac{2}{\Delta t} \right)^2 + \frac{2\|\mathbf{v}_h\|_K}{h_K^c} \right)^2 + 9 \left( \frac{4\kappa}{(h_K^d)^2} \right)^2 + \sigma^2 \right)^{-1/2} \quad (3.20)$$

The stabilization parameter controls in a diffusion dominated regime where the term  $\left(\frac{4\kappa}{(h_K^d)^2}\right)$  affects the most the stabilization and a convection dominated regime where the term  $\left(\frac{2\|\mathbf{v}_h\|_K}{h_K^c}\right)$  plays the most significant role in stabilizing the numerical solution.

The question that arises at this stage is: how do we choose the characteristic lengths to allow on one hand a good stabilization and on the other hand preserve the conservation properties.

When dealing with convection dominated problems, we adopt the method proposed in [Tezduyar oo] for computing the element's characteristic size  $h_K^c$ . This approach takes into account the information on the projected flow along the gradient of the basis functions relative to the element. It consists in computing  $h_K^c$  as the diameter of  $K$  in the direction of the velocity  $\mathbf{v}$  as follows (see Figure 3.2):

$$h_K^c = \frac{2|\mathbf{v}|}{\sum_{i=1}^{N_K} |\mathbf{v} \cdot \nabla \varphi_i|} \quad (3.21)$$

where  $N_K$  is the number of vertices of element  $K$  and  $\varphi_i$  is the shape function that is non-zero on the element  $K$ . We note the dependence of this definition on the flow direction as shown in figure 3.2.

On the other hand, in the diffusion dominated regimes, the characteristic length  $h_K^d$  for the stabilization of the Navier-Stokes equations will be defined such that:

$$\frac{\nu}{(h_K^d)^2} = \frac{1}{\Delta t} + 2\nu \int_K \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K) dK, \quad (3.22)$$

and for the stabilization of the convection-diffusion-reaction equation,

$$\frac{\kappa}{(h_K^d)^2} = \frac{1}{\Delta t} + \int_K \kappa \nabla b_K \cdot \nabla b_K dK, \quad (3.23)$$

where  $b_K$  denotes the bubble shape function on element  $K$ . The proof of these choices is based on a comparison between the considered stabilization parameter and its analogous one provided by a *Mini element* formulation. When anisotropic meshing is applied, the *Mini element* method [Arnold 84] that involves a static condensation of bubble

functions at the element level is advised as it overcomes the difficulty of choosing appropriate characteristic lengths for anisotropic elements. The derivation will be given in the following section.

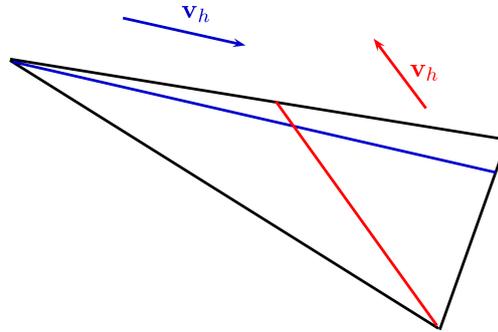


Figure 3.2 – Element's characteristic length in the streamline direction.

### 3.2.1 Illustrative example

To illustrate on the need to modify the definition of characteristic length, let us consider the following example. The numerical simulation consists of having a simple Poiseuille flow evolve inside a rectangular channel  $(0, 4) \times (0, 2)$ . The dynamic viscosity of the fluid is set to  $\mu = 10^{-2} Pa.s$  and the density to  $\rho = 1 kg/m^3$ . Parabolic Poiseuille flow inlet velocity is assigned on the vertical walls with a maximum of  $1.5 m/s$ . No slip boundary conditions are imposed on the horizontal walls. For this example we have applied the anisotropic mesh adaptation on the velocity field. The adapted mesh is presented in figure 3.3. We note how highly stretched and directional the elements are, hence allowing a good capture of the emerging boundary layers. The velocity profiles and 1D vertical cuts at the inlet and outlet boundaries are shown in figure 3.4. We distinguish the velocity magnitude distribution obtained with the classical characteristic element length on the left and the one obtained with the modified element length on the right of the figure. We observe that the velocity is zero at one wall, increases at the center to a maximum then decreases back to zero at the opposite wall. Comparing the profile of the velocity distribution at the inlet with the one at the outlet boundary we can clearly identify a discharge in the case where the classical characteristic length is used yielding a violation of the mass conservation property of the flow. The latter property is conserved when the modified mesh size is used. This validates the importance of the proposed stabilization parameters.



Figure 3.3 – Anisotropic mesh obtained for the Poiseuille problem.

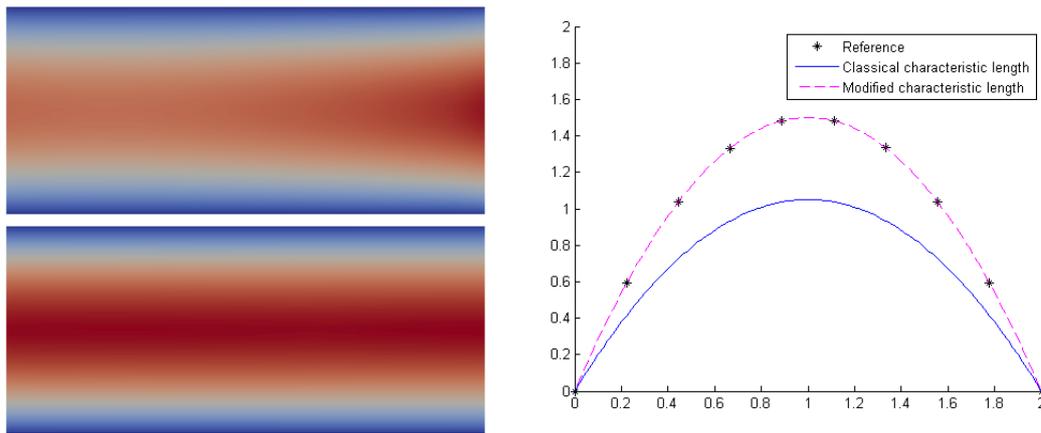


Figure 3.4 – Velocity profiles obtained with the classical characteristic length (top-left) and the modified one (bottom-left) and 1D vertical cut along the outlet boundary (right).

### 3.2.2 Validation of the characteristic lengths choice

### 3.2.3 Navier-Stokes equations with dominant viscous terms

When *Mini element/bubble condensation* formulation is used, the mass conservation property is satisfied on both isotropic and anisotropic meshes [Micheletti 02] whereas as shown in the Poiseuille flow numerical example, a velocity discharge occurs when the classical stabilization parameters are employed. To remedy this problem, we resort to identifying a new representation of the characteristic length through an analogy analysis with the *mini element/bubble condensation* (a  $P_{1+}/P_1$ ) method.

The method consists in enriching the finite element space with a space of bubble functions whose support remains inside the mesh elements. The problem will then be uncoupled into two-scales: the coarse and the fine scale. While the former is defined on the whole computational domain, the latter is defined on the elements' interior with zero Dirichlet boundary conditions.

Following the lines of [Hachem 09] for the derivation of the stable *mini element* method on the Stokes problem, we are interested, in this section, in extending this technique to a mixed discrete formulation for the transient incompressible Navier-Stokes equations given by: find the velocity  $\mathbf{u}(x, t)$  and the pressure  $p(x, t)$  such that:

$$\begin{aligned} (\rho \partial_t \mathbf{v} + \mathbf{v} \cdot \nabla \mathbf{v}) - 2 \nabla \cdot \mu \boldsymbol{\varepsilon}(\mathbf{v}) + \nabla p &= \mathbf{f} \quad \text{in } \Omega \times (0, T) \\ \nabla \cdot \mathbf{v} &= 0 \quad \text{in } \Omega \times (0, T) \end{aligned} \quad (3.24)$$

where  $\sigma$  is the stress tensor,  $\rho$  the density of the fluid,  $\mathbf{f}$  a given source term and  $\mu$  the dynamic viscosity.

**Classical mixed formulation:** We proceed as described in chapter 1 to derive the weak formulation of this problem. We discretize the domain  $\Omega$  using a finite element mesh  $\mathcal{H}_h$  and we define the discrete functional spaces for the velocity  $\mathcal{V}_h$ , weighting



**Remark 32.** *As the fine-scale space is assumed orthogonal to the finite element one, the crossed viscous terms vanish [Coupez 96].*

The fine-scale part is usually modeled via residual based terms that are derived consistently on the elements' interior. The modeling of the fine-scale velocity is followed by a static condensation that consists in substituting the effect of the fine-scale solution into the large-scale problem (3.29), thereby eliminating the explicit appearance of the fine-scale while still modeling their contribution to the coarse scale solution. Consequently, additional stabilized terms, tuned by a local time-dependent stabilizing parameter, will enhance the stability and accuracy of the standard Galerkin formulation for the transient nonlinear Navier-Stokes equations.

At this stage the objective is to determine the small scale velocity and retain its effects on the large scale problem.

Rearranging the terms, equation (3.30) is equivalent to:

$$\left\{ \begin{array}{l} \underbrace{\rho \left( \frac{\partial \tilde{\mathbf{v}}_h}{\partial t}, \tilde{\boldsymbol{\omega}} \right)}_{\text{time dependent subscale}} + \underbrace{\rho \left( (\mathbf{v}_h + \tilde{\mathbf{v}}_h) \cdot \nabla \tilde{\mathbf{v}}_h, \tilde{\boldsymbol{\omega}} \right)}_{\text{non-linear convection term}} + (2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{v}}_h) : \boldsymbol{\varepsilon}(\tilde{\boldsymbol{\omega}})) \\ = \left( \mathbf{f} - \rho \frac{\partial \mathbf{v}_h}{\partial t} - \rho (\mathbf{v}_h + \tilde{\mathbf{v}}) \cdot \nabla \mathbf{v}_h - \nabla p_h, \tilde{\boldsymbol{\omega}} \right) \quad \forall \tilde{\boldsymbol{\omega}} \in \tilde{\mathcal{V}}_{h,0} \\ = (\mathcal{R}_m, \tilde{\boldsymbol{\omega}}) \quad \forall \tilde{\boldsymbol{\omega}} \in \tilde{\mathcal{V}}_{h,0} \end{array} \right. \quad (3.31)$$

One can clearly notice the high nonlinearity and the time-dependence of these equations. Inspired by the work of [Codina ooa] to deal with the nonlinear convective term, we resort to applying a Newton Raphson linearization. Although this method requires the solution of the Newton equation to be computed at every iteration, thus increasing the computational cost, it is favorable as it features rapid convergence and good initial guess.

Again we follow the lines of [Hachem 09] for treating the nonlinear terms:

- We approximate the nonlinear convective term in the large-scale problem at the  $i^{\text{th}}$  Newton Raphson iteration up to first order as in equation (3.13).
- We approximate the nonlinear convective term in the fine-scale problem by its large-scale part:

$$(\mathbf{v}_h + \tilde{\mathbf{v}}) \cdot \nabla (\mathbf{v}_h + \tilde{\mathbf{v}}) \approx \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \tilde{\mathbf{v}} \cdot \nabla \tilde{\mathbf{v}} \quad (3.32)$$

Without loss of generality, we use an implicit time marching scheme, the generalization to other time discretizations is straightforward.

Taking into account the above assumption on the nonlinear term (eq. (3.32)), the

fine-scale momentum equation (3.31) becomes:

$$\rho \left( \frac{\tilde{\mathbf{v}}}{\Delta t}, \tilde{\boldsymbol{\omega}} \right) + \rho \left( \mathbf{v}_h^{i-1} \cdot \nabla \tilde{\mathbf{v}}, \tilde{\boldsymbol{\omega}} \right) + (2\mu \boldsymbol{\varepsilon}(\tilde{\mathbf{v}}_h) : \boldsymbol{\varepsilon}(\tilde{\boldsymbol{\omega}}_h)) = (\mathcal{R}_m, \tilde{\boldsymbol{\omega}}) \quad \forall \tilde{\boldsymbol{\omega}} \in \tilde{\mathcal{V}}_{h,0} \quad (3.33)$$

Note that the term  $\frac{\tilde{\mathbf{v}}^-}{\Delta t}$  from the implicit time discretization goes into the formula of the residual  $\mathcal{R}_m$  on the right hand side of the equation. Inhere we mean by  $(\tilde{\mathbf{v}}^-)$  the solution at the previous time step of the problem resolution.

The last equation shows that the fine-scale velocity field is closely related to the solution on the large-scale.

Using the fine-scale fields expansion proposed by [Masud 04], we have:

$$\tilde{\mathbf{v}}_h = \sum_{K \in \mathcal{H}_h} \mathbf{v}'_K b_K \quad \text{and} \quad \tilde{\boldsymbol{\omega}}_h = \sum_{K \in \mathcal{H}_h} \boldsymbol{\omega}'_K b_K \quad (3.34)$$

with  $b_K$  being the bubble shape functions,  $\mathbf{v}'_K$  the coefficients for the fine-scale velocity field and  $\boldsymbol{\omega}'_K$  the coefficient for the fine-scale weighting function at the level of the element  $K$ 's bubble.

Replacing  $\tilde{\mathbf{v}}_h$  and  $\tilde{\boldsymbol{\omega}}_h$  by their expressions in equation (3.33) implies:

$$\begin{aligned} \rho \left( \frac{b_K \mathbf{v}'_K}{\Delta t}, b_K \boldsymbol{\omega}'_K \right) + \rho \left( \mathbf{v}_h^{i-1} \cdot \nabla b_K u'_K, b_K \boldsymbol{\omega}'_K \right)_K \\ + (2\mu \boldsymbol{\varepsilon}(b_K \mathbf{v}'_K) : \boldsymbol{\varepsilon}(b_K \boldsymbol{\omega}'_K))_K = (\mathcal{R}_m, b_K \boldsymbol{\omega}'_K) \quad \forall K \in \mathcal{H}_h, \forall \boldsymbol{\omega}'_K \end{aligned} \quad (3.35)$$

**Remark 33.** *The integrals over the inter-element boundaries will be neglected as the bubble functions vanish on the elements' boundaries. Consequently the equation can be decoupled at the elements' level.*

Equation (3.33) is valid for any choice of  $\boldsymbol{\omega}'_K$  in particular for  $\boldsymbol{\omega}'_K = 1$  at the center of element  $K$  and  $\boldsymbol{\omega}'_K = 0$  on its boundary.

On the other hand, we apply the following approximation adopted from [Basset 06]:

$$\rho \left( \frac{b_K \mathbf{v}'_K}{\Delta t}, b_K \boldsymbol{\omega}'_K \right) = \frac{\rho}{\Delta t} \quad (3.36)$$

Therefore the fine-scale velocity derived from equation (3.35) reads as:

$$\mathbf{v}'_K = \frac{1}{\left[ \frac{\rho}{\Delta t} + \underbrace{\rho \left( \mathbf{v}_h^{i-1} \cdot \nabla b_K, b_K \right)_K}_{\text{convective term}} + \underbrace{(2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K}_{\text{viscous term}} \right]} (\mathcal{R}_m, b_K) \quad \forall K \in \mathcal{H}_h \quad (3.37)$$

The stabilizing parameter  $\tau_K^d$  naturally appeared from the solution of the small-scale problem. Assuming constant large scale residual  $\mathcal{R}_m$  on linear elements, it follows

that:

$$\tau_K^d = \frac{1}{\left[ \frac{\rho}{\Delta t} + \underbrace{\rho \left( \mathbf{v}_h^{i-1} \cdot \nabla b_K, b_K \right)_K}_{\text{convective term}} + \underbrace{(2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K}_{\text{viscous term}} \right]} \quad \forall K \in \mathcal{H}_h \quad (3.38)$$

**Remark 34.** *The stabilizing parameter depends on the choice of the bubble function. However as stated in [Hachem 09], when the same bubble functions are used for both the velocity and the weighting function, the convection term cancels out from the formula. Indeed under the assumption that  $\mathbf{v}_h^{i-1}$  is piecewise constant over the bubble, the choice of the Mini element yields:*

$$\left( \mathbf{v}_h^{i-1} \cdot \nabla b_K, b_K \right)_K = 0 \quad \forall K \in \mathcal{H}_h \quad (3.39)$$

Consequently, we have that:

$$\tau_K^d = \frac{1}{\left[ \frac{\rho}{\Delta t} + (2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K \right]} \quad \forall K \in \mathcal{H}_h \quad (3.40)$$

At this stage instead of solving for the small-scale velocity field at the element level, we apply a static condensation whereby one substitutes the fine scale solution, expressed in terms of large-scale momentum residual, into the large-scale problem. Upon this step, additional terms arise in the discrete problem and are tuned by a local time-dependent stabilizing parameter. These terms enhance the stability and accuracy of the standard Galerkin formulation for the transient non-linear Navier-Stokes equations.

Going back to the coarse-scale problem and taking into account the assumptions made on the nonlinear term, we obtain:

$$\begin{cases} \left( \rho \frac{\partial \mathbf{v}_h + \tilde{\mathbf{v}}_h}{\partial t}, \boldsymbol{\omega}_h \right) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \boldsymbol{\omega}_h) + \left( \rho \mathbf{v}_h^{i-1} \cdot \nabla \tilde{\mathbf{v}}_h, \boldsymbol{\omega}_h \right) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\boldsymbol{\omega}_h)) \\ - (p_h, \nabla \cdot \boldsymbol{\omega}_h) = (\mathbf{f}, \boldsymbol{\omega}_h) \quad \forall \boldsymbol{\omega}_h \in \mathcal{V}_{h,0} \\ (\nabla \cdot \mathbf{v}_h, q_h) + (\nabla \cdot \tilde{\mathbf{v}}_h, q_h) = 0 \quad \forall q_h \in \mathcal{Q}_h \end{cases} \quad (3.41)$$

Integrating by parts the third term of the first equation and the second term of the second equation then:

$$\begin{cases} \left( \rho \frac{\partial \mathbf{v}_h + \tilde{\mathbf{v}}_h}{\partial t}, \boldsymbol{\omega}_h \right) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \boldsymbol{\omega}_h) - \sum_{K \in \mathcal{H}_h} \left( \tau_K^d \mathcal{R}_m, \rho \mathbf{v}_h^{i-1} \cdot \nabla \boldsymbol{\omega}_h \right) + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\boldsymbol{\omega}_h)) \\ - (p_h, \nabla \cdot \boldsymbol{\omega}_h) = (\mathbf{f}, \boldsymbol{\omega}_h) \quad \forall \boldsymbol{\omega}_h \in \mathcal{V}_{h,0} \\ (\nabla \cdot \mathbf{v}_h, q_h) - \sum_{K \in \mathcal{H}_h} \left( \tau_K^d \mathcal{R}_m, \nabla q_h \right) = 0 \quad \forall q_h \in \mathcal{Q}_h \end{cases} \quad (3.42)$$

**Remark 35.** *Note that the term  $(\partial_t \tilde{\mathbf{v}}, \boldsymbol{\omega})$  vanishes as we consider static subscales.*

Substituting the fine scale velocity by its expression yields:

$$\left\{ \begin{array}{l} \left( \rho \frac{\partial \mathbf{v}_h}{\partial t}, \boldsymbol{\omega}_h \right) + (\rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h, \boldsymbol{\omega}_h) - \sum_{K \in \mathcal{H}_h} \left( \tau_K^d \left( \frac{\partial \mathbf{v}_h}{\partial t} + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \rho \frac{\bar{\mathbf{v}}_h}{\Delta t} - \mathbf{f} \right), \rho \mathbf{v}_h^{i-1} \cdot \nabla \boldsymbol{\omega}_h \right) \\ \quad + (2\mu \boldsymbol{\varepsilon}(\mathbf{v}_h) : \boldsymbol{\varepsilon}(\boldsymbol{\omega}_h)) - (p_h, \nabla \cdot \boldsymbol{\omega}_h) = (\mathbf{f}, \boldsymbol{\omega}_h) \quad \forall \boldsymbol{\omega}_h \in \mathcal{V}_{h,0} \\ \\ (\nabla \cdot \mathbf{v}_h, q_h) - \sum_{K \in \mathcal{H}_h} \left( \tau_K^d \left( \frac{\partial \mathbf{v}_h}{\partial t} + \rho \mathbf{v}_h \cdot \nabla \mathbf{v}_h + \nabla p_h - \rho \frac{\bar{\mathbf{v}}_h}{\Delta t} - \mathbf{f} \right), \nabla q_h \right) = 0 \quad \forall q_h \in \mathcal{Q}_h \end{array} \right. \quad (3.43)$$

This completes the mixed finite element formulation analysis. We can obviously detect the additional stabilizing terms involving integrals over the elements' interiors. Consequently a stable solution is obtained in the sense that it satisfies the inf-sup condition.

Recall that the main objective of this analysis is to compute the appropriate elements' characteristic lengths in case of diffusion dominated regimes and anisotropic mesh adaptation.

An equivalence between the residual free bubble approach and the Mini-element method was highlighted in the literature [Canuto 96, Russo 96]. Indeed the latter provides a natural way of determining the stabilizing terms. We note that the choice of the bubble function used influences directly the stabilizing parameter.

It was pointed out in [Russo 96, Canuto 96, Franca 02] that when the viscous term is large with respect to the convection term, the stabilizing parameter derived by a residual free bubble method is given by:

$$\tau_K^d = \frac{\int_K b_K dK}{|K|} \quad (3.44)$$

where  $b_K$  is the bubble function solution of the boundary value problem:

$$\left\{ \begin{array}{ll} -\mu \Delta b_K = 1 & \text{in } K \\ b_K = 0 & \text{on } \partial K \end{array} \right. \quad (3.45)$$

The authors of [Russo 96, Canuto 96, Franca 02] also demonstrated that the stabilization term in the diffusion dominated regime can be approximated by:

$$\tau_K^d \approx \frac{(h_K^d)^2}{\mu} \quad (3.46)$$

where  $(h_K^d)$  is the characteristic length of the element  $K$ .

Proceeding by analogy between the two forms of stabilization parameters defined by equations (3.40) and (3.46), we infer that:

$$\frac{(h_K^d)^2}{\mu} = \frac{1}{\left[ \frac{\rho}{\Delta t} + (2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K \right]} \quad (3.47)$$

and equivalently,

$$\frac{\mu}{(h_K^d)^2} = \frac{\rho}{\Delta t} + (2\mu \boldsymbol{\varepsilon}(b_K) : \boldsymbol{\varepsilon}(b_K))_K \quad (3.48)$$

This completes the proof of the choice of the characteristic length in case of the Navier-Stokes equations with high viscous terms.

### 3.2.4 Convection-Diffusion-Reaction equation with dominant viscous terms

We consider the convection-diffusion-reaction equation defined on the domain  $\Omega \times (0, T)$  by:

find a function  $u(x, t)$  such that:

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \nabla u - \nabla \cdot (\kappa \nabla u) + \sigma u = f \quad (3.49)$$

where  $\mathbf{v}$  is a given divergence-free velocity field,  $\kappa > 0$  is the diffusion coefficient,  $\sigma \geq 0$  the reaction coefficient, and  $f$  a source term function. For the sake of simplicity we assume a zero boundary condition on  $\partial\Omega$ . The generalization to other types of boundary conditions is straightforward. We discretize the domain with a mesh  $\mathcal{H}_h$  and define the functional space  $\mathcal{U}$  in which we are seeking the solution of the problem:

$$\mathcal{U} := H_0^1(\Omega) = \left\{ v \in H^1(\Omega) \mid v = 0 \forall x \in \partial\Omega \right\}. \quad (3.50)$$

**Standard formulation:** We enrich the solution's functional space by a discrete space corresponding to the bubble functions:

$$\widetilde{\mathcal{U}}_h = \left\{ \tilde{\mathbf{u}}_h, \tilde{\mathbf{u}}_{h|K_i} \in P^1(K_i) \cap H_0^1(K_i), \forall K \in \mathcal{H}_h, i = 1, \dots, D \right\} \quad (3.51)$$

We apply the same analysis and the same arguments as above to derive the stabilization parameters:

- As the small scale functional space is independent from the coarse scale one, the problem can be decoupled into two sub-problems.
- Using an Euler implicit discretization in time.
- Rearranging the terms in the fine-scale equation to express the small scale terms as a residual of the large scale problem.

The fine-scale problem becomes:

$$\left\{ \begin{array}{l} \left( \frac{\tilde{u}}{\Delta t} + v \nabla \tilde{u}, \tilde{\psi} \right) + (\kappa \nabla \tilde{u}, \nabla \tilde{\psi}) = \left( f - \frac{\partial u}{\partial t} - \sigma u - \mathbf{v} \cdot \nabla u + \nabla \cdot (\kappa \nabla u) - \frac{\tilde{u}^-}{\Delta t}, \tilde{\psi} \right) \\ \hspace{15em} = (\mathcal{R}_{\text{CDR}}, \tilde{\psi}) \end{array} \right. \quad (3.52)$$

We employ the fine-scale field expansion proposed by [Masud 04]:

$$\tilde{u}_h = \sum_{K \in \mathcal{H}_h} u'_K b_K \quad \text{and} \quad \tilde{\psi}_h = \sum_{K \in \mathcal{H}_h} \psi'_K b_K \quad (3.53)$$

with  $b_K$  being the bubble shape functions,  $\mathbf{u}'_K$  the coefficients for the fine-scale solution field and  $\psi'_K$  the coefficient for the fine-scale weighting function at the level of the element  $K$ 's bubble.

Substituting  $\tilde{u}_h$  and  $\tilde{\psi}_h$  by their expressions in equation (3.52) and using the property that the bubble functions vanish at the elements' boundaries, we can split the equation into sub-equations defined at the elements' level:

$$\left( \frac{u'_K b_K}{\Delta t} + v \nabla u'_K b_K, \psi'_K b_K \right) + (\kappa \nabla u'_K b_K, \nabla \psi'_K b_K) = (\mathcal{R}_{\text{CDR}}, \psi'_K b_K) \quad (3.54)$$

Then we apply the following approximation adopted from [Basset 06]:

$$\left( \frac{u'_K b_K}{\Delta t}, \psi'_K b_K \right) = \frac{1}{\Delta t} \quad (3.55)$$

Hence the fine-scale problem becomes:

$$\frac{u'_K}{\Delta t} + (\mathbf{v} \nabla u'_K b_K, \psi'_K b_K)_K + (\kappa \nabla u'_K b_K, \nabla \psi'_K b_K)_K = (\mathcal{R}_{\text{CDR}}, \psi'_K b_K)_K \quad \forall K \in \mathcal{H}_h \quad (3.56)$$

As the equation is satisfied for any choice of  $\psi'_K$  in particular for:

$$\psi'_K = \begin{cases} 1 & \text{on the element's center,} \\ 0 & \text{elsewhere.} \end{cases} \quad (3.57)$$

It follows that the small-scale solution will be given by:

$$u'_K = \frac{1}{\frac{1}{\Delta t} + (\mathbf{v} \nabla b_K, b_K)_K + (\kappa \nabla b_K, \nabla b_K)_K} (\mathcal{R}_{\text{CDR}}, b_K)_K \quad (3.58)$$

**Remark 36.** Using the same bubble function for the solution and the weighting function leads to the cancellation of the convection term.

The stabilizing parameter  $\tau_K^d$  naturally appeared from the solution of the small-scale problem. Assuming constant large scale residual  $\mathcal{R}_{\text{CDR}}$  on linear elements, it follows that:

$$\tau_K^d = \frac{1}{\frac{1}{\Delta t} + (\kappa \nabla b_K, \nabla b_K)_K} \quad (3.59)$$

At the diffusion limit, it was demonstrated in [Russo 96, Franca 98, Franca 92a] that the stabilization parameter derived using a residual free bubble approach can be approximated by:

$$\tau_K^d \approx C \frac{(h_K^d)^2}{\kappa} \quad (3.60)$$

where  $C$  is a constant and  $(h_K^d)$  is the characteristic length of the element  $K$ .

Consequently, proceeding by analogy between equations (3.60) and (3.59), we can infer

that:

$$\frac{\kappa}{C(h_K^d)^2} = \frac{1}{\Delta t} + (\kappa \nabla b_K, \nabla b_K)_K. \quad (3.61)$$

This concludes the derivation of the characteristic length choice in the case of diffusion dominated regime for the convection-diffusion-reaction problem.

**Remark 37.** *At the convection dominated limit, it was stated in [Franca 02] that using the stabilization parameters derived from the Residual free bubble method can lead to a loss of stability. To avoid this, we treat the convection dominated regime with a different choice of characteristic length:*

$$h_K = \frac{2|\mathbf{v}|}{\sum_{i=1}^{N_K} |\mathbf{v} \cdot \nabla \varphi_i|}.$$

There is no need for additional modifications to be performed in order to couple the stabilized finite element with an anisotropic mesh adaptation. As the latter is designed to fit in the general context, it does not depend on the treated problem nor on the chosen field of adaptation. We will show in chapter 5 how the metric construction will be extended to account for several fields in one shot. For the time being, we intend to apply the method on one field of interest which is considered as the motor of the simulation.

### 3.3 APPLICATIONS TO CONVECTION-DIFFUSION PROBLEMS

The numerical solution of the unsteady convection-diffusion equation using the Galerkin formulation normally exhibits global spurious oscillations in convection-dominated problems, especially in the vicinity of sharp gradients, and steep boundary and interior layers. Over the last two decades, investigations on the development of numerical methods dealing with such problems were very active. In the opening of his book, Morton [Morton 96] stated that: ‘*Accurate modelling of the interaction between convective and diffusive processes is the most ubiquitous and challenging task in the numerical approximation of partial differential equations*’. However, in the presence of steep boundary or interior layers, if an isotropic mesh is to be used, then a very small mesh size shall be considered everywhere in the domain in order to achieve a good level of precision. Hence, a high computational cost is expected. Mesh adaptation is a key tool for lowering the computational cost while maintaining a good level of accuracy. The idea is to concentrate the elements of the mesh in the domain region where the solution exhibits fast variations. Note that for boundary and interior layers, the solution varies faster in one direction than it does in the others, therefore a mesh mimicking this behavior by employing small mesh sizes in the direction orthogonal to the gradient and larger ones in the parallel direction is highly desirable. Consequently, anisotropic meshes are efficient tools as they render a higher accuracy for the same mesh complexity when compared to isotropic refinement.

On the other hand, as discussed in chapter 1 when a stabilized numerical scheme

is used, on an isotropic mesh, the computed layer might be smeared and the convergence rate is lowered. The choice of stabilizing parameters is still a puzzling question since different choices introduce different amounts of diffusion. It was pointed out in the literature that optimal convergence can be recovered using adapted meshes while reducing the excessive smearing. From this standpoint, anisotropic mesh adaptation can be considered as a keystone for recovering the optimal rate of convergence.

#### *Historical overview*

Research on the development of adaptive meshes together with numerical methods for the resolution of convection-diffusion problems date back to the year 1969 with the work of Bakhvalov [Linβ 85] who proposed the use of layer-adapted graded meshes. The latter are fine in the vicinity of boundary layers and coarse outside of it. Instead of using graded meshes, Shishkin [Shishkin 92, Shishkin 04] proposed to employ a piecewise uniform mesh in the 1990s. This idea led to fruitful contributions during that period [Linβ 99, Farrell 00, Xenophontos 03, Farrell 04, Reilly 05]. The Shishkin meshes are set a priori with fine resolution near the layers and coarse otherwise. They permit a good level of accuracy and the generation of a less diffusive solution. However, a major drawback of Shishkin meshes is that they should be set a priori that is, one must know the location and the nature of the layer in the pre-simulation phase to ensure its good performance. When simulating unsteady convection-diffusion problems, the solution gradients' locations and the direction in which they will evolve are not known a priori. This drawback highlights the need for a method that automatically follows and captures the emerging solution gradient. Anisotropic mesh adaptation is the best candidate for such applications. Very few research has been done on anisotropic mesh adaptation algorithm applied in the context of convection-diffusion problems [Nguyen 09, Micheletti 04, Picasso 09].

The method that we propose is capable of automatically detecting emerging layers in the solution. It is robust and efficient in resolving them. As it will be shown in the numerical validation, the method allows an improvement of the computed solution's accuracy in particular in the wake of boundary and internal layers. It is designed to work without modification on any kind of applications. It does not need an a priori knowledge on the layers form, position or evolution. It works under the constraint of a fixed number of nodes to produce an optimal mesh. From a practical point of view, this is an important advantage especially when dealing with industrial applications. Indeed, in order to perform long-time and large-scale industrial applications it is preferable to choose a number of nodes  $N$  based on the available computational resources and determine the optimal mesh yielding the most accurate solution.

### **3.3.1 Numerical experiments on the steady convection-diffusion problem**

In this section, we assess the performance of the newly developed adaptive algorithm on several steady convection diffusion problems exhibiting boundary layers. We start

with an evaluation of the order of convergence of the numerical method. Then we show the ability of the stabilization method associated with the anisotropic mesh adaptation technique to automatically detect and resolve the emerging layers without the appearance of non-physical oscillations. The obtained solutions are compared to a set of references presented in [Nguyen 09, Zhang 03, Huang 05, Farrell 00]. The results that we present hereafter were recently published in [Hachem 14].

As it will be shown in the numerical results, the developed algorithm based on the stabilized finite element method is capable of producing accurate results within a very reasonable execution time and a low computational cost.

### Example 1: Regular boundary layers

We consider a first test case with a continuous solution and regular boundary layers. This test has been studied by several authors [Nguyen 09, Zhang 03].

The computation domain is set to  $\Omega = (0, 1)^2$ , the velocity field to  $\mathbf{v}(x, y) = (1, 1)^T$  and the diffusion coefficient is varied  $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$ . The exact solution of the problem is given by:

$$u(x, y) = xy \left( 1 - \exp^{-\frac{1-x}{\kappa}} \right) \left( 1 - \exp^{-\frac{1-y}{\kappa}} \right). \quad (3.62)$$

It develops boundary layers at  $x = 1$  and  $y = 1$ . The boundary conditions and the source term are determined from the exact solution. When the diffusion coefficient tends to 0, the flow becomes convection dominated and thus the standard Galerkin approach is polluted by spurious oscillations. The latter are avoided and a smooth solution is obtained when applying the SUPG stabilization with anisotropic mesh adaptation. Recall that the amount of added artificial diffusion is related to the mesh size inside the layer region. This is computed as the largest edge of the element in the direction parallel to the velocity field. We can observe in figure 3.5 that as the diffusion coefficient  $\kappa$  tends to zero, the numerical solution becomes steeper without the appearance of any numerical oscillation. Figure 3.6 shows the anisotropic meshes made up of 5,000 nodes obtained for the different values of the diffusion coefficient. We note the concentration of the nodes along the boundary layers. This reflects how, for a controlled number of nodes, the mesh is naturally and automatically coarsened in smooth regions while extremely refined near the boundary. The zoom on the right side of the cavity illustrates the sharp capture of the boundary layers and the right orientation and deformation of the mesh elements (longest edges parallel to the boundary). The limited resolution and optimal distribution of the elements yield a good level of accuracy with a reduced computational cost. This example aims at emphasizing the spatial order of convergence when using the proposed mesh adaptation technique. As noted in chapter 1, around half an order of convergence is lost when a stabilized finite element method is used on an isotropic mesh.

The global convergence order is computed in the  $L^1$ ,  $L^2$  and  $H^1$  norms. In each case, the error has been computed with respect to the reference solution. As can be seen in

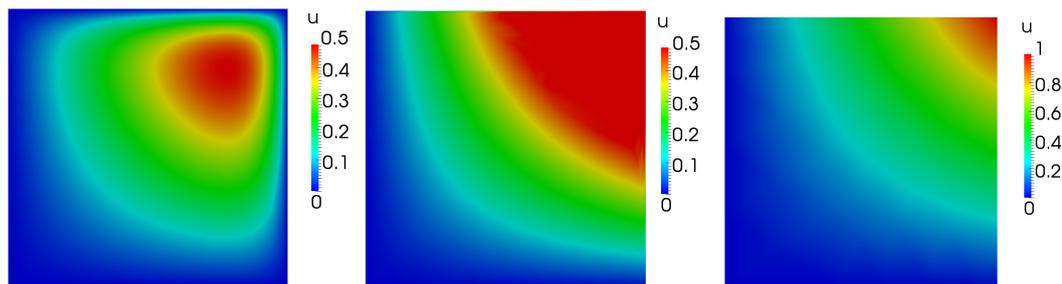


Figure 3.5 – Numerical solution for  $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$  (from left to right).

figure 3.7, the anisotropic mesh adaptation proves to be very efficient in recovering the method's order of convergence.

Comparing with the results obtained in [Nguyen 09], the anisotropic mesh adaptation technique asymptotically recovers the same order of convergence with lower interpolation error evaluated in the  $L_1$ ,  $L_2$  and  $H_1$  norms. The generated elements present in the convection dominated case, higher aspect ratios in the vicinity of the boundary layers. The elements are well oriented and elongated to efficiently provide a good level of accuracy.

#### Example 2: Interior layer with variable convective field

The second numerical experiment is taken from [Nguyen 09, Huang 05]. The aim behind this example is to test the ability of the developed numerical scheme together with the mesh adaptation algorithm to capture interior layers. We solve the steady convection-diffusion equation with the diffusion coefficient  $\kappa = 0.005$  on  $\Omega = (0, 1)^2$ . We choose the exact solution to be continuous all over the domain but to develop an interior layer along the line  $y = -x + 0.85$ :

$$u(x, y) = \left(1 + \exp \frac{x+y-0.85}{2\kappa}\right)^{-1}$$

The boundary conditions and the source term are determined from the exact solution. We consider a variable velocity field  $\mathbf{v} = (u(x, y), u(x, y))^T$ . Figure 3.8 (left) presents the numerical solution obtained on a mesh made up of around 2,000 nodes. The anisotropic mesh that is automatically generated by our developed adaptive algorithm is shown in figure 3.9. One can notice the high concentration of elements along the line  $y = -x + 0.85$  in order to accurately capture the produced inner layer. Taking a closer look at the mesh in the zoomed version near the left boundary we can detect the good orientation of the elements with the stretching in the direction of the sharp layer. This demonstrates the ability of the algorithm to work under the constraint of a fixed number of nodes and to effectively control the element sizes, orientations and locations.

Again the convergence is of first order in the  $H^1$  norm and second order in the  $L_1$  and  $L_2$  norms recovering the theoretical predictions as can be seen in figure 3.8 (right). The very small errors in the  $L_1$  and  $L_2$  norms reflect the advantage of the adapted mesh as it is well aligned with the solution's strong anisotropic features.

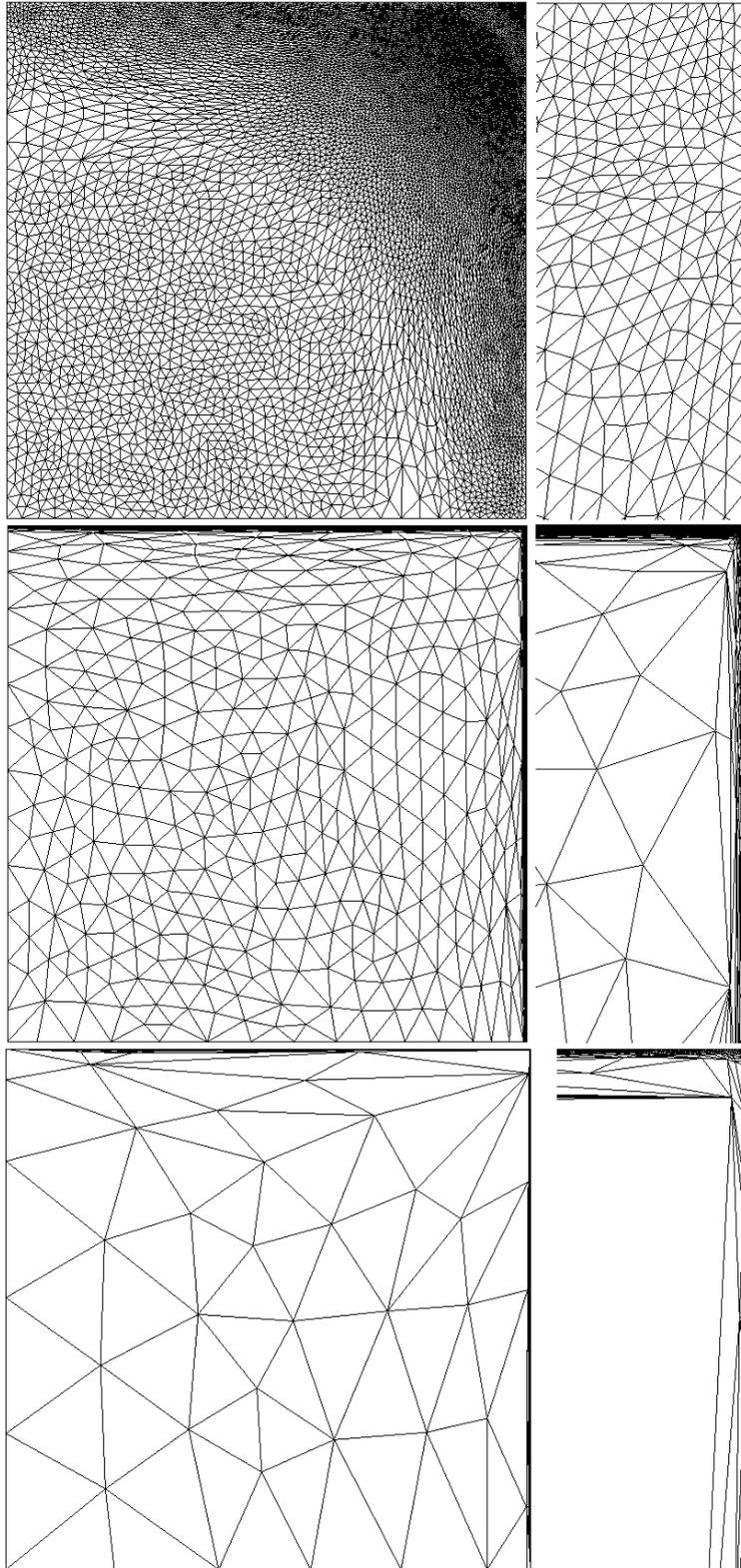


Figure 3.6 – Anisotropic meshes for  $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$  (top to bottom) and zooms on the boundary layer.

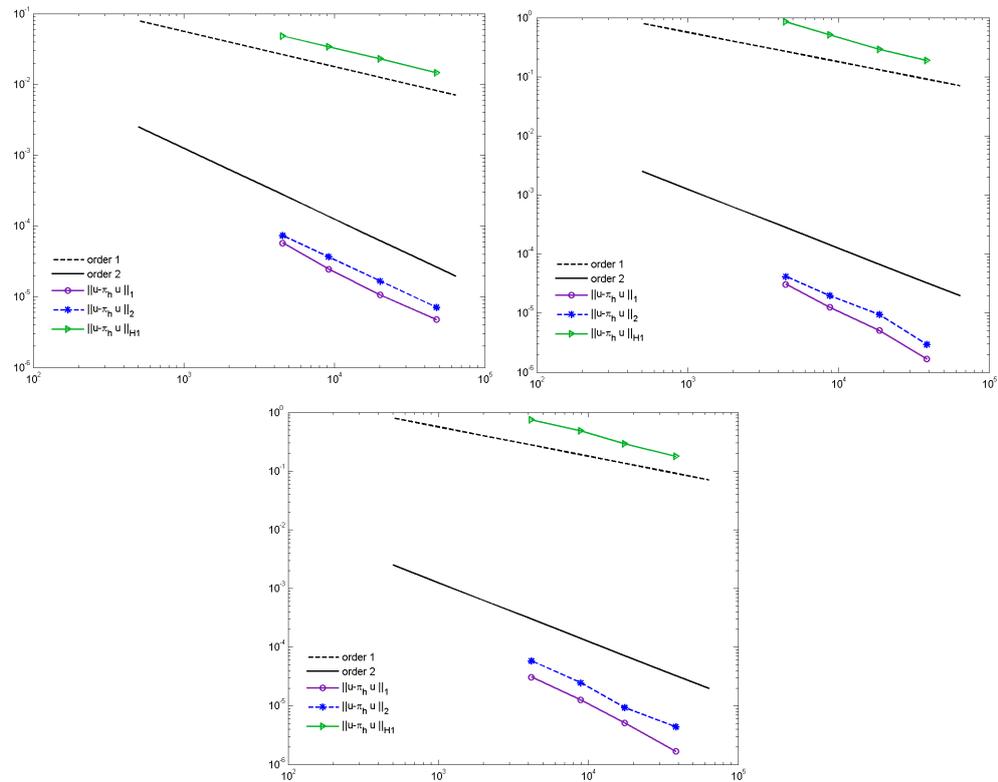


Figure 3.7 –  $L^\infty$ ,  $L^2$  and  $H^1$  norms of the error versus the number of elements in the mesh for  $\kappa = \{10^{-1}, 10^{-3}, 10^{-6}\}$  (top-left, top-right, and bottom).

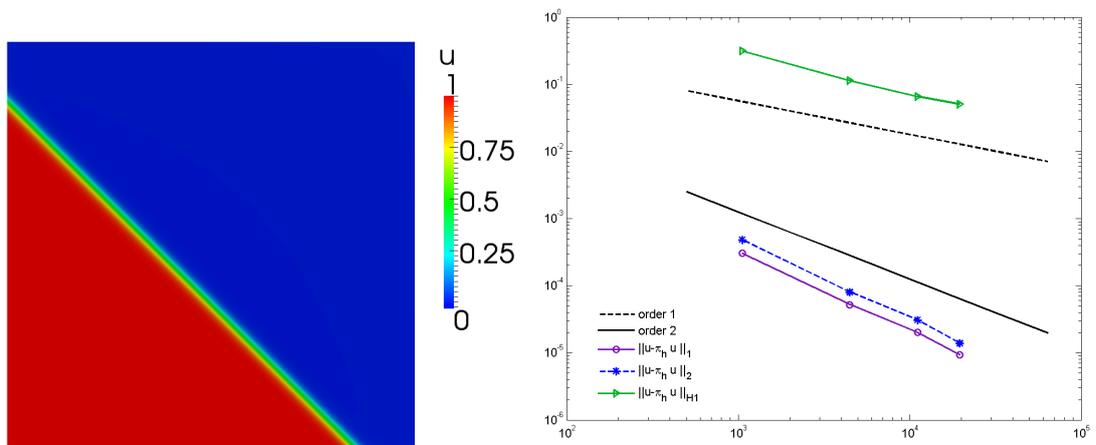


Figure 3.8 – Numerical solution (left) and convergence history in the  $L^\infty$ ,  $L^2$  and  $H^1$  (right) for  $\kappa = 0.005$  and a zoom on the interior layer.

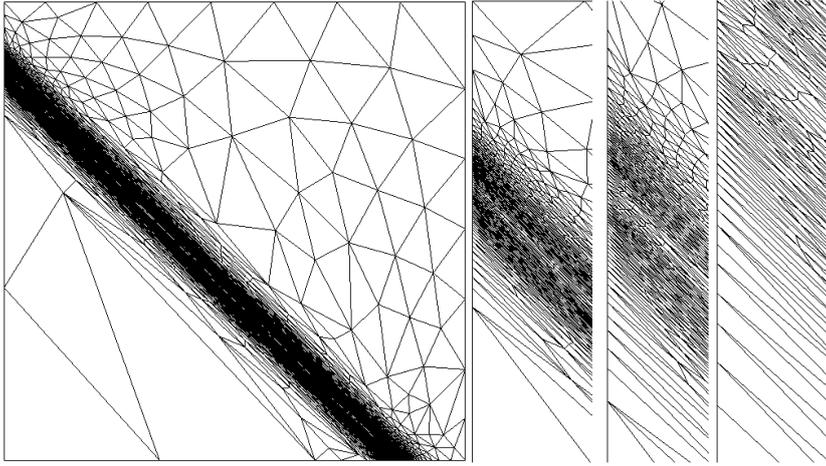


Figure 3.9 – Anisotropic mesh obtained for  $\kappa = 0.005$ .

So far we have considered two test cases where the exact solution is known and we have computed the order of convergence of the developed SUPG formulation combined with the anisotropic mesh adaptation technique. We present in what follows more complex situations inspired from [Nguyen 09, Farrell 00].

### Example 3: Regular boundary layers with smooth data

We consider the constant convective field  $\mathbf{v}(x, y) = -(2, 1)^T$  and the diffusion coefficient  $\kappa = 10^{-6}$  over the domain  $\Omega = (0, 1)^2$ . We define the source term as  $f(x, y) = -x^2(1-x)^2y^2(1-y)^2$  and impose a zero boundary condition on  $\partial\Omega$ . The solution of this problem develops regular boundary layers at the outflow boundaries  $x = 0$  and  $y = 0$ . Figure 3.10 shows the numerical solution obtained on an anisotropically adapted mesh of around 3,000 nodes. It can be clearly seen from

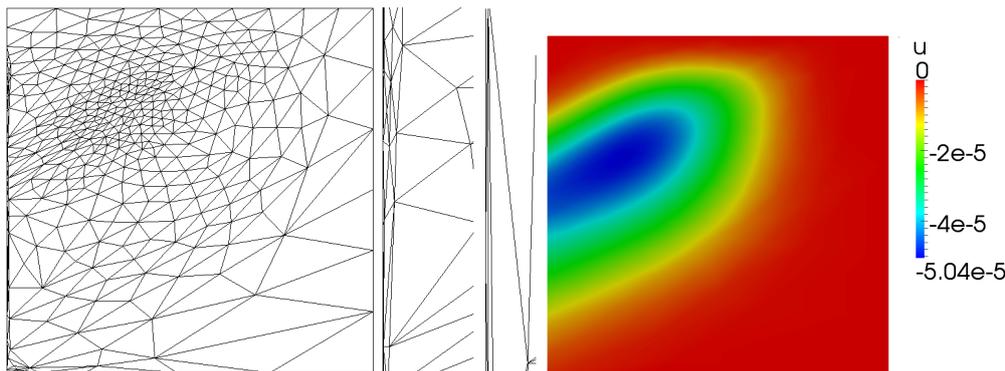


Figure 3.10 – Numerical solution for  $\kappa = 10^{-6}$  with its corresponding anisotropically adapted mesh.

the numerical results that the numerical scheme is capable of producing a smooth solution i.e. treating the spurious oscillations that appear when using the standard Galerkin formulation. Again the mesh responds very well to the solution with high resolution at the boundary layer and less dense elements in the rest of the domain. The elements are stretched in the direction of the boundary layer allowing its sharp capture.

**Example 4: Regular and corner boundary layers with non-smooth data**

In this numerical example, we consider a non constant convection field

$$\mathbf{v}(x, y) = -(2 + x^2y, 1 + xy)^T$$

with a small diffusion coefficient  $\kappa = 10^{-6}$  on the domain  $\Omega = (0, 1)^2$ . We define the source term as  $f(x, y) = -(x^2 + y^3 + \cos(x + 2y))$  and impose the following boundary condition:

$$u(x, 0) = u(0, y) = 0.$$

$$u(x, 1) = \begin{cases} 4x(1-x) & x < \frac{1}{2} \\ 1 & \text{otherwise} \end{cases} \quad u(1, y) = \begin{cases} 8y(1-2y) & y < \frac{1}{4} \\ 1 & \text{otherwise} \end{cases}$$

We expect the solution of this problem to develop regular and corner boundary layers at the outflow boundaries  $x = 1$  and  $y = 0$ . In figure 3.11, we present the numerical

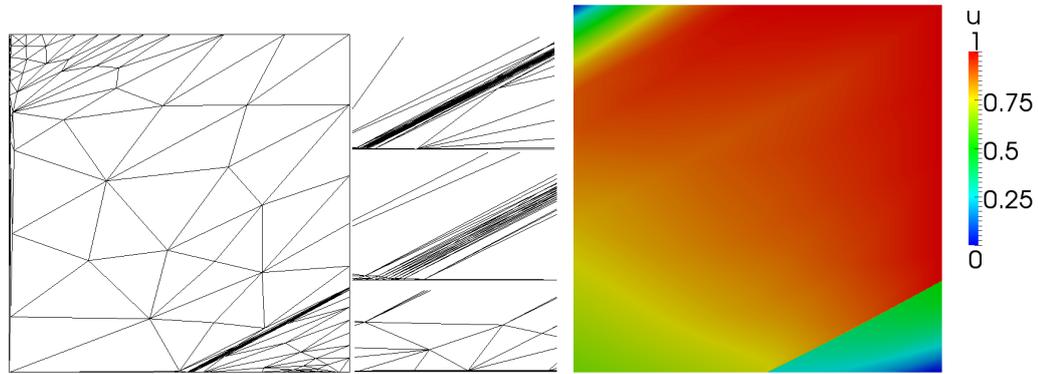


Figure 3.11 – Numerical solution for  $\kappa = 10^{-6}$ , its corresponding anisotropically adapted mesh with different zooming levels near the left boundary.

solution on a 3,000 nodes mesh obtained using the SUPG method and the mesh adaptation tool. We can observe the sharp stretching of the elements and the fine resolution at the regular and corner boundary layers and the location of the solution's steep gradient. We can also detect the gradual change of mesh element sizes in the zoom near the lower bound of the domain. It can be identified that the mesh has a better refinement in the orthogonal direction along the steep boundary layers. These observations reflect the ability of the anisotropic mesh adaptation algorithm to automatically adjust the shape and orientation of the elements while optimizing their numbers.

**Example 5: Parabolic layers**

We consider the domain  $\Omega = (0, 1)^2$ , the diffusion coefficient  $\kappa = 10^{-3}$ , a zero source term and the following boundary condition:

$$u(x, y) = \begin{cases} \left(6\sqrt{3}x(1-x)(2x-1)\right)^3 & \text{on } y = 0 \\ 0 & \text{otherwise on } \partial\Omega \end{cases}$$

The velocity field is set to  $\mathbf{v} = -(1,0)$ . Given the setting of the problem and the boundary conditions, we expect parabolic layers at  $y = 0$  with positive values for  $x > \frac{1}{2}$  and negative values for  $x < \frac{1}{2}$ . The solution obtained on an anisotropic mesh

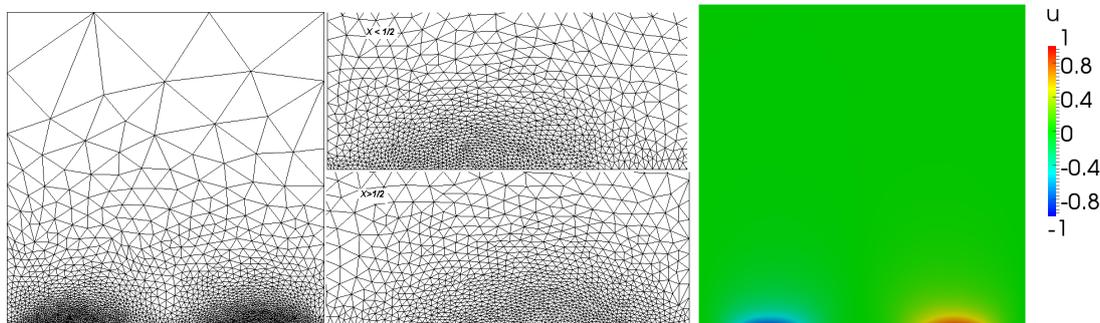


Figure 3.12 – Numerical solution for  $\kappa = 10^{-3}$  (right), its corresponding anisotropically adapted mesh with close-ups at the lower boundary to the left(middle-top) and right(middle-bottom) of the line  $x = \frac{1}{2}$ .

made of around 5,000 nodes is shown in figure 3.12. One can obviously notice the symmetric orientation and distribution of the elements in the mesh in response to the symmetric and parabolic profile of the solution. We also highlight the good quality of the automatically generated elements.

#### Example 6: Regular, Parabolic and Corner layers

We consider the same domain and constant velocity field as in the previous example, a zero source term and a diffusion coefficient  $\kappa = 0.05$ . The boundary condition is defined by:

$$u(x, y) = \begin{cases} (1-x)^2 & \text{on } y = 0 \\ y^2 & \text{on } x = 0 \\ 0 & \text{otherwise on } \partial\Omega \end{cases}$$

The solution as shown in figure 3.13 exhibits a parabolic boundary layer along  $y = 0$  and a regular one along  $x = 0$ . Corner boundary layers are detected at  $(0,0)$  and  $(0,1)$ . We can observe in the plot of the contour lines that the regular boundary layer has a steeper gradient than the parabolic one. One can easily see the high condensation of the elements near the corner layers and the gradual coarsening as we move away from them.

#### Example 7: Parabolic boundary layers with recirculating convective field

We consider the domain  $\Omega = (-1,1)^2$  and a non-constant flow with a diffusion coefficient  $\kappa = 0.05$  and a recirculating convective field defined by  $\mathbf{v} = (2y(1-x^2), -2x(1-y^2))$  exhibiting discontinuities at some parts of the boundaries leading to boundary layers. This test case is known as the double glazing problem [Elman 05] and models the spread of heat in a box with a heated wall. For this problem with complex physics, it is challenging to capture the structure of the parabolic layers. The results show the importance of the anisotropic mesh adaptation in capturing the

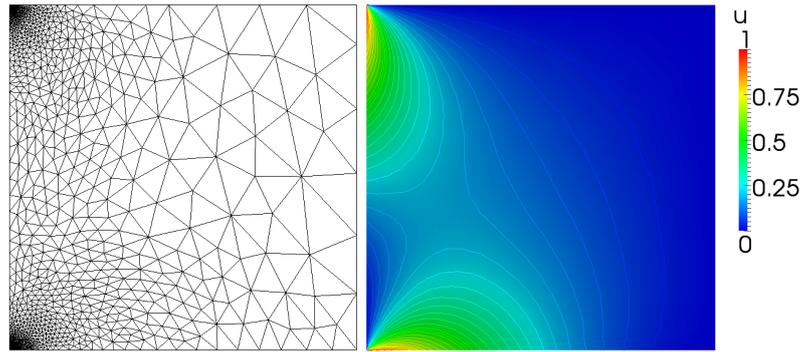


Figure 3.13 – Numerical solution for  $\kappa = 10^{-3}$  and its corresponding anisotropically adapted mesh.

details of the solution for such a challenging problem. Figure 3.14 presents the contour lines of the obtained numerical solution on an anisotropic mesh made up of around 3,000 nodes.

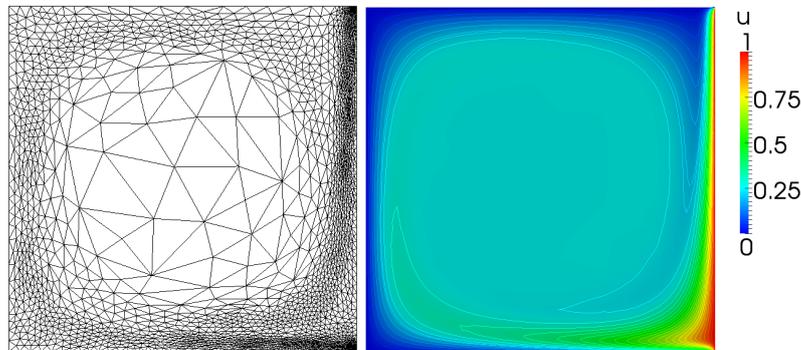


Figure 3.14 – Numerical solution for  $\kappa = 10^{-3}$  and its corresponding anisotropically adapted mesh.

### 3.4 APPLICATIONS TO INCOMPRESSIBLE FLOW PROBLEMS

In this section, we are interested in validating the developed anisotropic mesh adaptation tool on flow problems. A variational Multiscale method will be used for stabilization.

As we are interested in well capturing the large-scales of the solution, while modelling the effects of the fine-scales, we employ the method based on the averaged length distribution tensor for gradient recovery, the edge-based error estimates and the equi-distribution principle. This method proved to provide a good level of accuracy while maintaining a reduced computational cost. It focuses on automatically capturing evolving gradients of the solution under the constraint of a fixed number of nodes.

It is important to mention that the method tends to refine the mesh in the hierarchical importance of the solution's gradient. In other words, if new features (with high gradients) appear in the solution, the mesh will be automatically coarsened in regions with lower gradient and will be refined near the newly emerging features. In this case,

even with small number of nodes, the solution will still be well captured although not with the same degree of accuracy.

The performance of the proposed anisotropic mesh adaptation will be assessed on the benchmark driven cavity problem at low and high Reynolds numbers. In the following simulations we show that boundary layers as well as eddies can be well captured by the mesh. Results are compared with very accurate reference solutions taken from the literature and show that the flow solvers based on stabilized finite element method produce very accurate solutions on anisotropic meshes with highly stretched elements. Furthermore, new measurements positioned near boundary layers (see fig. 3.15) are presented in this section. The edge-based error estimation is applied on the velocity field; more precisely on its direction and its norm. We will discuss in chapter 5 how the metric construction is modified to take into account several fields for error estimation.

We present herein the most relevant results for the validation of the accuracy and efficiency of the coupling between the VMS approach and the anisotropic mesh adaptation technique.

Further analysis was conducted on this problem in order to provide benchmark results on the time dependent two and three dimensional incompressible Navier-Stokes equations with anisotropic mesh adaptation. It demonstrated the capability of the anisotropic mesh adaptation tools to capture the boundary layers developed by the numerical solution. The interested reader can refer to [Coupez 13a]. Such a new set of data can serve as a useful benchmark for comparison.

### 3.4.1 Driven flow cavity problem (2D)

We begin our validation of the adaptive technique by considering the classical 2D lid-driven flow problem with a zero source term. This test has been widely studied by a number of authors [Ghia 82, Sahin 03] using finite difference and finite volume methods, and [Hughes 86c, Franca 92b, Hachem 10b, Coupez 13a] using stabilized finite element methods. It serves as a benchmark for numerical methods. The computational domain is a unit square  $[0, 1]^2$ . We impose homogeneous Dirichlet boundary conditions for the velocity equal to one on the top boundary of the computational domain, i.e. at  $y = 1$ , and zero elsewhere. We set the density to  $1\text{kg}/\text{m}^3$  and adjust the viscosity in order to obtain Reynolds number 1,000, 5,000, 10,000, 20,000, and 100,000. We evolve the solution in time with a fixed time-step equal to 0.05s. We recall a key feature of the proposed mesh adaptation technique which is its capability to work with a fixed degree of freedom. Usually this number is set in accordance with the available computational resources. This feature is a main advantage as it avoids a drastic increase in mesh complexity and hence in the computational cost, consequently constituting a powerful tool for industrial applications. We fix the number of nodes to 10,000 in the following test cases for Reynolds number 1000, 5000 and 10,000. Starting with a uni-

form mesh presented in figure 3.15, we apply the mesh adaptation process every five time-steps. Note that a frequency of adaptation is applied in the view of reducing the computation time due to adapting the mesh. It is considered as an acceptable tradeoff between efficiency and accuracy.

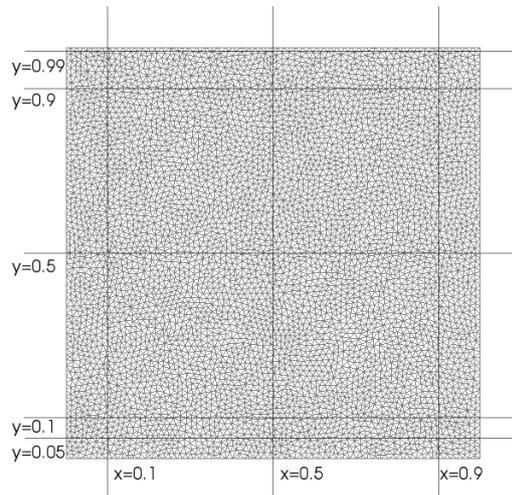


Figure 3.15 – Driven cavity problem: 2D computational domain and the different cross sections.

Figure 3.16 presents the three resulting converged meshes (containing exactly 9,854 nodes). We highlight the high resolution not only along all the boundary layers but also at the detachment regions. This reflects the anisotropy of the solution caused by the discontinuity of the boundary conditions and the nature of the flow. The elements at the central bulk of the cavity region around the primary vortex are mostly isotropic and increase in size as the Reynolds number increases. Again, this reflects and explains how, for a controlled number of nodes, the mesh is naturally and automatically coarsened in that region with the goal of reducing the mesh size around the secondary vortices in the bottom corners.

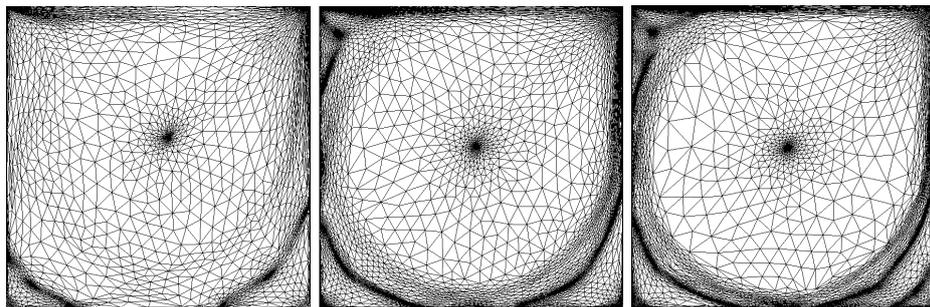


Figure 3.16 – Driven cavity problem: anisotropic meshes at Reynolds 1000, 5000 and 10,000.

As the Reynolds number increases, secondary eddies emerge in the domain. The mesh adaptive algorithm responds to this phenomena by refining at the level of these eddies and coarsening the domain regions exhibiting lower velocity gradient. We can clearly notice by comparing the meshes obtained for increasing Reynolds number, i.e. with emerging secondary vortices, how the elements are automatically and anisotropi-

cally adjusted to allow a good capture of these vortices. This is mainly due to the choice of the unit vector combining the velocity components and its norm. The proposed approach shows that, with around 10,000 nodes, it is capable of capturing all the vortices at the different scales. Indeed, the small eddies are not strong enough to be captured accurately by classical adaptive strategies.

The close-up on the right side of the cavity in Figure 3.17 highlights how sharp the layers' capture is. It shows the correct orientation and deformation of the mesh elements (longest edges parallel to the boundary). This yields a great reduction in the number of needed degrees of freedom to provide a good level of accuracy. These results give confidence that the extension of the approach to take into account all velocity components seems to work very well and plays an important role for transient flows. Note that the details on how the metric construction is extended to take into account several fields are covered in chapter 5.

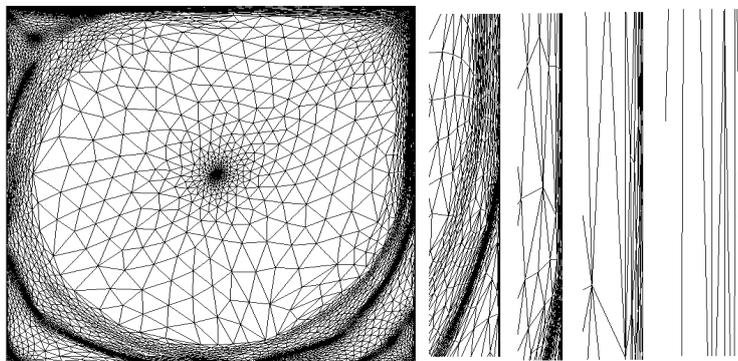


Figure 3.17 – Driven cavity problem: zooms on the mesh near the right wall.

At this point, the next objective of these simulations is to compare the obtained results to very accurate reference solutions. Therefore, we first plot the velocity profiles for  $V_x$  along  $x = 0.5$  at Reynolds 1000, 5000 and 10,000. Figure 3.18 shows respectively that all the results are in very good agreement with a reference solution computed on a  $600 \times 600 = 360000$  points fixed mesh [Hachem 10b].

For further comparison, we provide in Table 3.1 values of the vertical and horizontal components of the velocity along the line  $x = 0.5$ . We report the results obtained with simulations on a fine isotropic mesh (600x600) and second from the converged anisotropic meshes ( $\sim 10,000$  nodes). We compare these results to very precise computations found in [Botella 98] and [Guermont 11] performed on a uniform grid composed of  $5000 \times 5000$  nodes. The presented solution on an isotropic (600x600) unstructured mesh highly correlates with the references. Values agree, along with the profile, up to the fourth digit. On the other hand, results on anisotropic meshes compare well and agree up to the second digit with the highly accurate solutions computed on  $5000 \times 5000 = 25,000,000$  points [Guermont 11]. Again, results obtained with the VMS flow solver using either isotropic or anisotropic meshes coincide very well with the results in the literature.

It is worth mentioning that there is a lack, in the literature, of reference data at high

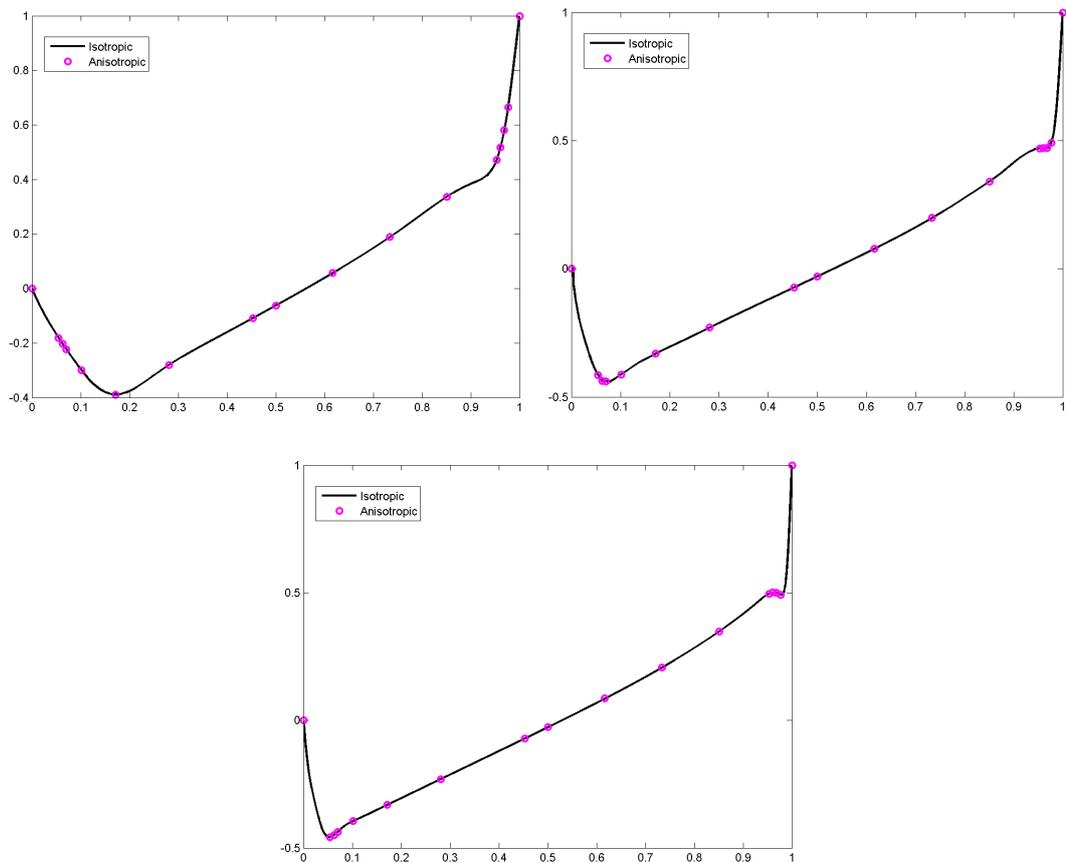


Figure 3.18 – Driven cavity problem: comparison of the first component of the velocity field  $V_x$  in the mid-plane  $x = 0.5$  for  $Re = 1,000$  (left), for  $Re = 5,000$  (middle) and for  $Re = 10,000$  (right).

| y      | Ref [Botella 98] | Ref [Guermond 11] | Isotropic  | Anisotropic |
|--------|------------------|-------------------|------------|-------------|
| 1.0000 | 1.0000000        | 1.0000000         | 1.0000000  | 1.0000000   |
| 0.9766 | 0.6644227        | 0.6644194         | 0.6644520  | 0.6627950   |
| 0.9688 | 0.5808359        | 0.5808318         | 0.5808070  | 0.5798160   |
| 0.9609 | 0.5169277        | 0.5169214         | 0.5168170  | 0.5126750   |
| 0.9531 | 0.4723329        | 0.4723260         | 0.4721800  | 0.4709190   |
| 0.8516 | 0.3372212        | 0.3372128         | 0.3370270  | 0.3359990   |
| 0.7344 | 0.1886747        | 0.1886680         | 0.1885570  | 0.1883090   |
| 0.6172 | 0.0570178        | 0.0570151         | 0.0569487  | 0.0572112   |
| 0.5000 | -0.0620561       | -0.0620535        | -0.0620790 | -0.0616432  |
| 0.4531 | -0.1081999       | -0.1081955        | -0.1082040 | -0.1078430  |
| 0.2813 | -0.2803696       | -0.2803632        | -0.2803280 | -0.2808320  |
| 0.1719 | -0.3885691       | -0.3885624        | -0.3883390 | -0.3863410  |
| 0.1016 | -0.3004561       | -0.3004504        | -0.3001060 | -0.3009820  |
| 0.0703 | -0.2228955       | -0.2228928        | -0.2225900 | -0.2243200  |
| 0.0625 | -0.2023300       | -0.2023277        | -0.2020430 | -0.2041190  |
| 0.0547 | -0.1812881       | -0.1812863        | -0.1810220 | -0.1828680  |
| 0.0000 | 0.0000000        | 0.0000000         | 0.0000000  | 0.0000000   |

Table 3.1 – Driven cavity problem: velocity profiles on the median plane  $x = 0.5$  at  $Re = 1000$ .

Reynolds number. In particular, measurements near the walls cannot be found in the literature. Thus, it makes the comparison and validation of the results on anisotropic meshes at high Reynolds number difficult. Therefore, we will use in what follows our results carried out on the isotropic meshes (600x600) as references.

We perform cross sections close to the walls; these are crucial for validating the efficiency and accuracy in capturing the boundary layers. We present the new plots taken near the corners in figure 3.19.

We show in these figures the velocity profiles for  $V_x$  along  $x = 0.1$  and  $x = 0.9$  at Reynolds 1000, 5000 and 10,000. All the results are in very good agreement with the proposed reference having 600x600 elements. It is clear that in regions of strong vortices and very close to the walls, the use of anisotropic meshes plays a critical role. All the boundary layers are sharply captured and automatically identified. We can clearly see from these plots that the creation of extremely stretched elements near the walls provides accurate results. The reference computed on a very fine isotropic mesh (600x600) agrees with the anisotropic results obtained using only 10,000 nodes.

Finally, in order to assess the capability of the VMS method to simulate high Reynolds number flows on anisotropic meshes, we repeated the simulation for Reynolds numbers 20,000 and 100,000. Note that details on obtaining the steady solution for different Reynolds numbers is analyzed by the authors in [Hachem 10b] and will not be repeated here.

We increased the number of nodes to 20,000 in order to capture more accurately the characteristic of the unsteady solution. We present in figure 3.20 snapshots of the meshes at a certain time instance. As expected, all the main directional features characterizing the velocity inside the lid-driven cavity are detected and well captured by the anisotropic error estimator. The mesh elements are highly stretched along the direction of the layers, at the detachments regions and around the developing secondary vortices. Once again, the developed incompressible Navier-Stokes VMS solver proves to be very efficient and robust at high Reynolds numbers using highly stretched elements. Again, for a given constraint on the number of elements we can find the mesh that maximizes the accuracy of the numerical solution.

To have a clear idea on the efficiency of the developed anisotropic mesh adaptation in simulating high Reynolds flows, we summarize in table 3.2 the history of CPU time needed to achieve 20 seconds of simulation with and without mesh adaptation. The fixed mesh used for comparison is the one that yields the same level of accuracy as in computing the solution with the anisotropic mesh adaptation. We also report the percentage of simulation time spent by the adaptation algorithm. We notice that on average the anisotropic mesh adaptation allows 5 times faster computations, and it uses almost 20% of the simulation CPU time.

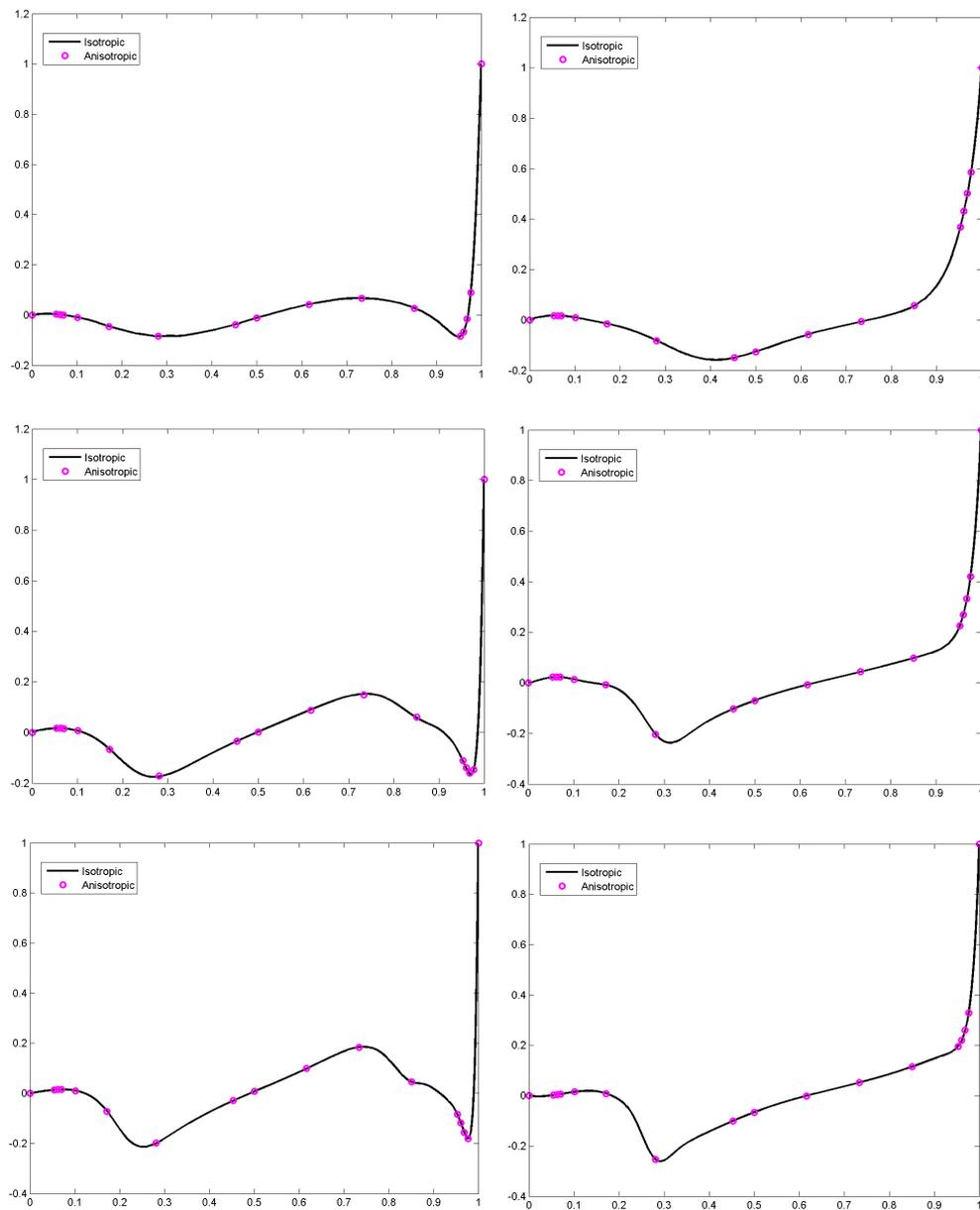


Figure 3.19 – Driven cavity problem: comparison of velocity profiles for  $Re = 1,000$  (top), for  $Re = 5,000$  (middle) and for  $Re = 10,000$  (bottom). Left: Velocity profiles for  $V_x$  along  $x = 0.1$ . Right: Velocity profiles for  $V_x$  along  $x = 0.9$ .

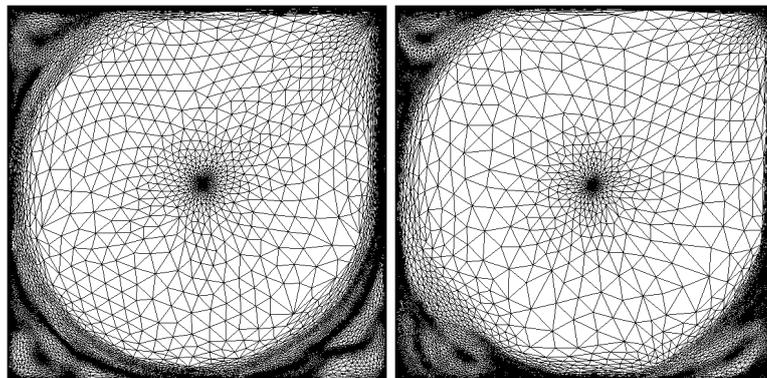


Figure 3.20 – Driven cavity problem: snapshots of the anisotropic meshes for Reynolds 20,00 and 100,000

| Reynolds | CPU time (fixed) | CPU time (adaptive) | % on anisotropic mesh | Speedup |
|----------|------------------|---------------------|-----------------------|---------|
| 1000     | 7655.1           | 1822.8              | 21                    | 4.2     |
| 5000     | 8158.6           | 2022.27             | 21.4                  | 4       |
| 10000    | 8563.46          | 2320.3              | 20.1                  | 3.7     |
| 20000    | 7978.2           | 3008.3              | 17.85                 | 2.6     |
| 100000   | 7427.8           | 4956.7              | 20.7                  | 1.5     |

Table 3.2 – History of CPU time needed to achieve 20s of simulation.

### 3.4.2 Driven flow cavity problem (3D)

We continue our investigation on the classical lid-driven flow problem and consider the three dimensional case on the cubic domain  $[0,1]^3$ . A zero source term is employed. Dirichlet boundary conditions are imposed on the velocity field:  $v = 1$  on the upper wall at  $z = 1$   $v = 0$  elsewhere. The solution is evolved with a fixed time-step equal to 0.05s. The viscosity is adjusted in order to obtain Reynolds number of 1000, 3200 and 5000.

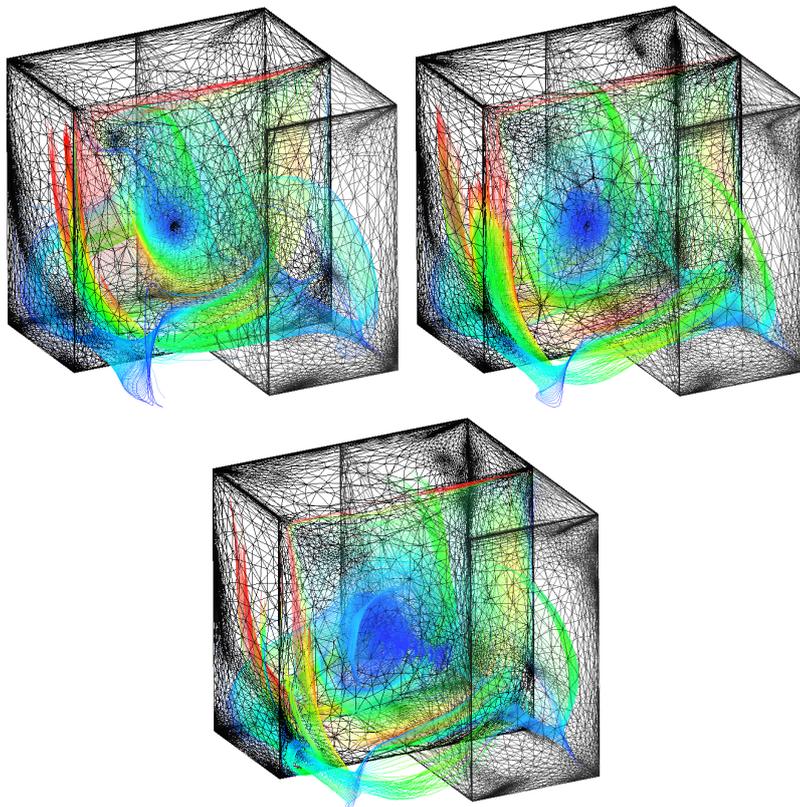


Figure 3.21 – Streamlines snapshots at Reynolds 1000, 3200 and 5000

All numerical experiments are done using a fixed number of nodes ( $\sim 150,000$ ). Again, in 3D where the flow characteristics are much more complicated, all the boundary layers as well as the vortices are sharply captured and identified.

Figure 3.21 highlights the stretching of the elements near the walls, in the corners and even at the bifurcations of the eddies for Reynolds number equal to 1000, 3200 and 5000 respectively. One observes that the higher the Reynolds number, the denser the mesh around these regions. The velocity profiles for  $V_x$  along  $x = 0.5$  at Reynolds 1000

and 3200 are presented in Figure 3.22. All the results are in good agreement with the reference having a twice finer mesh of  $96 \times 96 \times 96$  grid points [Albensoeder 05].

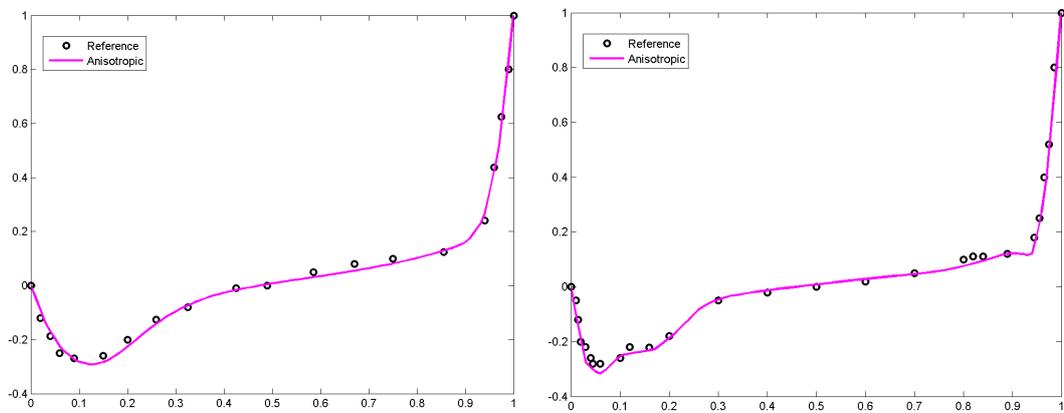


Figure 3.22 – Comparison of the velocity field's first component in the mid-plane  $x = 0.5$  for  $Re = 1,000$  (left) and for  $Re = 3,200$  (right).

In figure 3.23 we present different close-ups on the mesh, corresponding to  $Re = 3200$ , to show how the elements can be highly stretched along the direction of the layers. This reflects well the accuracy and details of the resolution caused by the discontinuity of the boundary conditions and the nature of the flow.

The anisotropic adaptive procedure modifies the mesh so that the local mesh resolutions become adequate in all directions. Recall again that these plots reflect for the given fixed number of nodes ( $\sim 150,000$ ) the mesh that optimizes the accuracy of the numerical solution. The presented test cases were considered here in the objective of demonstrating the capability of the VMS method with the modified characteristic length to simulate 2D and 3D high Reynolds number flows on anisotropic meshes.

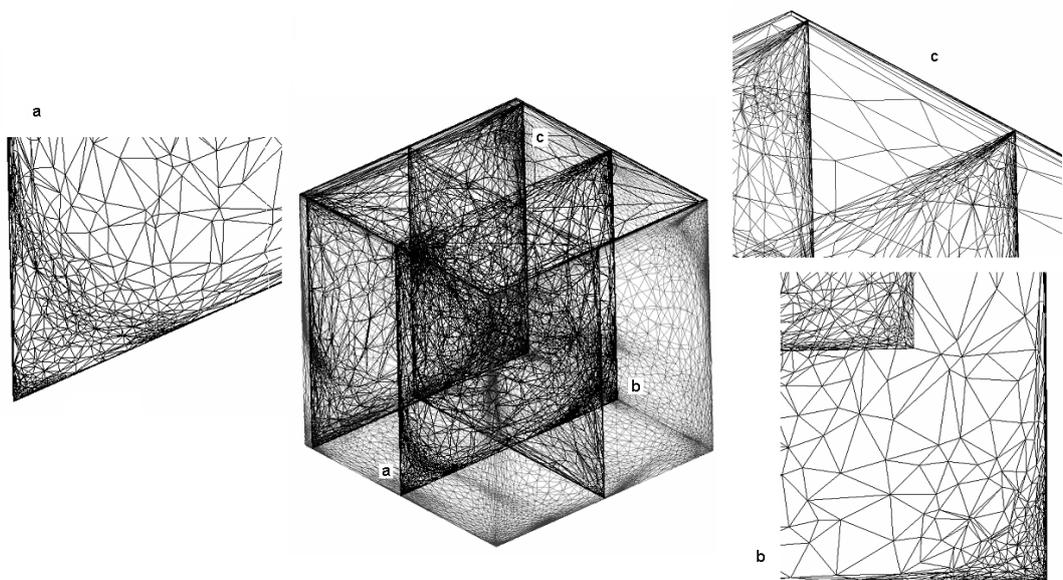


Figure 3.23 – Zooms on the mesh details inside the 3D cavity

### 3.4.2.1 Flow past a Naca 0012 airfoil

In this test case we intend to evaluate the capability of the stabilized finite element method together with the *Spalart-Allmaras* turbulence model to handle turbulent flows in the presence of our developed anisotropically adapted meshes. The steady flow around a NACA 0012 airfoil is a fundamental benchmark in aerodynamic simulations for its simplicity and ability to reflect many features that may appear in aircraft applications. The results that we provide in what follows are presented in [Cremonesi 14].

The anisotropic mesh adaptation coupled with the VMS approach and the *Spalart-Allmaras* model is assessed on a fully turbulent high Reynolds number flow reproduced from [Rumsey 13]. We immerse a NACA body in a sufficiently large domain to avoid the influence of the farfield boundary conditions on the nature of the flow near the airfoil.

Thus we set the domain to  $8 \times 20$  chords as shown in figure 3.24. Respecting the incompressibility of the flow, we prescribe periodic boundary conditions at the inlet and outlet. We choose the inlet boundary condition to yield a Reynolds number of  $Re = 6 \times 10^6$  and a Mach number  $Ma = 0.2$ . No slip conditions are imposed on the horizontal boundaries  $\mathbf{v} \cdot \mathbf{n} = 0$ ,  $\boldsymbol{\sigma} \mathbf{n} = \mathbf{0}$  and a null farfield pressure value is imposed.

At the level of the NACA body, no slip conditions are imposed on the velocity field coupled together with homogeneous Dirichlet conditions for the SA working variable. The time-step size is fixed to  $\Delta t = 10^{-4}$ s to ensure stability.

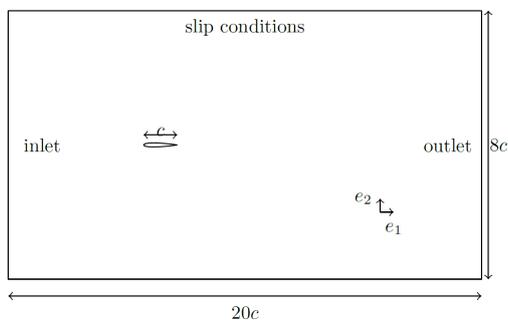


Figure 3.24 – Geometry for the flow behind a NACA body test case.

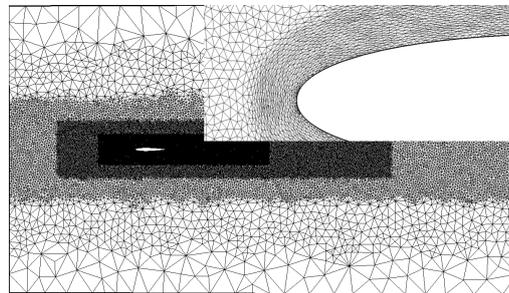


Figure 3.25 – Mesh around the NACA body with a detail of the anisotropy.

In this aerodynamic application, for comparison purposes, we are interested in the drag force on the airfoil. We set the mesh size in the fixed mesh to  $2.2 \times 10^{-5}$  resulting in a  $y^+$  value around 5 based on Schlichting's formula [Schlichting 68]. We investigated two aspects of mesh adaptation: the number of nodes, and the variables driving the adaptation. Note that, a poor mesh yields an unresolved boundary layer and may cause failure of the simulation's predictive abilities. First of all, exploiting the multi criteria formulation, that will be discussed in chapter 5, we tested different variable combinations (see fig. 3.26): pure velocity, velocity and pressure, velocity and turbulent viscosity. Figure 3.27 shows the obtained meshes at the stationary state: we infer that the pressure field is too localized to reflect the flow features. Nevertheless, when the

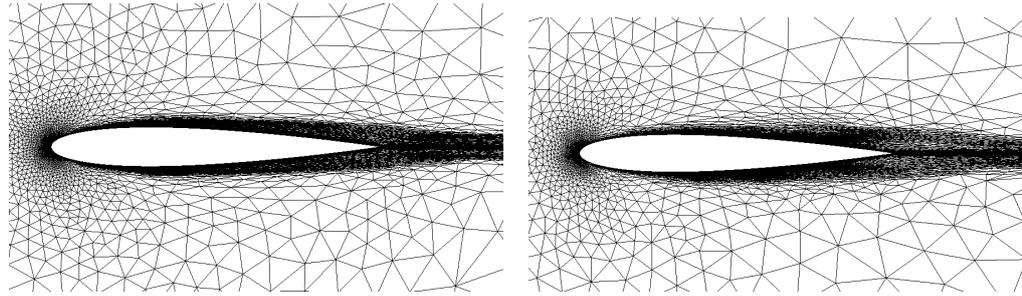


Figure 3.26 – Adaptation driven by the velocity field (left) and a combination of velocity and turbulent viscosity (right).

adaptation is performed on the velocity and turbulent viscosity, one can detect step anisotropic characteristics as a response to the sharp gradients of the latter two fields. Notice the high density of the nodes in the wake and its gradual anisotropic decrease as we move away. On the other hand, when adapting on the viscosity together with the

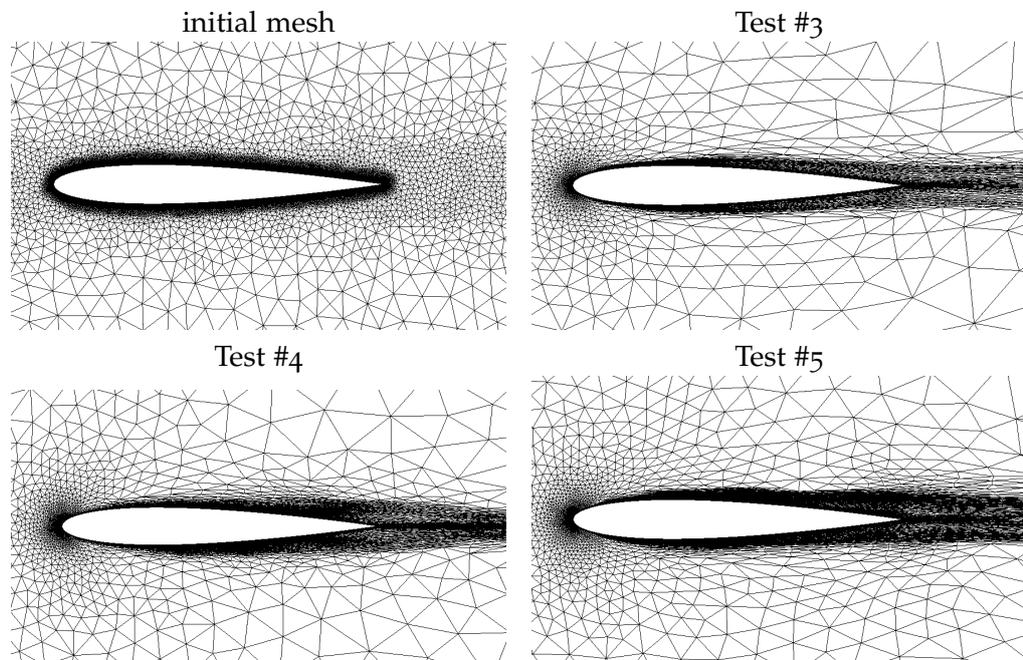


Figure 3.27 – Plots of initial and stationary state meshes.

velocity field, qualitatively better features can be explored. We adopt this combination for the rest of our analysis.

The test case configuration and level of anisotropy are reported in table 3.3. The results

|                            | Test #1 | Test #2 | Test #3                            | Test #4                            | Test #5                            | Test #6                  |
|----------------------------|---------|---------|------------------------------------|------------------------------------|------------------------------------|--------------------------|
| No. elements $\times 10^5$ | 0.5     | 1.5     | 0.5                                | 1.0                                | 1.5                                | 1.0                      |
| Adaptation                 | no      | no      | $\underline{\mathbf{v}}$ & $\nu_t$ | $\underline{\mathbf{v}}$ & $\nu_t$ | $\underline{\mathbf{v}}$ & $\nu_t$ | $\underline{\mathbf{v}}$ |

Table 3.3 – Summary of test cases.

appear quite promising when confronted with the ones presented in [Rumsey 13]. The effect of the mesh size is clearly visible from tests 3,4 and 5. The adaptation procedure in test 3 does not have enough degrees of freedom to well capture the interface, and a

|                      | Reference        | Test #1 | Test #2 | Test #3 | Test #4 | Test #5 | Test #6 |
|----------------------|------------------|---------|---------|---------|---------|---------|---------|
| $C_D \times 10^{-3}$ | $8.155 \pm 0.35$ | 8.12    | 8.16    | 7.90    | 8.12    | 8.16    | 8.10    |
| Relative error       |                  | 0.3%    | 0.1%    | 3%      | 0.3%    | 0.1%    | 0.6%    |

Table 3.4 – Computed drag coefficients for different meshes.

significant error in the drag computation is committed (note that this effect was artificially taken into account in test 1, at the expense of a very long and consuming mesh construction procedure rendering an important computational cost). It is important to mention that in test 1, the mesh is designed before starting computations, hence it required an apriori knowledge about the nature of the flow, so if new features were to be produced in some other location, away from the interface, the mesh will not be accounting for it and hence yields an important error. On the other hand, using the proposed anisotropic mesh adaptation algorithm, the mesh will be automatically adapted and updated during runtime without apriori knowledge about the problem at hand. Test 6 reveals that driving the mesh from the velocity and turbulent viscosity reduces the induced numerical error. As may be seen from table 3.4 all the tests except test 3 are in a 1% range of error thus confirming the predictive performance and reliability of this method. The final result plots for test 4 are shown in fig. 3.28 and appear in good agreement with similar tests in the literature.

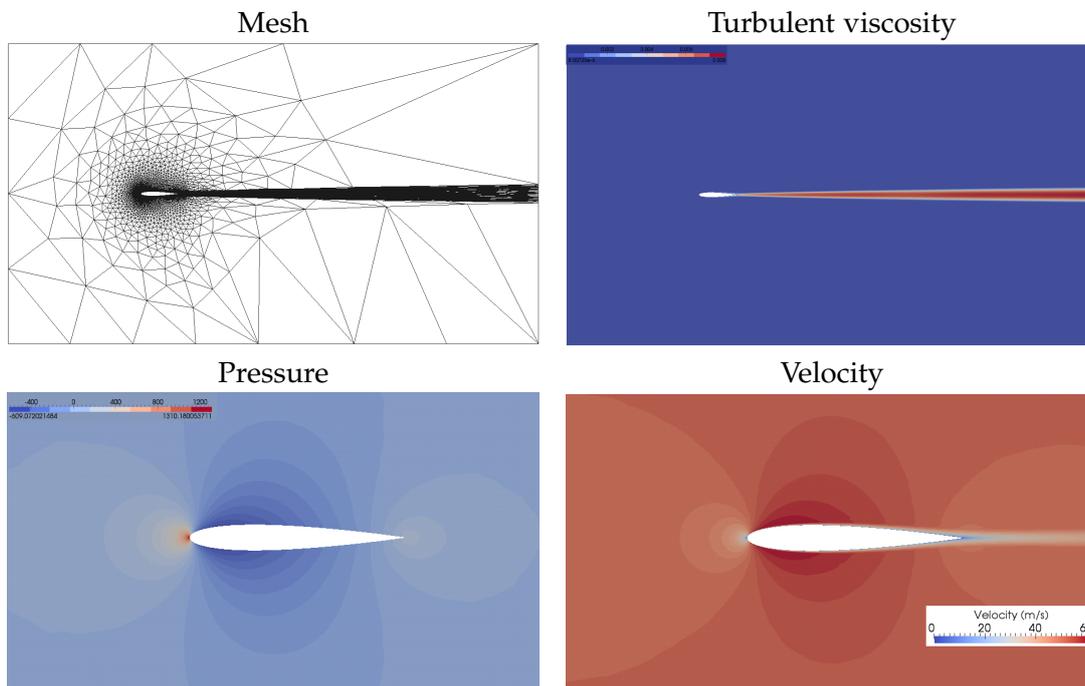


Figure 3.28 – Plots for Test # 4 at the final time.

### 3.4.3 Application to a 3D coupled heat transfer and fluid flows problem

So far, we have tested the anisotropic mesh adaptation algorithm separately on the convection-diffusion and the Navier-Stokes problems. In this test case we apply the developed anisotropic mesh adaptation to simulate the coupled heat transfer and fluid

flows inside an industrial furnace with complex geometry presented in [Hachem 09]. The objective of this numerical example is to show the applicability, efficiency and potential of the developed algorithm in simulating long time heating inside large scale complex three-dimensional enclosure. Stabilized finite element method is used to numerically solve time-dependent, three-dimensional conjugate heat transfer and turbulent fluid flows. More precisely, two additional stabilization terms have been introduced for the stabilization of the transient convection-diffusion-reaction equations: the first controls the oscillations in the streamline direction (SUPG) and the second controls the derivatives in the direction of the solution's gradient (SCPG). Moreover, the variational multiscale approach is applied for stabilizing the Navier-Stokes equations. Starting with a gas at rest and a constant temperature of 333K, the heated air is pumped into the furnace from 10 different inlets, with circular shape having 6m diameter and positioned at 1.7m from the ground. For confidentiality purposes, we omit the details on the geometry of the furnace. The velocity and temperature of injection are set to 10m/s and 1073K respectively. Adiabatic and no slip conditions are assumed on all the other boundaries for sake of simplicity. The air is vented out the furnace through an outlet located at the center of the top wall (at  $z = 2.42\text{m}$ ). The solution is evolved in time with a time-step size equal to 0.01s. The computations have been conducted using 16 2.4 Ghz Opteron cores in parallel (linked by an Infiniband network) [Digonnet 03].

We have anisotropically adapted the mesh every 10 time-steps with a fixed imposed mesh size of around 60,000 nodes. In fact, in such a simulation we are interested in accurately modeling the thermal evolution inside the enclosure. A combination of the temperature field, the velocity direction and the velocity norm is considered as the motor in driving the mesh toward optimality. The latter multi-criteria adaptation shall be discussed in chapter 5.

Figure 3.29 presents a median plane cut in the  $z$  – *direction* showing the evolution of the isothermal distribution and the velocity streamlines at different time-steps. Starting with a uniform isotropic mesh, the mesh adaptation technique automatically detects the gas pumping inside the furnace. As a response, an anisotropic refinement at the level of the burners is generated allowing a better representation and capture of the flow. When the hot fluid spreads along the volume of the furnace, it induces a turbulent and recirculating motion within the geometry. This forced convection is caused by the interaction of the moving stream and the stationary fluid inside the furnace. The streamlines and the temperature distribution clearly reflect the expected flow pattern. Again, the mesh elements are automatically redistributed and well oriented to give a better capture of the multi-scale flow characteristics and represent the emerging gradients.

The highly stretched elements and significant density at the level of the temperature gradient permit a better capture of the flow characteristics and the temperature field.

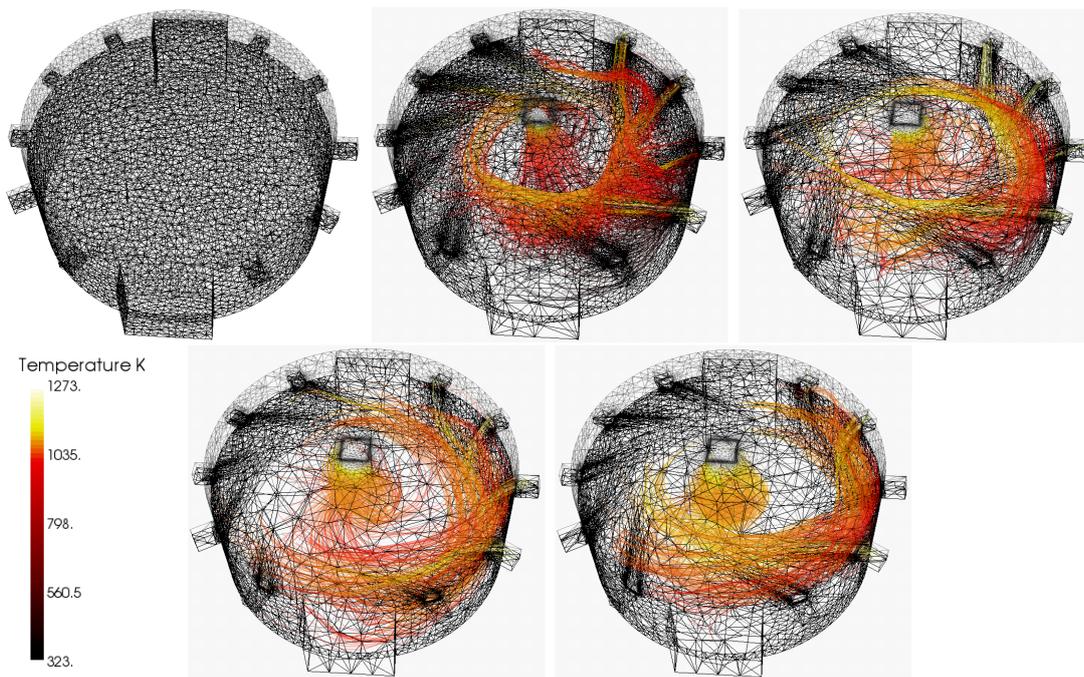


Figure 3.29 – Streamlines and isotherms inside the furnaces at different time instances.

One can observe a number of vortexes inside different buffer zones. They are due to the turbulence dissipation and mixing between the hot and cold air.

The objective of this simulation is to assess the applicability of the anisotropic mesh adaptation in simulating coupled heat transfer and fluid flows inside a complicated geometry. The problem can be perceived as a first step toward a complex real application. Comparative analysis in real contexts will be provided in chapter 7. The present results, describe only the first five minutes of the heating process, which required around 10 hours of computations. This limitation in the computational time is due to the small chosen time-step size. To provide the same level of accuracy, a fixed mesh should be fine everywhere in the domain in order to capture the spread of the velocity and temperature fields all over the furnace. Hence, a speedup of computations is highlighted using an anisotropic mesh adaptation.

However, it is still necessary to supply fast calculation in order to calculate full heating sequences inside industrial furnaces, with different ingots, in a reasonable computational time. These calculations would allow to predict different parameters and to understand the flow characteristics for heat treatment furnaces. We recall that one of the main industrial objectives of this work is to model and simulate conjugate heat transfer for multi-components systems which is considered very essential for industrial applications especially in the case of the heat treatment of high-alloy steel by a continuously heating process inside industrial furnaces.

### 3.5 CONCLUSION

In this chapter, we have modified the definition of the characteristic length used in the stabilization parameters to account for highly stretched elements. The performance of the developed anisotropic mesh adaptation algorithm and the modified solvers have been validated on several physical phenomena. We demonstrated the efficiency of the method in accelerating the computations for the same level of accuracy with respect to a fixed mesh. We assessed the adaptation on convection-diffusion problems at high Peclet number. It showed to be favorable for capturing steep interior and boundary layers. We have also validated the order of convergence and showed that using the anisotropic mesh adaptation together with stabilized finite element methods, we are capable of recovering the convergence orders. When applied to high Reynolds flows, with or without turbulence models, the method reflected good accuracy in automatically producing boundary layer meshes inside the 2D and 3D lid-driven cavities. The numerical results demonstrated that the flow solvers based on stabilized finite element method is able to exhibit good stability and accuracy properties in the presence of anisotropic meshes. All the main directional features characterizing the velocity inside the cavities are detected and well captured by the anisotropic error estimator. The mesh elements are highly stretched along the direction of the layers, at the detachments regions and around all vortices.

We have also investigated the feasibility of the developed anisotropic mesh adaptation algorithm in simulating coupled heat transfer and fluid flows inside complex three-dimensional geometries. The results demonstrated the potential of the method in well capturing the characteristics of the simulation. This method has lead to a considerable improvement in the computational time required for the same level of accuracy on a fixed mesh. However, the use of a constant time-step all over calculations might yield a significant temporal slow-down. One way to resolve this issue can be by adopting varying time-step sizes according to the nature and features of the simulated physical phenomena. In other words, reducing the computational time can be obtained by decreasing the time-steps when the solution exhibits radical changes and increasing them as the solution stabilizes, i.e. when the simulation error decreases for a given mesh: this process is known as temporal adaptation. It will be the subject of the following chapter.

### 3.6 RÉSUMÉ FRANÇAIS

Dans ce chapitre, nous avons modifié la définition de longueur caractéristique utilisée dans les paramètres de stabilisation pour tenir compte des éléments très étirés. La performance de l'algorithme d'adaptation de maillage anisotrope développé et des solveurs modifiés a été validée à travers la simulation de plusieurs phénomènes physiques. Nous avons démontré l'efficacité de la méthode vis-à-vis d'un maillage fixe.

La méthode d'adaptation a été validée sur des problèmes de convection-diffusion à grand nombre de Péclet. L'approche a montré sa capacité de capturer les couches internes et limites. Dans le cadre de méthodes de stabilisation, un demi ordre de convergence est perdu par l'approche éléments finis  $P_1$ , ceci est dû aux termes de diffusion ajoutés aux endroits de convection dominante. Couplés avec la méthode d'adaptation anisotrope du maillage, le second ordre de convergence a été récupéré. Cette propriété a été validée sur plusieurs cas de convection-diffusion.

Ensuite, le couplage adaptation anisotrope de maillage et solveurs stabilisés a été mis à l'épreuve sur des simulations d'écoulements à de nombres de Reynolds élevés, avec ou sans modèles de turbulence. La méthode a reflété une bonne capacité à reproduire automatiquement les couches limites de manière précise. Toutes les principales caractéristiques directionnelles caractérisant la vitesse à l'intérieur des cavités sont bien détectées et capturées par l'estimateur d'erreur anisotrope. Les éléments du maillage sont fortement étirés le long de la direction orthogonale aux forts gradients, au niveau des régions de détachements et autour des tourbillons qui se développent. Nous avons également étudié la performance de l'algorithme d'adaptation et la robustesse des solveurs dans le cadre de simulations 3D d'écoulements de fluide et de transferts thermiques à l'intérieur de géométries complexes telles que les fours industriels. Cette méthode a conduit à une amélioration considérable du temps de calcul obtenue pour le même degré de précision sur un maillage fixe. Plus de validations numériques et des confrontations avec des résultats expérimentaux seront évoqués dans le chapitre 7.

Cependant, l'utilisation d'un pas de temps constant durant toute la simulation pourrait induire un ralentissement temporel significatif. Une façon de résoudre ce problème peut être en adoptant des pas de temps qui varient en fonction de la nature et des caractéristiques des phénomènes physiques simulés. En d'autres termes, la réduction du temps de calcul peut être obtenue en diminuant les pas de temps lorsque la solution présente des variations rapides et en augmentant les pas de temps lorsque la solution se stabilise ou reflète des variations lentes. Ce procédé est nommé l'adaptation de pas de temps et sera l'objet du chapitre suivant.

# TIME ACCURATE ANISOTROPIC ADAPTATION FOR UNSTEADY SIMULATIONS

"The bad news is time flies. The good news is you're the pilot"  
Michael Altshuler

## CONTENTS

---

|       |                                                                                 |     |
|-------|---------------------------------------------------------------------------------|-----|
| 4.1   | STATE OF THE ART                                                                | 187 |
| 4.1.1 | Stability constraints on explicit advancing                                     | 187 |
| 4.1.2 | Local time-stepping                                                             | 188 |
| 4.1.3 | Overview on existing time-stepping algorithms                                   | 189 |
| 4.1.4 | Generating anisotropic space-time meshes                                        | 191 |
| 4.2   | TIME ADAPTATION PROCEDURE                                                       | 194 |
| 4.2.1 | Edge-based temporal error estimation                                            | 195 |
| 4.2.2 | General space-time adaptive algorithm                                           | 198 |
| 4.3   | NUMERICAL VALIDATION                                                            | 199 |
| 4.3.1 | Convergence analysis on an analytical test case                                 | 199 |
| 4.3.2 | A two-dimensional analytical test case with sinusoidal evolution in time        | 203 |
| 4.3.3 | Applications on 2D fluid flow problems                                          | 203 |
| 4.3.4 | Flow around a circular cylinder (2-D)                                           | 211 |
| 4.3.5 | Application to the unsteady convection-diffusion problem                        | 213 |
| 4.3.6 | Convection-diffusion in a plane shear flow                                      | 214 |
| 4.3.7 | Internal and boundary layers                                                    | 215 |
| 4.3.8 | Application to coupled heat transfer and fluid flows                            | 217 |
| 4.3.9 | Application to 3D heat transfer and turbulent flow inside an industrial furnace | 222 |
| 4.4   | CONCLUSION                                                                      | 226 |
| 4.5   | RÉSUMÉ FRANÇAIS                                                                 | 227 |

---

**P**HYSICAL problems and industrial applications are intrinsically dynamic. Their resolution requires a great challenge as it involves following their unsteadiness with a good accuracy while respecting an affordable CPU time and memory storage. In the previous chapter, we have discussed the advantages of employing anisotropic mesh adaptation for the resolution of physical phenomena (heat transfer and fluid flows). The method permits an automatic and good capture of the physical phenomena's anisotropy. We have emphasized the substantial gain in computational cost compared with using an isotropically adapted mesh or even with a uniform mesh. When combined with stabilized  $P_1$  finite element method, a recovery of second order convergence of the numerical scheme is obtained. However, using a time-step that is generally set to a constant value or adaptively modified to satisfy stability conditions induces considerable CPU time usage. Indeed, the choice of the time-step sizes is highly dependent on the nature and characteristics of the problem being solved. We recall that one of the objectives of the present work is to enhance the efficiency of computations. For that purpose, we intend to resort to a time adaptive algorithm that accelerates computations while maintaining a good level of accuracy.

This chapter is devoted to the derivation and implementation of an appropriate time control algorithm. It emphasizes the importance of time adaptation in increasing the efficiency of the simulations. We start by briefly presenting an overview on existing time-adaptive schemes. Then we present the algorithm that is developed in this thesis. The analysis departs from the existing methodology of adaptation based on an explicit treatment of time advancing and their corresponding stability conditions. The problematic related to this choice of time incrementation are then addressed pointing out the necessity for a different strategy of evolution. To deal with this issue, we move on next to discuss several possible choices of time adaptation. We elaborate on their advantages and disadvantages. We then derive the algorithm that we adopt in the anisotropic adaptation framework and clarify its features in simulating complex phenomena.

## 4.1 STATE OF THE ART

A large variety of physical phenomena are characterized by their multiscale features as they involve the interaction of several processes evolving simultaneously at different temporal and spatial scales. The simulation of industrial and engineering problems usually deals with large-scale domains and should be run over a long period of time. Nevertheless, mimicking the behavior of physical phenomena through numerical simulations raises some challenges regarding the computational cost required to account for the various physical parameters and scales of the problem. Note that the study should be performed with a certain level of accuracy. The use of very small time-steps will certainly permit a better capture of the physical patterns disclosed by real and complex phenomena. However, such a choice of time-steps necessitates important computational costs and execution times. Therefore in practice, if no adaptation is applied, a compromise between computational efficiency and accuracy should be sought.

Temporal discretization can be done using different approaches of which we recall the explicit (*e.g.* explicit Euler or Runge Kutta), and the implicit (*e.g.* Euler backward, Crank Nicolson and fractional time-steps) schemes. The resolution methods relying on an explicit time integration scheme require a CFL condition that determines the range of the time-steps. This condition guarantees the solution's stability if the time-step is smaller than the time required for a compression wave to traverse the smallest finite element. The advantage of these methods resides in the fact that the resolution of the equation is immediate, i.e. the solution at time  $t^{n+1}$  is directly obtained from the given one at time  $t^n$  which is not the case for implicit schemes. On the other hand, the implicit resolution methods are unconditionally stable but they are not straightforward: they require the resolution of a system at each time-step. Note that although these schemes are unconditionally stable, a very large time-step can yield an inaccurate solution.

### 4.1.1 Stability constraints on explicit advancing

When employing an explicit time discretization of the fluid flows and heat transfer equations, stability constraints on the choice of time-step sizes are to be respected [Johnston 04]. The restrictions imposed by viscous terms stability conditions are defined by:

$$\Delta t_v \leq \frac{1}{\mu} \left( \frac{h_{\min}}{2} \right)^d, \quad (4.1)$$

where  $\mu$  refers to the kinematic viscosity of the fluid involved in the simulation,  $h_{\min}$  represents the smallest edge in the mesh and  $d$  the space dimension. Moreover, in order to allow a good capture of the different characteristics of the flow, the choice of time-steps should not exceed a Courant-Friedrichs-Levy (CFL) type limit for controlling the

convection terms. This bounding on the time-steps for incompressible flows reads as:

$$\Delta t_c \leq \frac{h_{\min}}{\|\mathbf{v}\|_{\infty}} \quad \text{with} \quad \|\mathbf{v}\|_{L_{\infty}} = \max_{x \in \Omega} \left\{ \left( \sum_{i=1}^d \mathbf{v}_i^2 \right)^{1/d} \right\}, \quad (4.2)$$

where  $\mathbf{v}$  stands for the velocity field and  $\mathbf{v}_i$  corresponds to its  $i^{\text{th}}$  component. Enforcing the two stability constraints for the convective and viscous terms, the time-step size is selected such that:

$$\Delta t \leq \min_{i \in \{1, N\}} \left( \min \left( \Delta t_v^i, \Delta t_c^i \right) \right), \quad (4.3)$$

where  $N$  denotes the number of nodes in the finite element mesh  $\mathcal{H}_h$ . It is important to mention that selecting  $h_{\min}$  as the smallest edge in the mesh which is not an optimal choice when this distance is taken in a different direction than the convective field. When anisotropic mesh adaptation is applied, the parameter  $h_{\min}$  can be determined from the metric tensors  $\mathcal{M}^i$ , taking into account the flow direction, as follows:

$$h_{\min} = \min_{i \in \{1, N\}} \frac{1}{\left( \frac{\mathbf{v}^t}{\|\mathbf{v}\|} \mathcal{M}^i \frac{\mathbf{v}}{\|\mathbf{v}\|} \right)^{1/2}} \quad (4.4)$$

Nevertheless, in the context of anisotropic mesh adaptation, the mesher tends to generate highly stretched elements in the orthogonal directions to the solutions' gradients. Consequently,  $h_{\min}$  takes very small values and in turn, the time-step size will be significantly reduced yielding a considerable CPU cost. Therefore the use of a stability driven time-step selection scheme constitutes a serious bottleneck for numerical simulations.

#### 4.1.2 Local time-stepping

A first attempt to reduce the computational cost due to time-stepping would be to consider local time-steps dictated by local stability conditions. In these methods, the computational domain is decomposed into several regions, called buffer zones, and the solution is evolved accordingly with different time-step sizes. Usually these methods are employed on structured meshes [Tam 03, Lörcher 07, Lörcher 08]. The extension to unstructured or anisotropically adapted meshes requires great attention. A good tuning between the solutions on different buffer zones must be performed. This can be done by deriving transfer coefficients between submeshes, that can be viewed as local boundary conditions. Moreover, a non heuristic parameter to delimit the buffer zones shall be defined. A third issue that needs to be carefully studied is the parallelization of this procedure. The method seems quite interesting however it is out of the scope of the current work as it is not possible to apply it with the current CIMLIB parallel library that we use.

In this work, an implicit time marching scheme is adopted. This type of temporal discretization allows to circumvent the restrictive stability criteria and to take more relaxed choices of time-step sizes. Yet if a good temporal accuracy is desired, prescribing

time-steps that respect condition (4.3) would be preferable. Nevertheless, this choice can drastically affect the computational time of the simulation. Therefore a compromise between accuracy and efficiency shall be considered. We present next an overview on the methods that we can find in the literature for adaptively evolving the solution in time. A brief summary about the algorithms will be provided next, more details are given in [Jannoun 11]. We note that we omit the time-marching schemes that are problem dependent as we are looking for an automatic algorithm that can be applied without modification on any problem at hand.

### 4.1.3 Overview on existing time-stepping algorithms

#### 4.1.3.1 Time-stepping based on local truncation error analysis

In 1999, Sloan et al. [Sloan 99, Kavetski 02] proposed an automatic consolidation algorithm that attempts to take a step-size in such a way that the error lies close to a certain predefined tolerance. After a first resolution of the system of equations at times  $t^n$  and  $t^{n+1}$ , a local truncation error is determined and the solution is linearly reconstructed along the time interval  $[t^n, t^{n+1}]$ . A relative error with respect to the reconstructed solution is computed and compared to a user prescribed tolerance. Then the time-step size is affected by a certain factor and the process is repeated until the relative error becomes smaller than the predefined tolerance. Although this time-adjustment algorithm is conceptually simple, it requires recomputing the problem's solution at each failure of the time-step size which is computationally expensive when simulating complex 3D phenomena.

In turn, Chen and Feng [Chen 04b] proposed in 2004 a time adaptation algorithm based on an a posteriori error analysis applied on the diffusion equation. The basic idea is to equally partition a tolerance value over time sub-intervals  $[t^n, t^{n+1}]$ . They start by computing the solution at time  $t^{n+1}$  then measure the temporal error induced on  $[t^n, t^{n+1}]$  and compare it to the equally distributed tolerance. They defined error ranges and divided the time-step size by a constant factor related to the error range.

The method of Berrone and Marro [Berrone 09] is added in 2009 to the above mentioned works on time-adaptivity derived from an a posteriori error indicators. The key point of their algorithm is to reduce the time-step size when the solution has a fast evolution and to enlarge it when the solution exhibits slow changes. The initial step of their method consists in tuning the time-step sizes so that the temporal error is bounded by an interval around a specific tolerance.

As there are no specific study on the optimal choices of the prescribed parameters (choice of the tolerance, factors for modifying the time-step sizes,  $\dots$ ), it is hard to employ this method in a general framework.

#### 4.1.3.2 Heuristic time-stepping schemes

In the heuristic time stepping methods [Celia 92, Rathfelder 94, Paniconi 91] the time-stepping adaptation for the  $(n + 1)^{th}$  time-step is based on the number of iterations  $N_{it}$  performed to obtain the solution of the nonlinear system at the previous time-step. Although this method is easy to implement and requires a low computational cost, it cannot be applied in a general framework for several reasons. First, the user-specified parameters depend on the problem at hand and require a priori knowledge of the solution's behavior. In practice, when the user does not know much about the profile of the solution the choice of the parameters will negatively affect the accuracy and efficiency of the solution.

Moreover, this method relates the time-stepping size to the number of iterations needed to solve the nonlinear system; in other words, the time-stepping choice is not directly dependent on the actual induced temporal error. On the other hand, this scheme is dependent on both the solver and the problem at hand so if any of the two is changed then one has to re-estimate the parameters all over again.

#### 4.1.3.3 The Residual-based time-stepping methods

Fidkowski and Luo proposed in 2011 an adjoint-weighted residual method for the computation of the adaptive time-steps [Fidkowski 11, Kast 13]. Their work relies on the study of primal and adjoint systems of equations. Through the use of an adjoint system they aim to account for the effects of error propagation. From a theoretical point of view this method has its pros as it provides a good level of accuracy however it requires considerable implementation challenges and computational expenses since it necessitates for each chosen time-step the solution of both the primal and the adjoint problems.

#### 4.1.3.4 Fictive residual methods

Fictive residual methods were introduced in 2009 by Georgiev et al. [Georgiev 09]. The time-step adaptive procedure is based on an error indicator computed as the difference between the approximated solution obtained using the Crank-Nicholson and the backward Euler temporal discretizations. This error is then compared with a user defined threshold and decisions on reducing or increasing the time-step sizes are deduced accordingly. Nevertheless, as the procedure computes the solution twice at every time increment, once using the Crank-Nicholson time discretization and once using the backward Euler discretization, it becomes computationally expensive.

#### 4.1.3.5 Fixed point adaptive staggering method

An adaptive staggering method was proposed by Zohdi in [Zohdi 07] whereby a classical staggering approach is used to solve each field equation at a time in a decoupled mode while activating only the primary field variable. After solving the equation, the

field is updated and the process is applied to the next field equation. Once the analysis is carried out for all the field equations, the time-step size is incremented. The originality proposed in [Zohdi 07] resides in the fixed point iterative staggering process in which the computations are repeated recursively until the system converges with respect to a tolerance value within a predefined number of iterations. The modification of the time-step size is controlled as a function of the spectral radius of the considered coupled system.

#### 4.1.3.6 Space and time residual based methods

Micheletti and Perotto proposed space and time residual-based error estimators [Micheletti 08c, Micheletti 08b]. Starting with initial guesses of the domain mesh and the time-step size, the solution is approximated on the corresponding space-time slab and a temporal error indicator is deduced. Depending on whether the estimated temporal error is below or above a certain threshold value, the algorithm either reduces the time-step size and repeats the computations or advances to the following time increment. In this work, an adhoc procedure for the recovery of the numerical solution's time derivative is employed followed by the calculation of the temporal residual as the jump in the solution over the space-time slab. The error estimation relies on residual mathematical analysis of the primal and dual problems. The approach has been derived separately on the convection-diffusion-reaction and the Navier-Stokes equations. The derivation of a formulation for the coupled heat transfer and fluid flow problems is a difficult task.

In this chapter, we adopt a different route for time-marching that we build in such a way to complement the anisotropic mesh adaptation described in chapter 2. The method that we develop is simple, easy to implement, automatic, independent of the problem at hand and does not require a user-defined tolerance. It consists in equidistributing the error in both space and time hence yielding the same level of accuracy.

The first idea that comes to mind is to extend the metric construction to take into account the time dimension. Hence, an optimal space-time mesh is built. This idea will be explored in the following section.

#### 4.1.4 Generating anisotropic space-time meshes

We consider a finite element discretization  $\mathcal{H}_h^n$  of the computational domain  $\Omega \times [t^n, t^{n+1}]$  made up of pentatope elements (4-simplex elements). We want to determine the optimal spatio-temporal mesh  $\widetilde{\mathcal{H}}_h^n$ . For the clarity of illustration, we denote by  $X_n^i$  the spatial node  $X^i$  at time  $t^n$ . The problem of building the space-time mesh is cast as a four-dimensional constrained optimization problem of finding the metric tensor that minimizes the total interpolation error in space and time under the constraint of a fixed number of nodes. The existing method of metric construction can be leveraged and needs only minor modifications to accommodate for the additional direction. Looking toward that end, the length distribution tensor  $\mathbb{X}^i$  at each node  $X_n^i$  will account

for an additional edge  $\mathbf{X}^{in+1}$  connected to  $X_n^i$ :

$$\mathbb{X}^i = \frac{1}{|\Gamma(i)| + 1} \left( \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} + s_{in+1}^2 \mathbf{X}^{in+1} \otimes \mathbf{X}^{in+1} \right) \quad (4.5)$$

where  $\Gamma(i)$  represents the number of nodes connected to  $X^i$  and  $s_{ij}$  the stretching factor of edge  $\mathbf{X}^{ij}$  in its own direction. Figure 4.1 shows a schematic interpretation of the 4D mesh. It is important to mention that in this case, the coordinate vector  $X_n^i$  contains an additional component referring to the time-dimension.

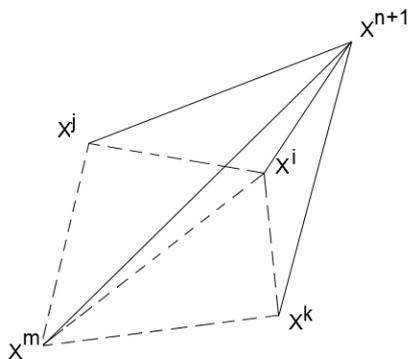


Figure 4.1 – Sketch of the edges connected to node  $X^i$  in a 4D mesh.

The stretching factors are determined based on an edge-based error estimation and the equi-distribution process. The associated metric tensor will be deduced as in chapter 2:

$$\mathcal{M}^i = \frac{1}{d+1} \left( \mathbb{X}^i \right)^{-1} \quad (4.6)$$

with  $d$  being the space dimension. The constructed metric will then be given to the MTC mesher which in turn will generate the corresponding optimal mesh. We recall that the MTC mesher's capability to generate 4D meshes has been validated [Gruau 05]. Therefore, this method seems quite promising, however it will not be pursued further and validated in this thesis for several reasons. First of all, adopting this adaptation technique requires the development of a tool that transforms the results computed on the 4D mesh into a set of data that can be visualized and physically interpreted. Second, it necessitates the development and implementation, in the CimLib library, of space-time stabilized finite element methods. Moreover, this type of meshes and solvers are not bearable by the available parallel toolbox. Consequently, using this adaptation scheme, the numerical simulations that we will be able to consider will be limited in complexity which is a major drawback of this method. However, preparing the ground for this technique and its validation constitutes a subject for perspective development.

Therefore in this work, the analysis is limited to considering that the time dimension is orthogonal to the space dimension. At this level, two possibilities for treating the space-time mesh optimization problem arise. The first one involves constructing a space-time metric while imposing the orthogonality assumption and adopting space-

time finite element schemes [Tezduyar 06] carried out for one space-time “slab” at a time. In this approach, the mesh is made up of right prisms whose basis are triangles in  $2D$  and tetrahedra in  $3D$ . The length distribution tensor at a spatial node  $X_n^i$  will be defined by:

$$\mathbb{X}^i = \begin{pmatrix} \mathbb{X}_s^i & 0 \\ 0 & (s_{in+1}\Delta t^n)^2 \end{pmatrix} \quad (4.7)$$

where  $\mathbb{X}_s^i$  stands for the length distribution tensor whose components contain information from the spatial edges only. Since the temporal edge is orthogonal to the spatial ones, the latter do not undertake any change in time hence the zero component below  $\mathbb{X}_s^i$ . Moreover, the temporal edge do not present any variation with respect to space hence the zero component to the right of  $\mathbb{X}_s^i$ . Based on the equi-distribution principle, the optimal metric would be the one that equi-distributes the error over the spatial and temporal edges in the mesh.

We recall that a metric tensor can be diagonalized into  $\mathcal{M} = \mathcal{R}\Lambda\mathcal{R}^t$  where  $\mathcal{R}$  is a rotation matrix whose columns are eigenvectors of  $\mathcal{M}$  and  $\Lambda$  is a diagonal matrix with diagonal elements  $\lambda_k$  the eigenvalues of  $\mathcal{M}$ . In an anisotropic mesh adaptation framework, the mesh sizes are determined from the eigenvalues of the metric  $h_k = \left(\frac{1}{\lambda_k}\right)^{\frac{1}{2}}$ . Applying the same principle in the space-time context, an additional eigenvector and its corresponding eigenvalue shall be accounted for. They reflect the direction corresponding to the temporal edge and its associated time-step size. To accommodate for the orthogonality assumption, the eigenvector will be set to a  $d + 1$  vector  $r_t = (0, \dots, 0, 1)$  where  $d$  is the space dimension. Consequently, the metric tensor will be defined by:

$$\mathcal{M}^i = \begin{pmatrix} \mathcal{M}_s^i & 0 \\ 0 & (s_{in+1}\Delta t^n)^{-2} \end{pmatrix} \quad (4.8)$$

where  $\mathcal{M}_s^i$  is the spatial metric tensor at node  $X_n^i$  defined in the previous chapter.

The metric tensor can then be provided to the MTC mesher in order to generate the optimal space-time mesh. It is important to point out that since we are accounting for each node twice, one at time  $t^n$  and its duplicate at time  $t^{n+1}$  then the prescribed number of nodes in the mesh should be increased. However, as the space and time dimensions are only connected through the equi-distributed error and in order to maintain a good level of accuracy, the temporal error should not exceed the spatial one. We recall that a second order finite element method is employed for evolving the solution in space whereas an implicit first order time-marching scheme is adopted. Therefore to enhance the accuracy of the space-time adaptation, the error control shall be driven from an analysis based on the spatial error to compute an equi-distributed error on the edges of the spatial mesh respecting the imposed fixed number of nodes in the mesh. Then the error in time is limited to respect the equi-distributed error bound. At this level, either a  $4D$  remeshing is performed by giving the metric tensor to the mesh generator, or a splitting of space and time dimensions is employed while respect-

ing the error equi-distribution property. This is the second approach for developing a space-time adaptation procedure. As it is well suited for capturing anisotropic features of physical problems, it will definitely improve the accuracy of unsteady simulations while preserving an affordable computational cost.

In [Belme 11], a space-time metric is constructed where the aim is to generate a spatial mesh that minimizes the space and time errors over a slab of time under the constraint of a fixed space-time complexity. This approach yields good accuracy control over an interval of time. However, an explicit time-marching scheme is adopted with a fixed time-step size or dynamically evolving time-steps subject to a Courant condition. The authors pointed out the limitation of the latter time evolution criterion. This is due to the fact that under a Courant condition, the time-step length depends on the smallest mesh size thus the applications using this adaptive process are restricted to ones having smooth solution fields.

The second approach, as discussed previously, consists in decoupling the space-time optimization problems, solve for the optimal metric in space and thus generate the well-adapted anisotropic spatial mesh. The construction relies on the edge-based error estimation and the equi-distribution principle. Once the spatial mesh has been obtained, we control the time-step size by applying the same principles for adaptation as in space and imposing the error equi-distribution property.

## 4.2 TIME ADAPTATION PROCEDURE

The basic idea of the time-adaptive algorithm that we develop in this chapter is to extend the spatial error analysis introduced in the previous chapter to the time dimension. The main objective is to produce a time-step which preserves the accuracy of the mesh adapted solution while accelerating the computations. Following the lines of the mesh adaptation technique, the time-step will be controlled by the interpolation error. Here we consider the time dimension as being analogous to a  $1D$  spatial problem and apply the whole theory exploited in chapter 2 in  $1D$ . Now that we have shown how to construct a metric field at each node  $X^i$  of the discretized domain that would lead to an adapted anisotropic mesh in  $\mathbb{R}^d$ , we move on to use the information from this construction in order to build an adaptive time marching technique. The latter should at the same time provide accurate estimations and not require a large computational cost.

Based on the information given by the derived error estimator in space and the history of the solutions, the algorithm should automatically calculate an appropriate time-step for the following computations regardless of the properties of the problem at hand. The time adaptation study is very similar to that of the anisotropic mesh adaptation except that the analysis is performed in a one dimensional context where the only studied variable being the time.

We consider a scalar field  $u \in \mathbf{C}^0(\bar{\Omega}) = \mathcal{V}$  and  $\mathcal{V}_h$  a simple  $P_1$  finite element approximation space:

$$\mathcal{V}_h = \{w_h \in \mathbf{C}^0(\bar{\Omega}), w_h|_K \in P_1(K), K \in \mathcal{K}\} \quad (4.9)$$

where  $K$  is a simplex (segment, triangle, tetrahedron,  $\dots$ ) in the mesh  $\mathcal{H}_h$ . We define

$$\mathbf{X} = \{X^i \in \mathbb{R}^d, i = 1, \dots, N\}$$

as the set of spatial nodes and

$$\mathcal{T} = \{t^0, \dots, t^{n-1}, t^n, t^{n+1}, \dots\},$$

as the set of temporal nodes in the time interval  $[0, T]$ . We let  $t^{nk} = |t^n - t^k|; n, k \in \mathcal{T}$  be a temporal edge with length  $\Delta t^n = t^{n,n+1}$  as shown in figure 4.2. It is important to mention that the number of temporal nodes, and hence time edges, is not known a priori as it depends on the behavior of the solution along  $[0, T]$ .

The space-time adaptive method can be split into two sub-optimization problems whereby one intends to minimize the global interpolation error in both space and time under the constraint of a fixed number of nodes. In the first sub-problem, we seek a metric field defined at the nodes of the mesh and yields an optimal well-adapted spatial mesh. The second sub-problem aims at finding the global time-step size that minimizes the temporal interpolation error under the constraint of maintaining the temporal error bounded by the spatial equi-distributed error. In the previous chapter, we have emphasized how to derive the metric tensor that results in the optimal anisotropic mesh adaptation. In this section, we intend to explain how to derive the optimal time-step sizes.

Assume that the solution  $u$  is already computed on the whole domain up to time  $t^n$ . The aim is to determine an appropriate time-step  $\Delta t^n$ .

#### 4.2.1 Edge-based temporal error estimation

Without loss of generality, the analysis will be carried on an arbitrary spatial node  $X^i$ . Note that at a spatial node  $X^i$ , we only have one time edge to be determined ( $t^n t^{n+1}$ ).

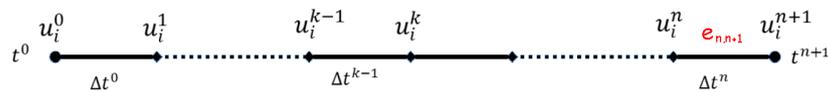


Figure 4.2 – Temporal discretization at the spatial node  $X^i$ .

Define  $\{\tau_{n,n+1}\}$  to be the temporal edge scaling (stretching) factor such that:

$$\begin{aligned} \tilde{e}_{n+1,n} &= \tau_{n+1,n}^2 e_{n+1,n} \\ |\tilde{t}^{n+1,n}| &= \tau_{n+1,n} |t^{n+1,n}|, \end{aligned} \quad (4.10)$$

where  $e_{n+1,n}$  is an approximation of the interpolation error known as the edge based error (more details are defined later),  $\tilde{e}$  and  $|\tilde{t}|$  are respectively the target error at the temporal edge  $t^n t^{n+1}$  and its associated edge length.

Let  $u_i^{n-1}$ ,  $u_i^n$  and  $u_i^{n+1}$  be the solutions at node  $X^i$  and times  $n-1$ ,  $n$ , and  $n+1$ , respectively. We introduce the time-step distribution function  $\mathcal{L}_i^n$  at the spatial node  $X^i$  and time  $t^n$ , which is equivalent to the length distribution tensor in space. As we are evolving the solution with a fixed time-step, the length distribution function reduces to a scalar value that is the same over all spatial nodes.

$$\mathcal{L}^n = \frac{1}{2} \left( (\Delta t^n)^2 + (\Delta t^{n-1})^2 \right). \quad (4.11)$$

Then applying the second order recovery gradient in  $1D$ , we get:

$$\dot{u}^n = (\mathcal{L}^n)^{-1} U_i^n, \quad (4.12)$$

where,

$$U_i^n = \frac{1}{2} \left( u_i^{n,n+1} \Delta t^n + u_i^{n,n-1} \Delta t^{n-1} \right). \quad (4.13)$$

Substituting equations (4.11) and (4.13) in (4.12), yields the recovered derivative:

$$\dot{u}^n = \frac{u_i^{n,n+1} \Delta t^n + u_i^{n,n-1} \Delta t^{n-1}}{(\Delta t^n)^2 + (\Delta t^{n-1})^2}. \quad (4.14)$$

Hence the change in the derivative along the temporal edge  $[t^{n-1}, t^n]$  is determined by:

$$\dot{u}_i^{n,n-1} = \dot{u}_i^n - \dot{u}_i^{n-1} \quad (4.15)$$

and edge-based error estimation is given by:

$$e_i^{n,n-1} = \dot{u}_i^{n,n-1} \Delta t^{n-1}. \quad (4.16)$$

As discussed in the previous chapter, the error estimates change quadratically as a response to changing or stretching the edge by a factor  $\tau_i^{n-1}$ . Therefore:

$$\tilde{e}_i^{n,n-1} = \tau_i^{n-1} \dot{u}_i^{n,n-1} \Delta t^{n-1}, \quad (4.17)$$

where  $\dot{u}_i^{n,n-1} = \dot{u}_i^n - \dot{u}_i^{n-1}$ .

At this level, the analysis will deviate from the one adopted in the derivation of the anisotropic mesh adaptation. The direct extension would have been to fix a total number of time-steps and to determine the respective time-step sizes over the simulation based on an error equi-distribution on the subintervals. However as the solution behavior is not known apriori for the full time interval, the error estimates on the temporal edges cannot be predicted. Therefore the time-adaptive approach cannot be completed in this way. We will instead continue the analysis in a space-time framework, that is

we equi-distribute the error in space and time. To do so, we modify the time-step size in a way to prevent the temporal error from exceeding the equi-distributed error in space at time  $t^n$  computed according to equation 2.113. Hence, based on the space-time equi-distribution error argument, we write

$$\tilde{e}_i^{n,n-1} = e(N, t^n), \quad (4.18)$$

where  $e(N, t^n)$  is the equi-distributed error in space for a total number  $N$  of nodes. Consequently, the stretching factor of the time-step size, driven by the assumption of the quadratic change in the error as a result of scaling the temporal edge, is given by:

$$\tau_i^{n-1} = \left( \frac{e(N, t^n)}{e_i^{n,n-1}} \right)^{\frac{1}{2}}, \quad (4.19)$$

and the optimal time-step is determined by:

$$\widetilde{\Delta t^n} = \min_i \tau_i^{n-1} \Delta t^n. \quad (4.20)$$

However, if we consider this formula closely we find out that it requires the solution at time  $t^{n+1}$  which is not yet computed. Therefore instead of computing the optimal time-step  $\widetilde{\Delta t^n}$  we calculate  $\widetilde{\Delta t^{n-2}}$  and we let

$$\Delta t_n = \widetilde{\Delta t^{n-2}}. \quad (4.21)$$

It is important to note that when a frequency  $f$  of spatial adaptation is used, the temporal stretching factors will be modified to account for a new equi-distribution. The new stretching factor is then defined by:

$$\tau_i^{n-1} = \left( \frac{e(N, t^n) / f}{e_i^{n,n-1}} \right)^{\frac{1}{2}}. \quad (4.22)$$

In other words, the equi-distributed error  $e(N, t^n)$  estimated at time  $t^n$  to generate the anisotropically adapted mesh that will last  $f$  time increments will be spread over the time subinterval  $[t^n, t^{n+f}]$ . So that the total temporal error over  $[t^n, t^{n+f}]$  would not exceed  $e(N, t^n)$ . The time-step adaptation is employed at every time increment to maintain a good level of accuracy. This is possible and computationally inexpensive as the algorithm for time-adaptation consists of simple arithmetic operations that are affordable and yield a high level of accuracy. We note that throughout this thesis work an implicit time marching scheme is employed. Nevertheless, adapting the mesh every  $f$  time-steps using the developed space and time adaptive algorithms is not an optimal choice when a second order time discretization scheme is used. We present in chapter 6 a new method that proceeds by slabs of time conserving second order temporal accuracy.

### 4.2.2 General space-time adaptive algorithm

The steps for the space and time adaptation are summarized in algorithm 3. The decoupled space-time mesh optimization problem starts by evaluating the edge-based spatial error estimates. To do that, a length distribution tensor is computed and a gradient reconstruction at the mesh nodes is performed. After determining the error estimates on the edges in the mesh, an equi-distributed error is predicted. If the mesh adaptation process is activated, the stretching factors for the edges and thus the length distribution tensors and optimal metrics are deduced. Consequently, these metrics are provided to the MTC mesher that generates the anisotropically adapted mesh. The solution fields are interpolated into the new mesh. Afterward, derivative reconstruction on mesh nodes and error estimates on the edges are determined yielding a new definition of the optimal time-step size for the following time increment. The process is repeated iteratively until the end of the simulation.

---

#### Algorithm 3: Space and time adaptive algorithm

---

**Input:** The initial mesh  $\mathcal{H}_h^0$ , problem's parameters  $u_h^0$ , and time-step size  $\widetilde{\Delta t} = \Delta t_0$ .

**Output:** The numerical solution at time  $t^n$ , the optimal mesh and time-step size.

**begin**

```

  n = 1, t = 0 // Initialization
  while tn-1 < T do // Time loop
    Prescribe the number of nodes in the mesh N
    Increment the time tn := tn-1 + Δtn-1
    Compute uhn on mesh Hhn-1
    for each node Xi do
      Compute Xi // The length distribution tensor
      Compute Gi // Recovery gradient operator
    for each edge Xij do
      Compute Gij // Change of the gradient over the edge
      Compute eij // Edge based error
      Compute e(N, tn) // The equidistributed error
    for each node Xi do
      Compute ũin-1, n-2 using eqs.(4.2.1) and (4.15) // Temporal derivative
      Compute ẽin-1, n-2 following eq.(4.16) // Temporal error
      Compute τin-2 using eq.(4.22) // Temporal stretching factor
      Compute Δtn-2 using eq.(4.20) // The optimal Δtn-2
      Set Δtn = Δtn-2 // Set the following time-step
  
```

---

When the target field  $u$  for time adaptation is a vector, the procedure is applied to every component of this vector field and then the minimum prescribed time-step size is selected for the following time-increment. More details about multi-field adaptation will be provided in chapter 5.

It is important to mention that the desired mesh density can be interactively updated during the simulation time through an implemented user-interface. Moreover, an anisotropic adaptation based on a desired tolerance instead of a fixed number of nodes can be generated following exactly the same algorithm while replacing the equidistributed error by the threshold value. Clearly, this will be at the cost of losing the control on the mesh density. A combination between a control based on a tolerance value and a maximum mesh density can be obtained by taking the maximum between the prescribed error and the homogenous error determined with respect to the number of nodes  $N$ .

### 4.3 NUMERICAL VALIDATION

The main objective of this work is to provide a space and time adaptive method that can be used to simulate diverse industrial applications. For practical reasons, it is highly desirable to find the most accurate solution using the available computational resources within a reasonable time. In this section, we assess the performance of the newly developed space and time adaptive algorithms on several time dependent problems. We first assess the order of convergence of the couple mesh/time-step size on an analytical function. Then we move on to analyze the dynamic properties of the space-time adaptive procedure. We start by investigating the impact of anisotropic mesh adaptation and time-stepping control on the accuracy and efficiency of computations. We move on next to test the capability of the method to well capture fluid flow problems at high Reynolds numbers. Finally we evaluate the performance of the developed algorithms on coupled fluid flows and heat transfer problems.

#### 4.3.1 Convergence analysis on an analytical test case

The objective of this numerical example is to illustrate how the anisotropic mesh adaptation and time-step control enable an accurate capture of the displacement of a function's anisotropic features over a time interval. The analytical function that we consider was first presented in [Coupez 11] and was taken to be constant in time. In the previous chapter, we have investigated the developed metric construction technique on this function and it showed good performance in accurately capturing the sharp solution gradient over the domain. In this chapter, we let the analytical function vary both in space and time yielding a good test for the developed error estimators, meshing techniques and time adaptivity. It is defined on the domain  $[0, 1]^d$  and is being evolved over the time interval  $[0, 1]$  as follows:

$$\begin{aligned}
a(\mathbf{x}) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X}\|\right)\right) \\
b(\mathbf{x}, t) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X} - t\mathbf{C}\|\right)\right) \\
\mathbf{C} &= \begin{pmatrix} 1 \\ \cdot \\ 1 \end{pmatrix} \\
u(\mathbf{x}, t) &= a \circ a(\mathbf{x}) + b \circ b(\mathbf{x}, t)
\end{aligned} \tag{4.23}$$

The parameter  $E$  varies from 1 to 32; the larger  $E$  is, the sharper the function's gradient, favoring anisotropic meshing. Despite the simplicity of this function, it is characterized by the displacement of the high gradients over time. The complexity of this example resides in the expansion and interference of the evolving circular gradients forcing isotropic meshing at the intersection zones.

We start our analysis by validating the proposed anisotropic mesh and time adaptation algorithms. In figure 4.3 the analytical function is depicted at three different times, with  $E = 16$ , on the anisotropically adapted meshes that correspond to an equivalent number of nodes (around 20,000 nodes) obtained at different time instances. The mesh has been adapted on the  $L_1$  norm of the interpolation error every 5 time increments. We can detect how the elements in the mesh are well oriented and stretched along the tangential direction to allow a good capture of the function's circular gradient. The nodes are being automatically distributed with a higher density in the regions surrounding high gradients. The reported zooms on the mesh reflect how well the elements are elongated to provide a better capture of the solution's sharp gradient. At the intersection of two sharp layers (radial gradients), the mesh is isotropic giving equal importance to the different directions and yielding equal capturing of the two sharp layers. This observation is a direct result of the error equi-distribution principle.

We vary the number of nodes in the mesh between 4,000 and 80,000 and study the interpolation error's temporal order of convergence. Figure 4.4 reports the temporal convergence history. The first order convergence in the range [4,000 – 16,000] reflects that the number of nodes is not high enough to allow a correct capture of the function's characteristic steep variations. As of 16,000 nodes, we can obviously detect nearly second order accuracy in time which is in accordance with the construction of the time-adaptive algorithm. In figure 4.5 we can identify that the number of time-steps generated by the adaptive algorithm varies linearly with respect to the imposed number of nodes in the mesh. Therefore by controlling the temporal error under the constraint of a fixed number of nodes in the mesh, a second order temporal convergence is implied.

Figure 4.6 gathers the time-step evolutions corresponding to three configurations of mesh adaptation: adapting on the  $L_1$ ,  $L_2$  and  $L_\infty$  norms of the interpolation error. The computations were performed on a mesh of nearly 20,000 nodes that is being anisotrop-

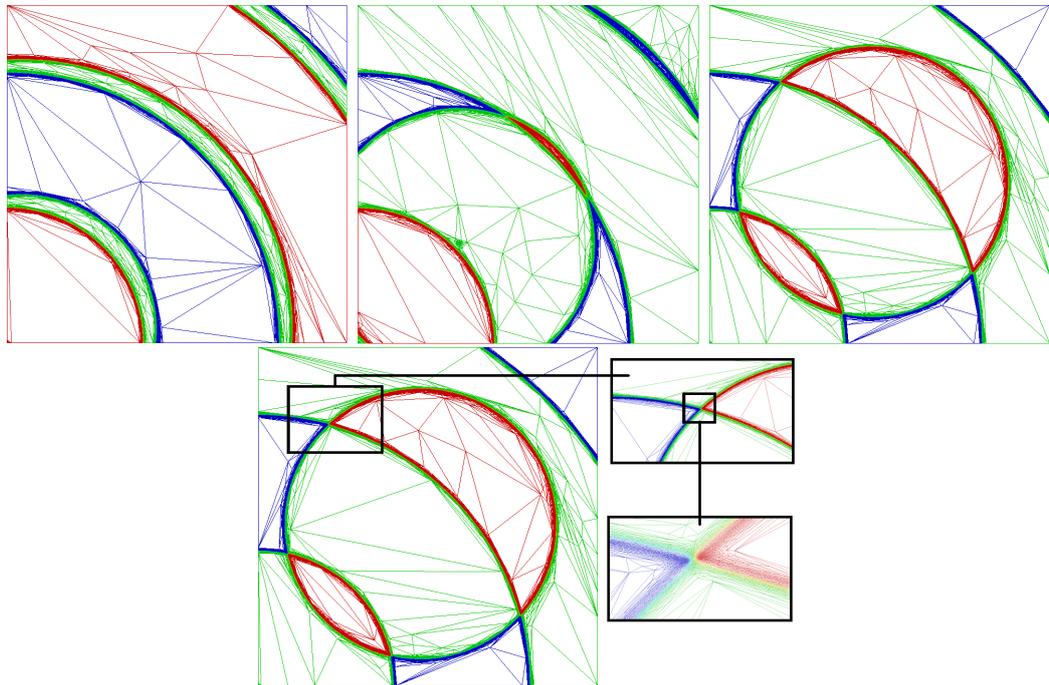


Figure 4.3 – Anisotropic mesh obtained at different time instances (top) and closeups around the radial steep gradient region (bottom).

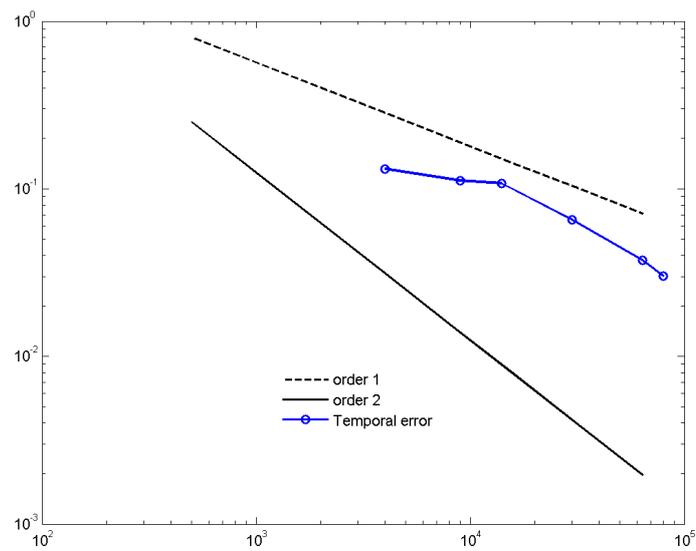


Figure 4.4 – Temporal convergence of the interpolation error with respect to the mesh complexity.

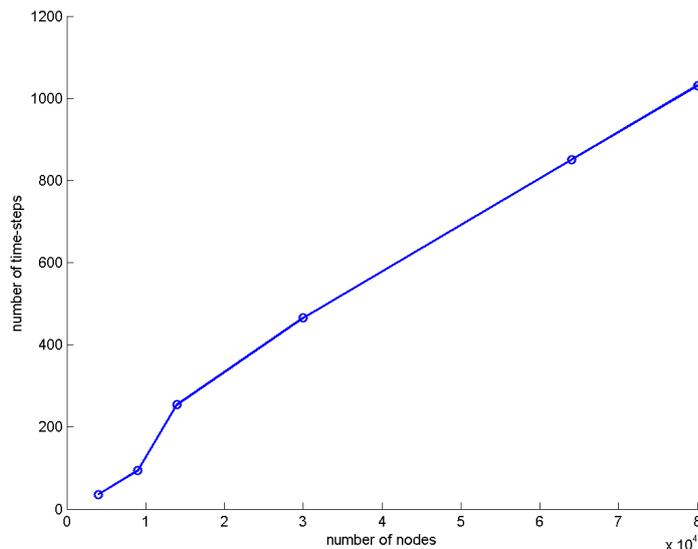


Figure 4.5 – Variations of the number of time-steps with respect to the imposed number of nodes in the mesh.

ically adapted every 5 time increments. We notice that adapting the mesh on the  $L_\infty$  norm of the error leads to smaller time-steps which goes in parallel with expectations as the temporal error is not allowed to grow faster than the spatial one. The history of the time-steps is characterized by a short period of abrupt increase, probably biased by the choice of the initial time-step that is taken to be  $5 \times 10^{-5}$ , allowing the emergence of the ring shaped gradient of the function into the domain. As the function evolves linearly with time for the different norms, all the profiles of time-stepping reach a phase of steady oscillation around a critical value that guarantees the equi-distribution of the error in space and time. Since the  $L_\infty$  norm is more restrictive than the other norms, the time-step will oscillate around a smaller value than the other norms.

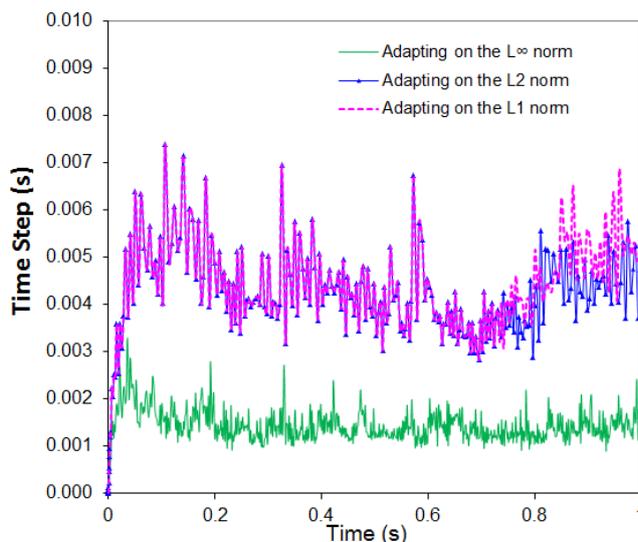


Figure 4.6 – Time-step evolution corresponding to adapting the mesh on the  $L_1$ ,  $L_2$  and  $L_\infty$  norms.

We compare in figure 4.6 and table 4.1 the time-step evolution and the CPU time obtained using the developed mesh adaptation algorithm and time-stepping technique.

The results are compared to the CPU time needed to evolve the function on anisotropic meshes with a fixed time-step size set to be equal to two times the smallest mesh size. This result validates the efficiency of the newly developed method in terms of computational time.

| Method                     | Number of nodes  | $N = 8,000$ | $N = 16,000$ | $N = 32,000$ | $N = 64,000$ |
|----------------------------|------------------|-------------|--------------|--------------|--------------|
|                            | Fixed $\Delta t$ |             | 37           | 54           | 65           |
| spatio-temporal adaptation |                  | 22          | 29           | 38           | 53           |

Table 4.1 – Needed CPU times (mins) for a fixed time-stepping and an adaptive control of time-step sizes by the proposed algorithm for different mesh complexities.

### 4.3.2 A two-dimensional analytical test case with sinusoidal evolution in time

The objective of this test case is to illustrate the behavior of the time-step algorithm relative to sinusoidal evolution of the function in time. We consider the same analytical function with a sharpness parameter equal to 16 and we let it vary sinusoidally in time as follows:

$$u(\mathbf{x}, t) = a \circ a(\mathbf{x}) + b \circ b\left(\mathbf{x}, \sin\left(t \frac{\pi}{2}\right)\right) \quad (4.24)$$

The rings emerging inside the domain move periodically back and forth over time. Figure 4.7 shows the time-step history obtained using the time-adaptive algorithm. The mesh made of around 20,000 nodes is being adapted every 5 time increments. We notice that the profile of time-stepping presents a periodic evolution over time. A phase of slow steady time-steps reflects the expansion of the rings inside the domain. A deceleration of the function follows and is accompanied by a quick increase of the time-step sizes. An abrupt decrease of the latter is observed once the function changes its direction of evolution. This succession of time-step behavior is periodically repeated in accordance with the sinusoidal displacement of the function in time.

### 4.3.3 Applications on 2D fluid flow problems

We move on now to assess the developed spatio-temporal adaptation on fluid flow problems with varying Reynolds numbers. These results were reported in a recent publication on the subject [Coupez 13b].

#### 4.3.3.1 Driven flow cavity problem (2-D)

In this example, we assess the performance of the newly developed space and time adaptive algorithms on the classical 2-D lid-driven flow problem at low and high Reynolds numbers with a zero source term. This test has been addressed in the previous chapter to validate the anisotropic mesh adaptation method. Now, we re-consider the problem to evaluate the accuracy and efficiency of the time-adaptive technique. We consider a fluid with density  $\rho = 1\text{kg/m}^3$ . We vary the kinematic fluid viscosity in

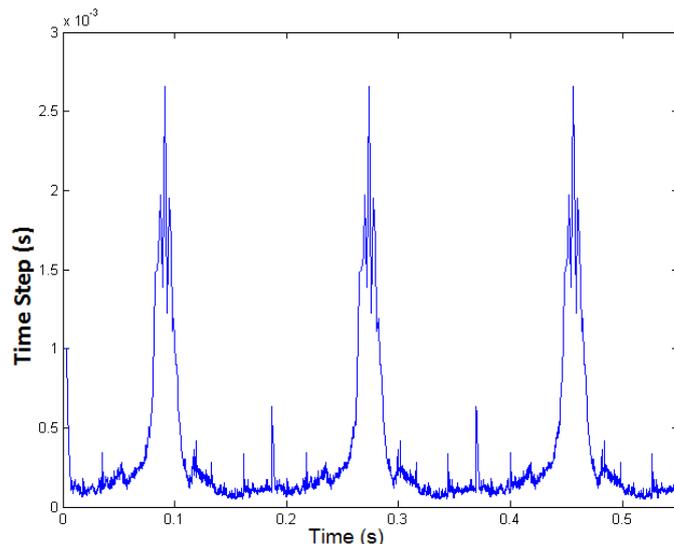


Figure 4.7 – History of the time-steps (s) obtained with the developed algorithm on problem (4.24) .

order to generate and compare the solutions for Reynolds numbers of 1,000, 5,000, 10,000 and 20,000; we fix the number of nodes  $N$  to 15,000. Note that the initial mesh used before any adaptation is an unstructured isotropic mesh made up of 20,000 nodes. We start all the numerical simulations using a time-step equal to 0.01s. We apply the space-time adaptive algorithm every 5 time-increments while adapting on the norm and direction of the velocity field. We assume that the steady state is reached when the normalized velocity deviations within one step are lower than a chosen tolerance of  $10^{-6}$ . The results on the converged meshes ( $\sim 15,000$  nodes) for the different Reynolds numbers are shown in Figure 4.8. Note the concentration of the resolution not only along all the boundary layers but also at the detachment regions. This reflects well the anisotropy of the solution caused by the discontinuity of the boundary conditions and the nature of the flow. The elements at the central bulk of the cavity region around the primary vortex are mostly isotropic and increase in size as the Reynolds number increases. Again, this explains how, for a controlled number of nodes, the mesh is naturally and automatically coarsened in that region with the goal of reducing the mesh size around the secondary vortices in the bottom corners. This observation reflects that the space-time adaptive method tends to refine the mesh in the hierarchical importance of the solution's gradient. That is when new features (with high gradients) appear in the solution, the mesh is automatically coarsened in regions with lower gradient and refined near the newly emerging features yielding an optimal capture of the solution with the available degrees of freedom.

The velocity profiles for  $u_x$  and  $u_y$  along the lines  $x=0.5$  and  $y=0.5$  respectively are depicted in figures 4.9 and 4.10. We note that a time-averaged velocity profile is represented for  $Re = 10,000$ .

The main objective of these simulations is to compare the newly obtained results to very accurate reference solutions. For that purpose, we first plot the velocity profiles for  $V_x$  along  $x = 0.5$  and for  $V_y$  along  $y = 0.5$  at Reynolds 1000 and 5000. Figure

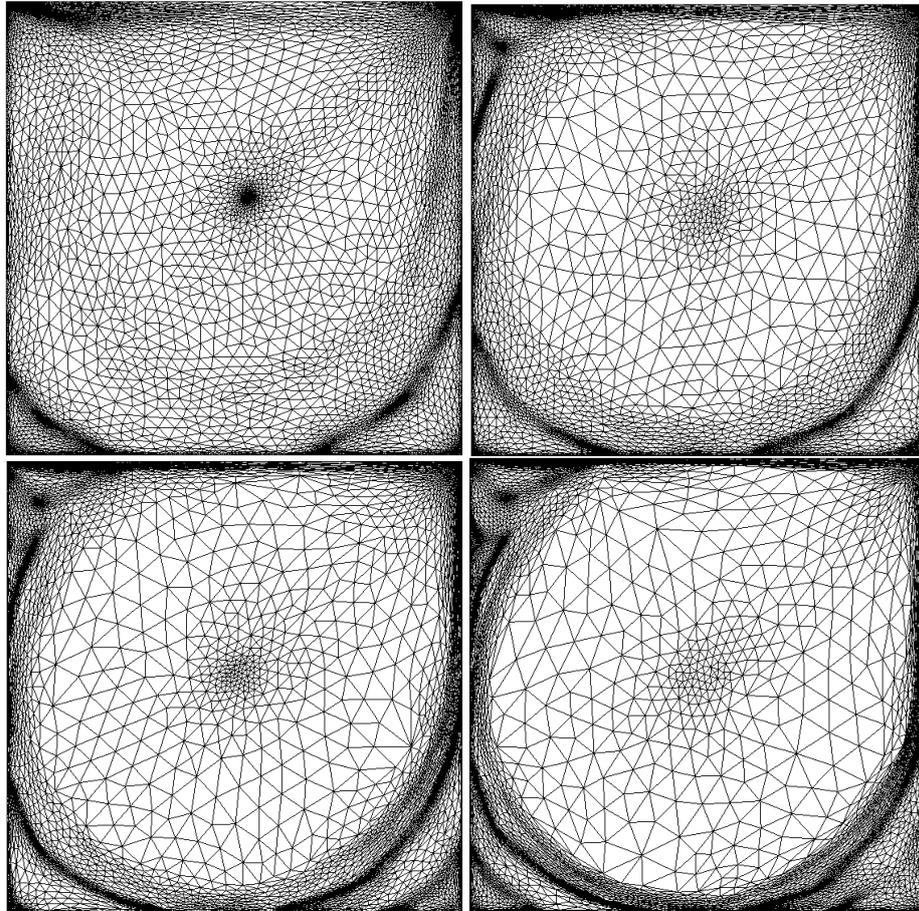


Figure 4.8 – Anisotropic meshes at Reynolds 1,000, 5,000, 10,000 and 20,000

4.9 shows respectively that all the results are in very good agreement with a reference solution computed on a  $600 \times 600 = 360,000$  points fixed mesh [Hachem 10b].

We present in table 4.2 the execution time and number of time-steps taken by each of the test cases using an adaptive method and a non-adaptive one. For the non-adaptive case, the computations were performed on a mesh having 100,000 nodes. Note that we needed 5,000 time-steps  $\Delta t = 0.1$  to reach the final time 1,000s. An extreme improvement in the CPU time needed to perform the computation of the numerical solution was observed. This reflects the high efficiency of the proposed time-adaptive algorithm.

| Reynolds Number | 1,000 | 5,000 | 10,000 | 20,000 |
|-----------------|-------|-------|--------|--------|
| Non-Adaptive    | 900   | 2233  | 4745   | 7820   |
| Adaptive        | 131   | 224   | 637    | 1365   |
| Ratio           | 6.87  | 9.97  | 7.44   | 5.72   |

Table 4.2 – CPU time (s) needed for computing the solution with non-adaptive and a space/time adaptive methods

Figure 4.11 shows that the time-adaptive algorithm meets the expected variations of the time-steps. The test included time-adaptation on a fixed mesh (of around 60,000

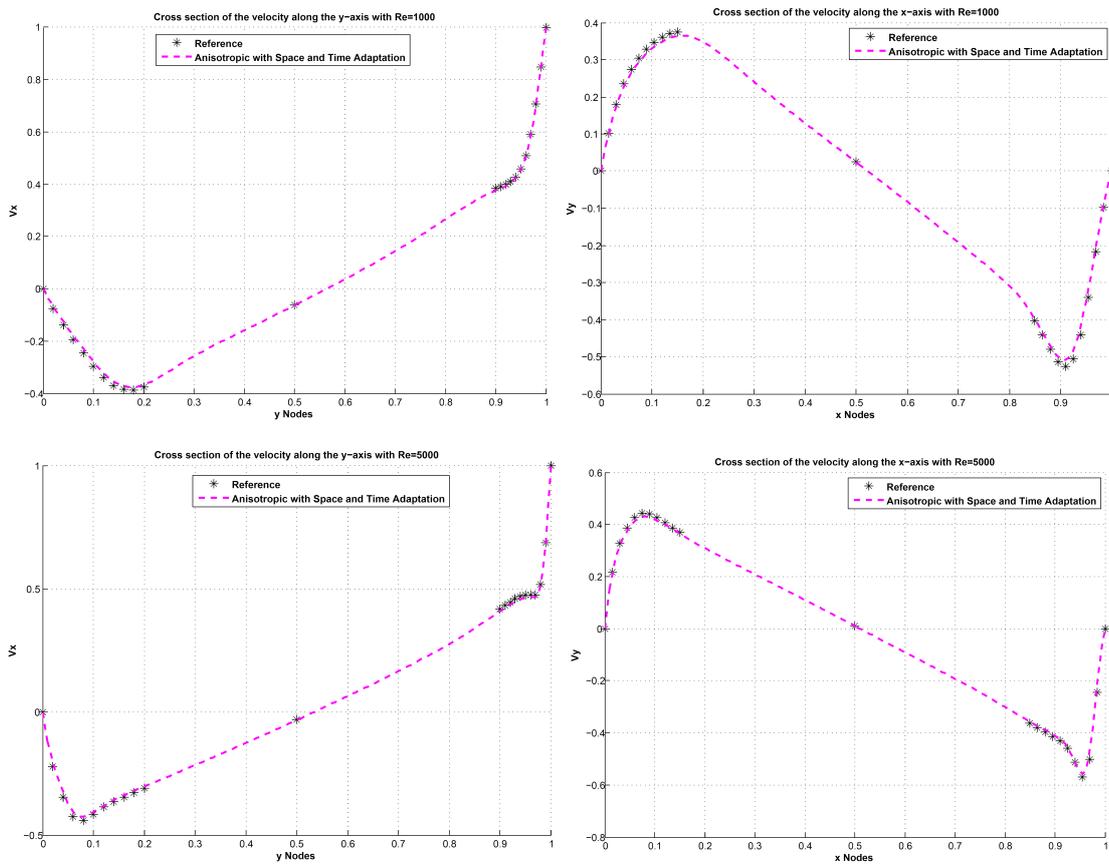


Figure 4.9 – Comparison of velocity profiles in the mid-planes for  $Re = 1,000$  (top) and  $Re = 5,000$  (bottom). Left: Velocity profiles for  $U_x$  along  $x = 0.5$ . Right: Velocity profiles for  $U_y$  along  $y = 0.5$ .

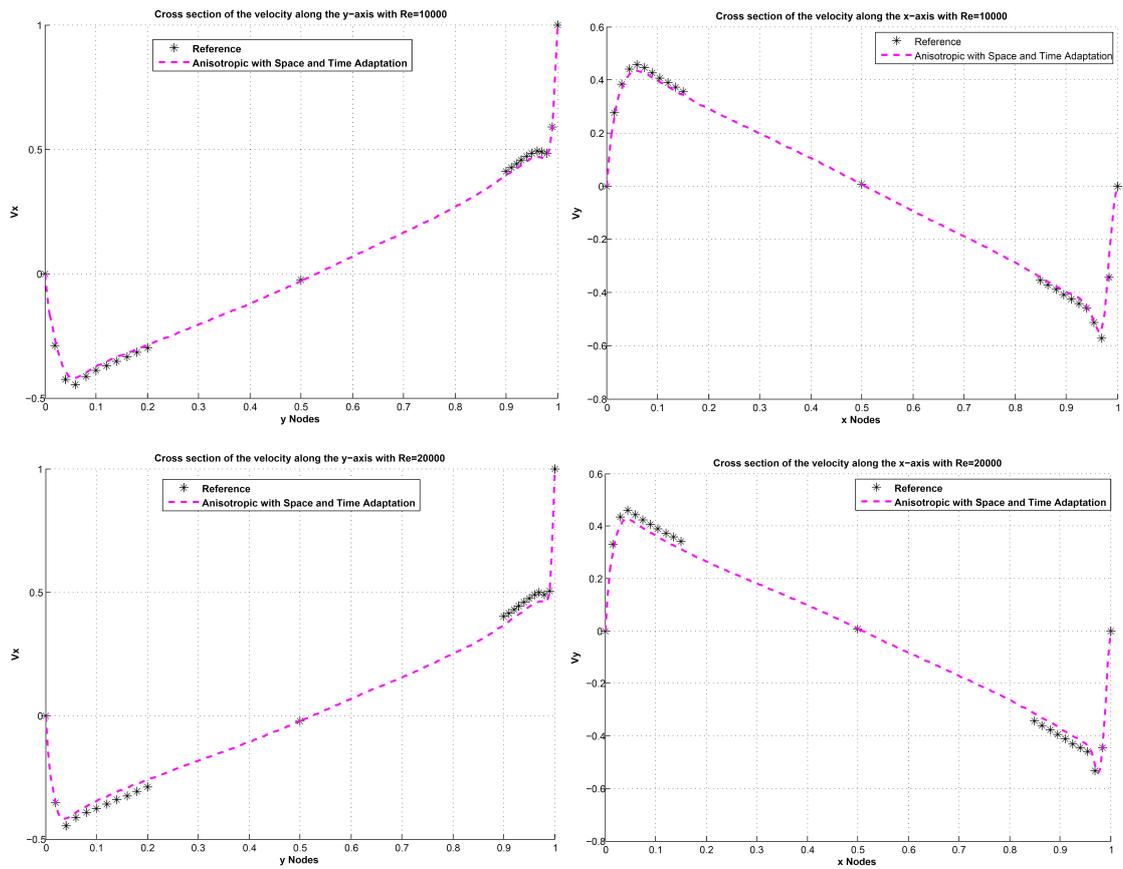


Figure 4.10 – Comparison of time-averaged velocity profiles in the mid-planes for  $Re = 10,000$  (top) and  $Re = 20,000$  (bottom). Left: Velocity profiles for  $U_x$  along  $x = 0.5$ . Right: Velocity profiles for  $U_y$  along  $y = 0.5$ .

nodes) and space and time-adaptation with a prescribed number of nodes equal to 15,000. Recall that for Reynolds numbers less than 5,000, the solution reaches a steady-state and this predicts high time-step sizes. On the other hand, as we increase the Reynolds numbers, the solution exhibits the emergence of secondary vortices of different sizes that develop inside the domain and the flow becomes unsteady and very chaotic. The time-step size is expected to behave accordingly and to oscillate all over the simulation time as the solution does not stabilize. Indeed, for low Reynolds numbers  $Re = 1,000$  and  $Re = 5,000$ , the solution stabilizes after a certain time and consequently, the time-steps increase considerably and then become almost constant once the solution stabilizes when the spatial error, on which the algorithm depends, becomes almost constant. On the other hand, for high Reynolds numbers  $Re = 10,000$  and  $Re = 20,000$ , the solution is unsteady and develops vortices of different sizes; in parallel, the time-steps are oscillatory and change in small amplitude (as shown in the plots of the solutions obtained on isotropic meshes with only time adaptation). We can clearly detect differences in the time-steps variations between running the simulation on a fixed mesh and anisotropically adapting the mesh. For laminar flows, both profiles reflect almost the same behavior, once the steady solution was reached, large time-steps are generated. The slight difference which can be viewed as a hesitation of the space-time adaptive algorithm to take large values of time-steps are due to the dynamic change of the mesh. As the mesh is modified toward optimality, a smaller time-step size is computed allowing a better capture of the solution. On the contrary, when the Reynolds number increases and the flow becomes more and more fluctuating, the sequence of generated time-steps presents faster evolution as compared with the one obtained on a fixed mesh. This result can be related to the higher accuracy provided by the anisotropic mesh adaptation algorithm allowing the creation of larger time-step sizes and hence faster computations. At  $Re = 10,000$  and  $Re = 20,000$ , the time-step evolution on a fixed mesh presents an oscillatory periodic behavior that keeps the computations within the same accuracy range. On the other hand, as the control of accuracy is provided by both space and time adaptation algorithms, the time-step evolution needs not maintain a periodic oscillatory profile.

Moreover, all the main directional features characterizing the velocity inside the lid-driven cavity are detected and well captured by the anisotropic error estimator. The mesh elements are highly stretched along the direction of the layers, at the detachments regions and around all the secondary vortices that are being developed. Once again, the developed incompressible Navier-Stokes VMS solver shows to be efficient and robust at high Reynolds numbers using highly stretched elements and adaptive time-marching approach.

In order to assess the capability of producing a more accurate solution as we increase the number of nodes, we solve the driven cavity problem with Reynolds number 1000 and we vary the allowed number of nodes: 5,000, 10,000, 20,000, and 40,000. It can be seen in figure 4.12 that as the number of nodes in the mesh increases, the solution becomes more accurate.

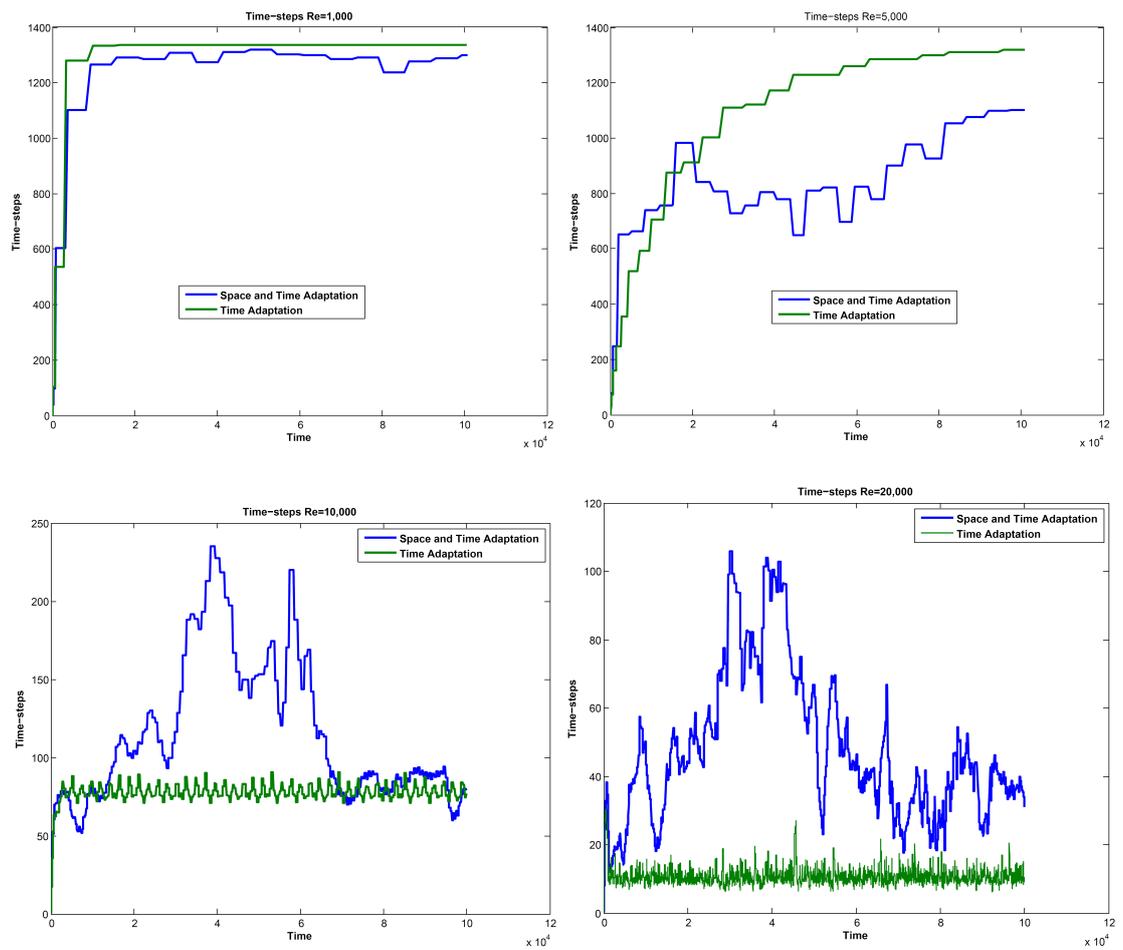


Figure 4.11 – Comparison of the time-step sizes' variations (s) for  $Re = 1,000$ ,  $Re = 5,000$ ,  $Re = 10,000$  and  $Re = 20,000$ .

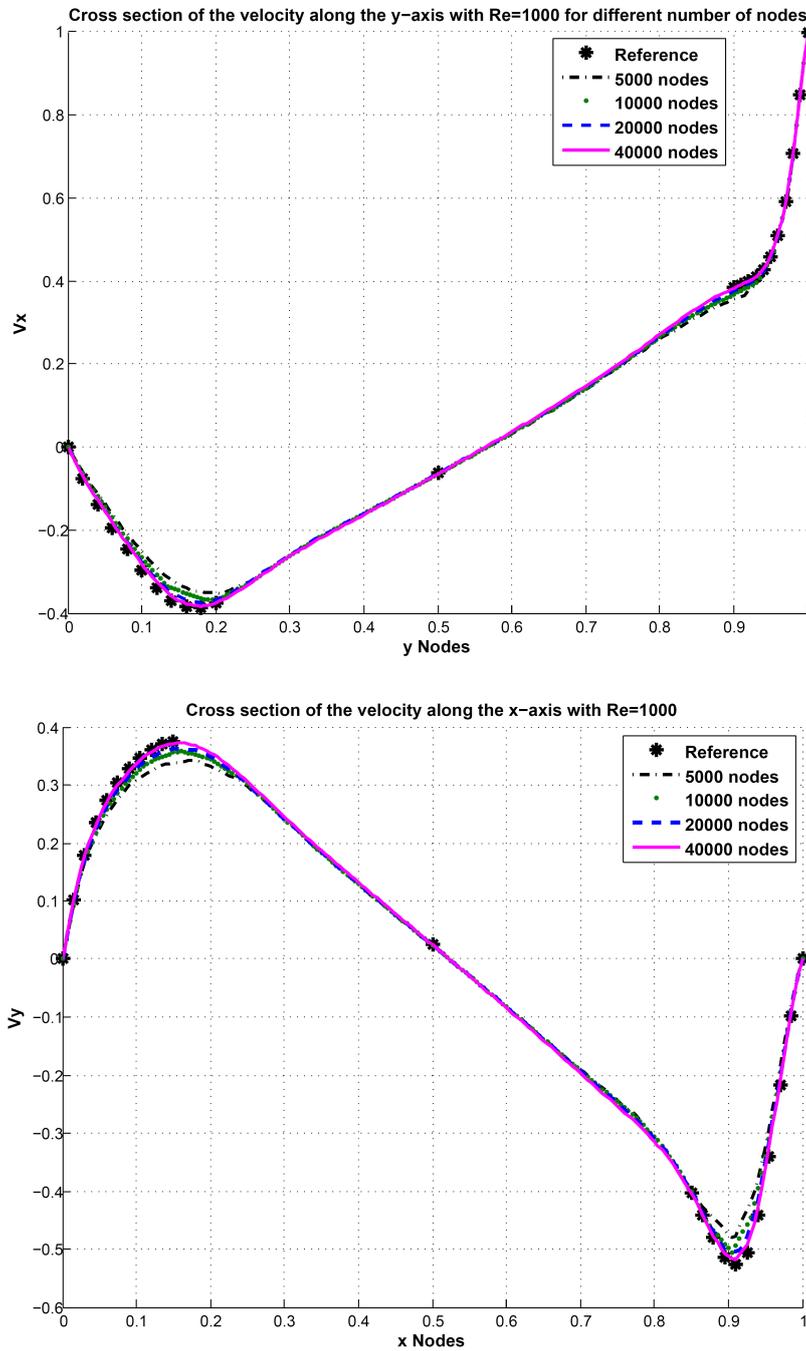


Figure 4.12 – Comparison of velocity profiles in the mid-planes for  $Re = 1,000$  computed using different number of nodes.

We have pushed the analysis further to evaluate the response of the time-adaptive algorithm to sudden changes in the setup of the problem. We consider the same flow at Reynolds number 5,000 and introduce an outlet of size 0.1 unit length at the bottom wall allowing the fluid to exit the cavity. This outlet is the 2D analogous to opening the door of an industrial furnace after a period of heating. The outlet is activated at time  $t = 4,000s$ , by that time, the flow would have settled inside the cavity. Figure 4.13 shows the abrupt change in the time-step evolution as a results of this perturbation. We notice that the algorithm automatically decreases the time-step size in order to better capture the changes in the physical phenomena inside the cavity.

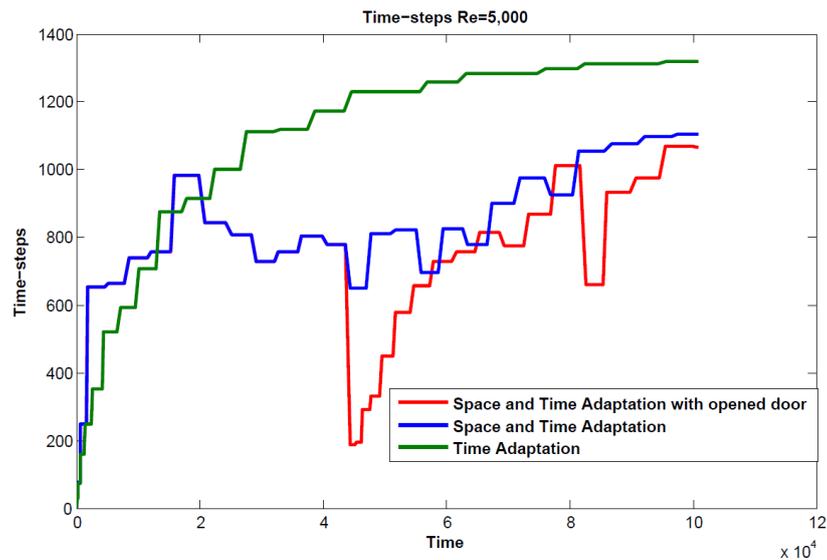


Figure 4.13 – Time-steps evolution (s) after introducing an outlet to the cavity at time  $t = 4,000$  seconds.

#### 4.3.4 Flow around a circular cylinder (2-D)

The following numerical test is another widely-studied problem, namely that of a channel flow with a cylindrical obstruction. This test case is a very popular benchmark problem. It was defined by M. Schäfer and S. Turek [Schäfer 96] and has been tested by many authors [Volker 04b, Volker 01, Frochte 09].

Important benchmark properties of the flows around bodies are the drag coefficients and the difference in the pressure between two points at the edge of the obstacle. To get proper values for these parameters one needs to have a high accuracy and very small time-steps.

We consider the incompressible Navier-Stokes equations with a zero source term. The setting of the problem is shown in figure 4.14. It consists of a rectangular channel (2.2x0.41)m with a circular obstruction of diameter 0.1m. Zero initial condition is considered inside the domain and a zero flow boundary condition is imposed on the circle

boundary. On the inflow and outflow boundaries the velocity is defined by:

$$u(t;0,y) = u(t;2.2,y) = 0.41^{-2} \sin\left(\frac{\pi t}{8}\right) (1.2y(0.41 - y), 0) \text{ m/s}$$

No-slip conditions are prescribed at the other boundaries.

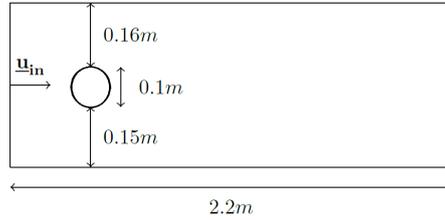


Figure 4.14 – Geometry of the problem adopted from [Schäfer 96].

For this problem, we set the kinematic viscosity  $\mu$  to  $10^{-3} \text{m}^2/\text{s}$  and the fluid density to  $\rho = 1 \text{kg}/\text{m}^3$  in order to obtain Reynolds numbers  $0 \leq \text{Re} \leq 100$ . This problem was well studied in [Volker 04b, Volker 01] and reference values for the characteristic properties of the flows were provided.

Our motivation behind this simulation is to show that the proposed algorithm can be used to “follow the physics” of a flow’s transition from a state of rest to a steady state solution. The mesh has been adapted on the velocity field at a frequency of 5 time-steps with an assigned number of nodes equal to 20,000.

Figure 4.15 retraces the evolution of the velocity magnitude over time with the corresponding meshes. The initial flow inside the channel is at rest. As a result of the inflow on the left wall of the cavity, the wake starts increasing in size and two vortices emerge behind the cylinder (figure 4.15 (top)). This phase is followed by the detachment of the vortices from the cylinder and due to the asymmetry in the channel’s geometry, a vortex street shedding develops (figure 4.15 (bottom-left)). As the oscillatory vortices interact with the walls, the symmetry is completely broken (figure 4.15 (bottom-right)). The obtained behavior is in very good agreement with the reference [Volker 04b].

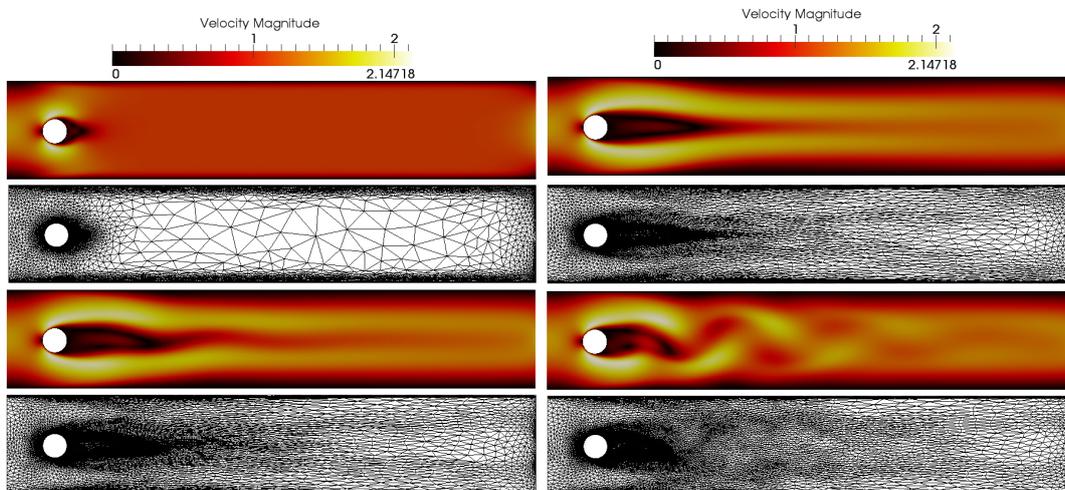


Figure 4.15 – Flow past a cylinder test case: evolution of the velocity magnitude (top) and of the mesh (bottom) at different time instances.

The time-step evolution history is depicted in figure 4.16. We can identify a profile that responds to the physics of the problem: small values are generated at the beginning to capture the emergence of the flow. An increase is then identified and is associated with the spread of the velocity inside the cavity. When the flow hits the cylinder, and the two vortices start to develop, a deceleration of computations is noticed allowing a better resolution of the new features. Once the characteristics of the flow have been identified and a vortex street is developed, the time-step sizes increase progressively allowing faster computations. We recall that the method works under a fixed number of nodes to produce the optimal mesh.

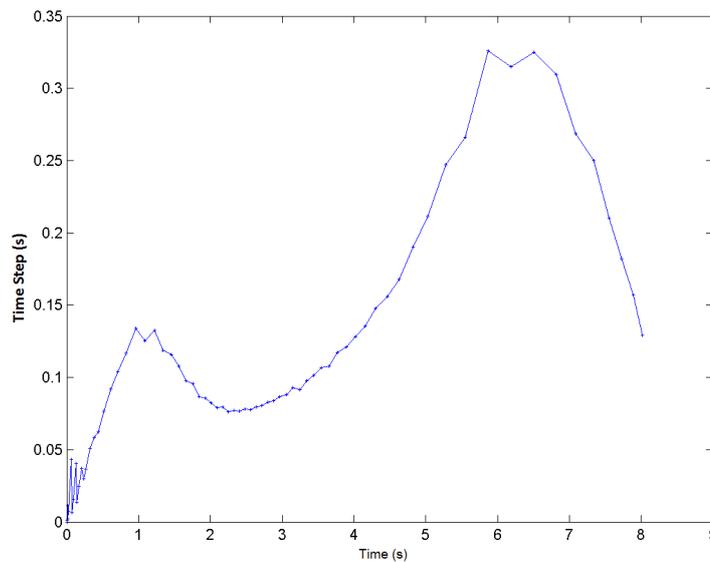


Figure 4.16 – Flow behind a circular cylinder: time-steps (s) history.

Figure 4.17 shows the evolution of the benchmark parameters over time. The first solution is computed using a fixed mesh of 50,000 nodes and a constant time-step  $\Delta t = 0.0025$ . It is in good agreement with the solution presented in [Schäfer 96]. The second solution is computed using the newly developed adaptive time-stepping algorithm. The plots show the good tendency of the approximated solution (obtained within only 73 time-steps) to the reference solution (obtained within 3200 time-steps) although the small wiggles in the coefficient are not reproduced. In fact, for this kind of problem, in order to get very accurate solution one needs to take very small time-steps. But since the aim of the algorithm is to simulate long time computations and come up with an approximate solution, the method seems very competitive as it is capable of capturing the general profile of the solution, not necessarily its details, within a few number of iterations. Further analysis on this example will be provided in the following chapter.

#### 4.3.5 Application to the unsteady convection-diffusion problem

In this subsection, we assess the performance of the newly developed anisotropic mesh and time adaptation algorithms on time-dependent convection-diffusion problems with sharp boundary layers. The focus is on the dynamic properties of the space-time adap-

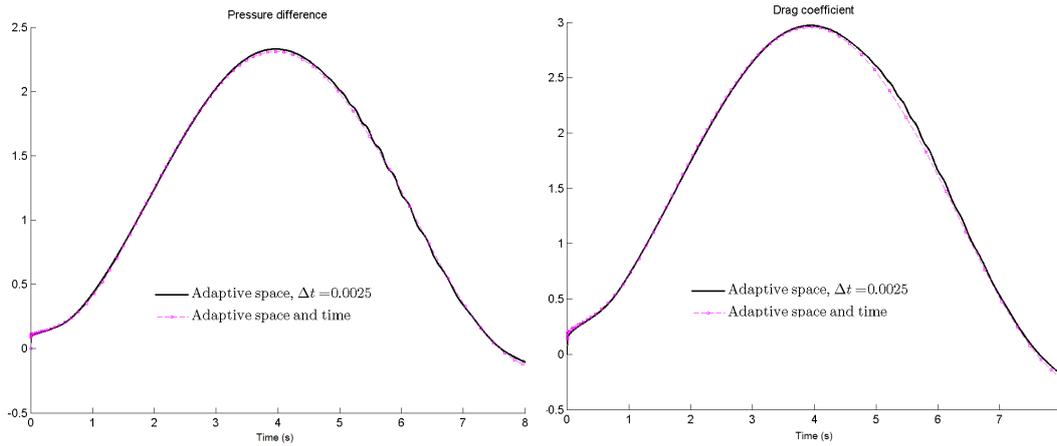


Figure 4.17 – Flow around a circular cylinder (2D): pressure difference  $P_{diff} = P(0.15, 0.2) - P(0.25, 0.2)$  (left) and drag coefficient (right).

tive procedure. In these examples, we present the numerical solutions obtained on problems with boundary layers and internal layers that move throughout the domain. We compare the obtained results with others from the literature [Picasso 09]. It will be demonstrated that the combination of the space and time adaptive algorithms constitutes a tool for numerical analysis. They allow the simulation of real unsteady convection dominated problems and produce the optimal numerical solutions with the available computational resources. These results have been the subject of a recent publication [Jannoun 14b].

#### 4.3.6 Convection-diffusion in a plane shear flow

We consider a problem proposed in [Baptista 95] using a space-time adaptive algorithm. It models the transport of a small source in a plane shear flow. The computational domain is set to  $\Omega = (0, 24000) \times (-34000, 34000)$ . We consider zero source term, a small diffusion coefficient  $\kappa = 1$ , a final time  $T = 9000s$  and a convective field  $\mathbf{v} = (\mathbf{v}_0 + \lambda y, 0)^t$  where  $a_0 = 0.5$  and  $\lambda = 1e^{-3}$ . The initial condition is a point source with mass  $m$  at  $(x_0, y_0) = (7200, 0)$ . The analytical solution of the problem is given by:

$$u(x, y, t) = \frac{m}{4\pi\epsilon t \left(1 + \lambda^2 \frac{t^2}{12}\right)^{1/2}} \exp^{-\zeta}$$

where

$$\zeta = \frac{(x - \bar{x} - \lambda y t / 2)^2}{4\epsilon t \left(1 + \lambda^2 \frac{t^2}{12}\right)} + \frac{y^2}{4\epsilon t} \quad \text{and } \bar{x} = x_0 + a_0 t.$$

We start the numerical resolution at time  $t = 2400s$  with

$$m = 4\pi\epsilon t_0 \left(1 + \lambda^2 \frac{t_0^2}{12}\right)^{1/2}$$

The objective of this test case is to assess and compare the performance of the

developed adaptation technique with the one proposed in [Picasso 09]. Therefore, we repeated the simulations using the same number of nodes.

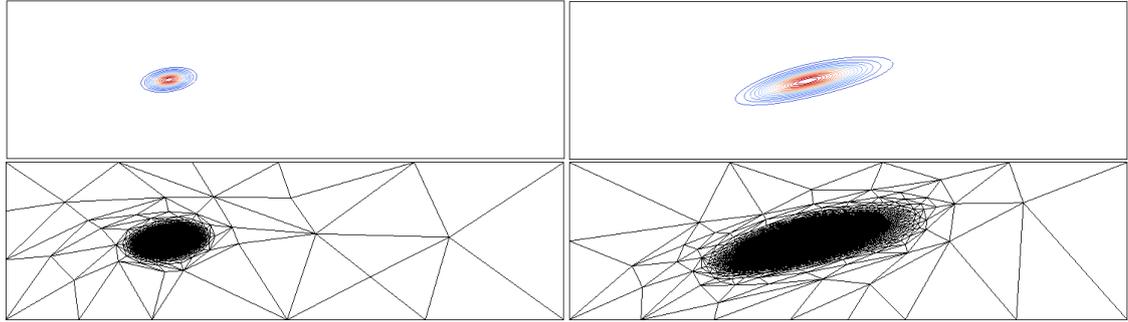


Figure 4.18 – The obtained isovalues and the adapted meshes at two different time-step.

Figure 4.18 presents the isovalues of the temperature field and the adapted mesh for a fixed number of nodes 8000. We point out how well the mesh is refined in the direction of high solution's gradient. The corresponding history of the approximation errors in the  $L_2$  norm and the number of time-steps to reach the adapted solution at the final time are provided in Table 4.3.

| Number of nodes | $\epsilon_{\text{present work}}$ | $\epsilon_{\text{reference}}$ | $N_{\tau \text{ present work}}$ | $N_{\tau \text{ reference}}$ |
|-----------------|----------------------------------|-------------------------------|---------------------------------|------------------------------|
| 1679            | 0.04556                          | 0.116                         | 27                              | 159                          |
| 3728            | 0.02303                          | 0.0551                        | 37                              | 206                          |
| 11990           | 0.00970                          | 0.0265                        | 58                              | 281                          |
| 40525           | 0.00149                          | 0.0129                        | 160                             | 400                          |

Table 4.3 – The approximation error and the number of time-steps needed to reach the adapted solution at final time.

We can see how the time-step size increases as the solution is convected and diffused. We note that the profile of the time-step variations is similar to the one presented in [Picasso 09] with a better control of the errors'  $L_2$  norm for almost the same number of nodes. We can identify that, for almost the same number of nodes, we achieve with the proposed adaptation algorithms a better control on the approximation error within a fewer number of time-steps.

Finally, we emphasize in Figure 4.19 that the optimal second order convergence is reached with respect to the number of nodes and to the number of time-steps. In accordance with [Picasso 09], we have the same order of convergence with fewer time-steps and a lower error which proves the efficiency and accuracy of the proposed technique.

#### 4.3.7 Internal and boundary layers

We present a second test case, taken from [Picasso 09], with challenging anisotropic features, exhibiting internal and boundary layers. The convective field is considered to be constant  $\mathbf{v} = (2, 1)^t$  in the whole computational domain  $\Omega = (0, 1)^2$ . The diffusion

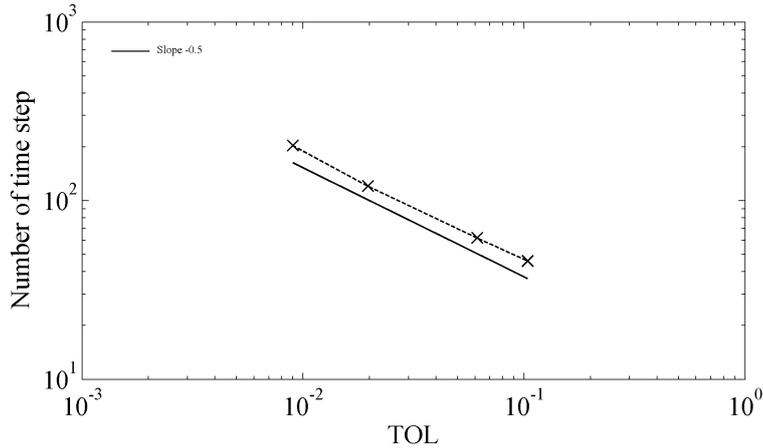


Figure 4.19 – Total number of time-steps with respect to the tolerance.

coefficient is set to  $\kappa = 10^{-3}$  and zero source term is assumed. The initial condition is zero everywhere except at the boundary where the boundary condition is defined by:

$$u(x, y) = \begin{cases} 1 & \text{if } \{x = 0, 0 \leq y \leq 1\} \cup \{0 \leq x \leq 1, y = 1\}, \\ \frac{\delta - x}{\delta} & \text{if } \{x \leq \delta, y = 0\}, \\ \frac{y - 1 + \delta}{\delta} & \text{if } \{x = 1, y > 1 - \delta\}, \\ 0 & \text{if } \{x > \delta, y = 0\} \cup \{x = 1, y \leq 1 - \delta\}. \end{cases}$$

As time advances, the gradient of temperature at the left boundary spreads into the domain creating an internal layer that reaches the right wall resulting in a boundary layer. On the other hand, the discontinuity at the top wall is reduced with time. Figure 4.20 shows the evolution of the temperature over time together with the corresponding anisotropically adapted mesh. A localized refinement of the mesh where the solution exhibits steep layers can be clearly observed. To keep up with the fixed number of nodes, the mesh is automatically coarsened at the locations of lower solution gradients. The plots also reflect how well the mesh follows the propagation of the solution with time.

Note the importance of this test case in reflecting the potential of the developed algorithm. If non adaptive mesh is used, a very fine mesh size would be necessary everywhere inside the domain and a very small time-step would be required to correctly capture the layers' propagation along the whole domain.

Figure 4.21 shows the history of the time-step sizes that responds very well to the solution's profile. We notice that the algorithm starts by generating small time-steps in order to allow a good capture of the solution's gradients. As the temperature diffuses, the size is progressively increased then as it stabilizes (around the time  $t = 0.5s$ ), significant increase of the time-steps is detected.

We have reported in Table 4.4 the obtained error and the total number of time-steps for different numbers of nodes  $N$ . As observed in the previous example, compared to [Picasso 09], we achieve a better control of the approximation error with fewer time-steps. Figure 4.22 shows that the optimal second order convergence with respect to

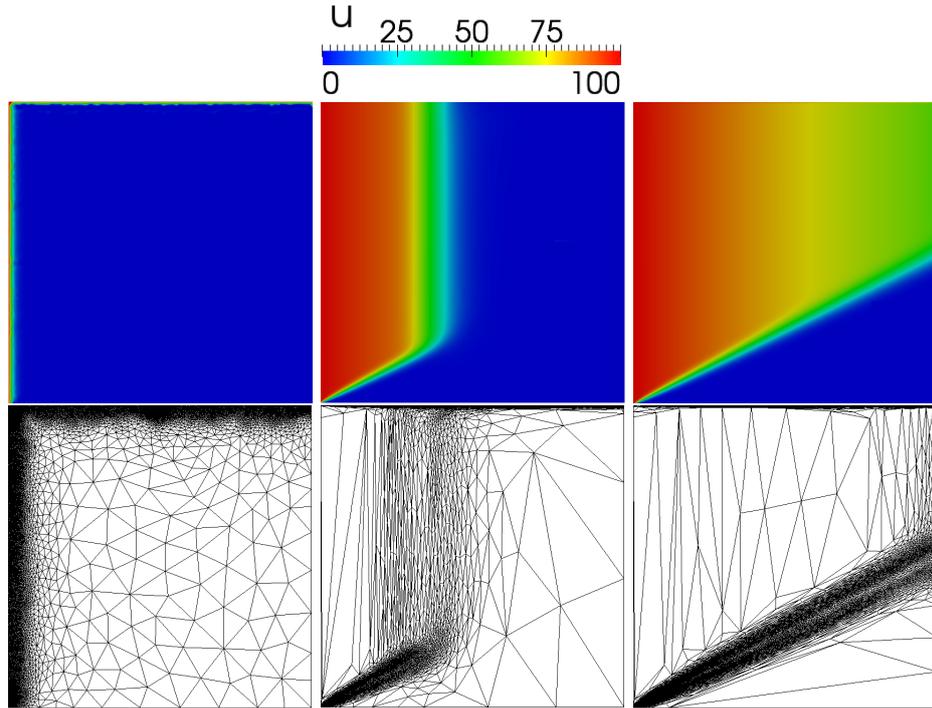


Figure 4.20 – Numerical solutions at different time-steps with their corresponding anisotropically adapted meshes.

the time-steps is again recovered as the number of time-steps is multiplied by  $\sqrt{2}$ , in accordance with the reference when the user's specified tolerance (TOL) is divided by two. It is important to mention that for comparison purposes, we have fixed the imposed mesh density to the one obtained in [Picasso 09] under a specified tolerance for the error.

| Number of nodes | $\epsilon_{\text{present work}}$ | $\epsilon_{\text{reference}}$ | $N_{\tau \text{ present work}}$ | $N_{\tau \text{ reference}}$ |
|-----------------|----------------------------------|-------------------------------|---------------------------------|------------------------------|
| 3987            | 0.10369                          | 0.25                          | 46                              | 163                          |
| 12222           | 0.06134                          | 0.125                         | 62                              | 247                          |
| 38874           | 0.01964                          | 0.0625                        | 121                             | 353                          |
| 140057          | 0.00900                          | 0.03125                       | 204                             | 502                          |

Table 4.4 – The error and the number of time-steps of the adapted solution at final time.

#### 4.3.8 Application to coupled heat transfer and fluid flows

The last two examples deal with 2D and 3D coupled heat transfer and fluid flow problems. We solved simultaneously the Navier-Stokes equations along with the heat transfer convection-diffusion equation. Therefore, the adaptivity takes into account different normalized fields: the temperature  $\frac{T}{T_{\infty}}$ , velocity  $\frac{\mathbf{v}}{\|\mathbf{v}\|_2}$ , and velocity norm  $\frac{\|\mathbf{v}\|_2}{\|\mathbf{v}\|_{\infty}}$ . More details on the multi-criteria adaptation will be provided in chapter 5. The aim of these numerical examples is to demonstrate the ability of the proposed method to automati-

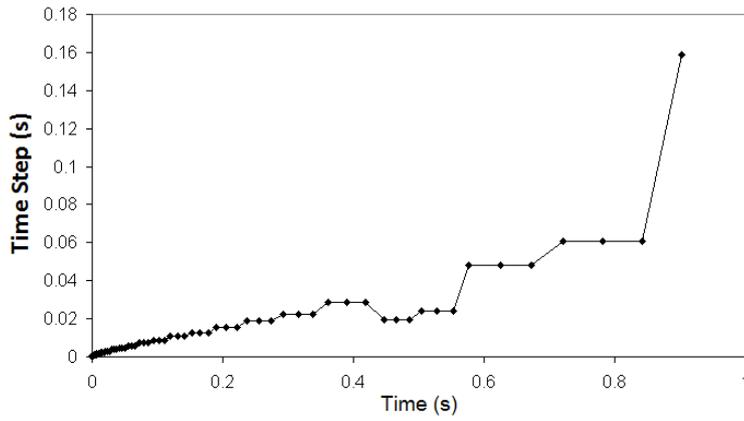


Figure 4.21 – Evolution of the time-step size over time.

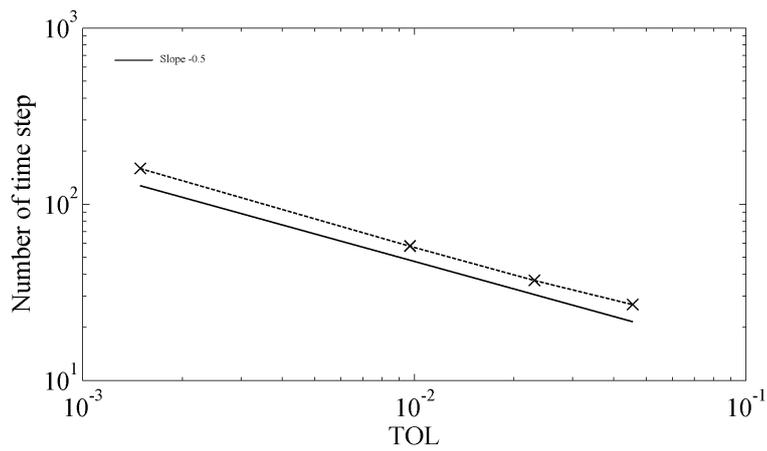


Figure 4.22 – Total number of time-steps with respect to the tolerance.

cally detect and reflect evolving/emerging layers, to produce oscillation free solutions and to reduce the computational costs of large scale simulations.

#### 4.3.8.1 Natural convection inside a cavity

As a first application to coupled heat transfer and fluid flows inside enclosures, we reconsider the 2D natural convection example 1.6.2. We repeat exactly the same setup for  $Ra = 10^6$  using the developed anisotropic mesh adaptation and time adaptive techniques for a fixed number of nodes 20,000. It is worth mentioning that a thorough analysis has been conducted in the thesis work of Veysset [Veysset 14a] to determine the adaptation criteria for this test case in two and three dimensional spaces. The objective was to determine the functional variables based on which the mesh shall be adapted in order to get the most accurate solution. Several configurations have been tested: adapting on the temperature field  $\left\{ \frac{T}{T_{\max}} \right\}$ , the velocity norm  $\left\{ \frac{\|\mathbf{v}\|}{\|\mathbf{v}\|_{\max}} \right\}$ , the velocity norm and direction  $\left\{ \frac{\mathbf{v}}{\|\mathbf{v}\|}, \frac{\|\mathbf{v}\|}{\|\mathbf{v}\|_{\max}} \right\}$ , and the temperature and velocity fields  $\left\{ \frac{T}{T_{\max}}, \frac{\mathbf{v}}{\|\mathbf{v}\|}, \frac{\|\mathbf{v}\|}{\|\mathbf{v}\|_{\max}} \right\}$ . The numerical results obtained by Veysset have shown that adapting on the temperature field yields the most accurate numerical solution. Decisions have been made based on the obtained Nusselt number and variations of the temperature isotherms. Indeed, the findings are in good agreement with the nature of the application since it is the gradient of the temperature that drives the flow inside the cavity as a result of the gravitational forces and density changes. Hence, in this example, we adapt the mesh on the temperature field every 5 time increments. Figure 4.23 depicts the temperature isotherms at time  $t = 20s$  for Rayleigh number  $10^6$  with its corresponding obtained anisotropic mesh.

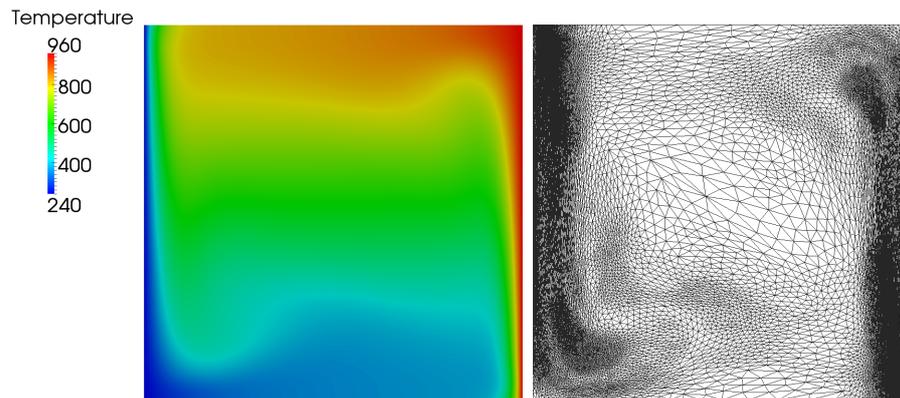


Figure 4.23 – Natural convection 2D: temperature isotherms at time  $t = 20s$  for  $Ra=10^6$  and the corresponding adapted mesh.

The elements in the generated meshes are well oriented and highly condensed in the region of steep temperature's gradient. The characteristics of the flow that is being developed can be inferred just by looking at the meshes. One can detect the vortical shapes formed by the mesh and reflecting the swirling motion of the flow.

Quantitatively, we are interested in evaluating the accuracy of the coupled

space/time adaptation and stabilized finite element methods by comparing statistics on the Nusselt number to the the ones obtained without adaptation and to existing benchmark values present in the literature [Paillère 05]. We present in table 4.5 statistics on the Nusselt number obtained in the present work and in the reference [Paillère 05] for  $Ra = 10^6$ . We can clearly identify a good agreement with the references for both the adaptive and the non-adaptive cases. However, the numerical results with only 20,000 nodes are closer to the reference values than the ones obtained on an isotropic non adapted mesh. Indeed, as the elements are well localized around the temperature's gradient, they transport the solution's variations over time with a higher precision. The small discrepancies in the results with respect to the references can be attributed to the difference in the discretization schemes. It is important to mention that in the references, quadratic finite element method and semi-implicit time discretization have been applied whereas our results were obtained using a linear finite element method and an implicit time discretization.

|                      | Vierendeels       | Dabbene  | Beccantini | Kloczko  | Heuveline | Non-adaptive* | Adaptive* |
|----------------------|-------------------|----------|------------|----------|-----------|---------------|-----------|
| $\overline{Nu(0,y)}$ | 8.85978           | 8.86380  | 8.85990    | 8.86200  | 8.859778  | 8.73698       | 8.79884   |
| $\overline{Nu(1,y)}$ | 8.85978           | 8.86200  | 8.86007    | 8.86380  | 8.85978   | 8.73056       | 8.79295   |
| $Nu(0,0.5)$          | 7.81938           | 7.82170  | 7.81978    | 7.82010  | N.A.      | 7.65966       | 7.77361   |
| $Nu(1,0.5)$          | 8.79636           | 8.81710  | 8.79646    | 8.79750  | N.A.      | 8.61058       | 8.75864   |
| $\max Nu(0,y)$       | 19.59642          | 19.62600 | 19.59538   | 19.61070 | 19.59633  | 19.19471      | 19.45033  |
| $\min Nu(0,y)$       | 1.07345           | 1.07690  | 1.07356    | 1.07380  | 1.07345   | 1.05583       | 1.06756   |
| $\max Nu(1,y)$       | 16.36225          | 16.35200 | 16.36333   | 16.37510 | 16.36226  | 15.9499       | 16.20312  |
| $\min Nu(1,y)$       | 0.85512           | 0.86102  | 0.85542    | 0.85620  | 0.85513   | 0.86811       | 0.85482   |
| Mesh density         | $4.2 \times 10^6$ | 102400   | 87616      | 57600    | 200000    | 50000         | 20000     |

Table 4.5 – Natural convection: statistics on the Nusselt number for  $Ra = 10^6$ . Reference values are taken from [Paillère 05]. Results obtained in the present work on the two rightmost columns.

#### 4.3.8.2 Forced convection inside a cavity

As a second validation on applications involving heat transfer and fluid flows, we consider a 2D forced convection problem. This test case can be regarded as a simplified model of a gas-fired furnace or a temperature distribution inside a room.

Heated air is pumped into the enclosure from an inlet at the left wall with a velocity of magnitude  $1m/s$  and a temperature of  $100K$ . Adiabatic condition for the temperature is applied at all other boundaries. The air is vented out of the enclosure through the outlet positioned at the right wall. The computations were run over a period of 10,000s and the mesh adaptation algorithm is called every 5 time-increments.

For a fixed fluid density  $\rho = 1kg/m^3$ , we have considered two different dynamic viscosity values ( $\mu = 0.01Pa \cdot s$  and  $\mu = 0.0005Pa \cdot s$ ) to test the performance of the adaptive algorithms for quasi-static and unsteady turbulent flows. We have also varied the parametric components of the vector field used to adapt the mesh. The aim of this test case is first to select the best set of parameters for meshing and second to validate the efficiency of the space and time adaptive techniques. The numerical results have shown that the adaptation vector field taking into account the velocity and the temperature i.e.  $\frac{T}{T_\infty}, \frac{\mathbf{v}}{\|\mathbf{v}\|_2}, \frac{\|\mathbf{v}\|_2}{\|\mathbf{v}\|_\infty}$  gives the best results. Figure 4.24 shows the obtained

approximate temperatures and velocity streamlines and their corresponding meshes obtained using the latter adaptation criteria. Note the concentration of the elements, not only along the boundary layers, but also around the emerging vortices and detachment regions. This reflects how, for a fixed number of nodes equal to 20,000, the mesh is automatically coarsened in regions with low solution gradient and is refined in regions of high gradients. Also note the alignment between the direction of the gradient and the direction of refinement which allows an accurate capture of the physical phenomena's anisotropy.

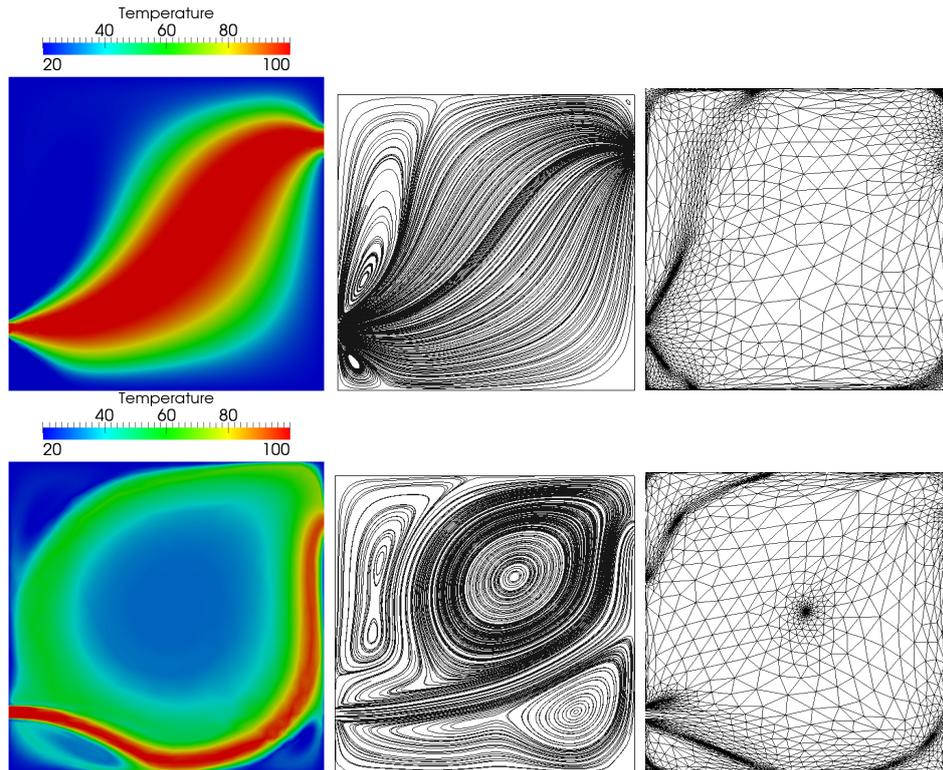


Figure 4.24 – Temperature fields and velocity streamlines with the corresponding adapted mesh for  $\mu = 0.01$  (top) and  $\mu = 0.0005$  (bottom).

We have tested the time-step algorithm with the different possible combinations for the adaptation vector field. Figure 4.25 provides the evolution of the time-step sizes. The profiles of time-stepping, for the different combinations, are as expected in the case where the solution stabilizes after a certain time. However, it can be noticed from the plot corresponding to the unsteady case that when adapting with respect to the velocity field only, the time-step grows significantly and in turn the resulting numerical solution is never stable. However, when adapting on the temperature alone and on the combination of both the temperature and the velocity field, the time-step size oscillates around a small value allowing a good capture of the solution's details. This highlights once more the capability of the algorithm in producing automatic and controlled time-steps.

Figure 4.6 shows a comparison of the execution times and the computational costs needed for the simulation of the test cases. Note that these results correspond to adapt-

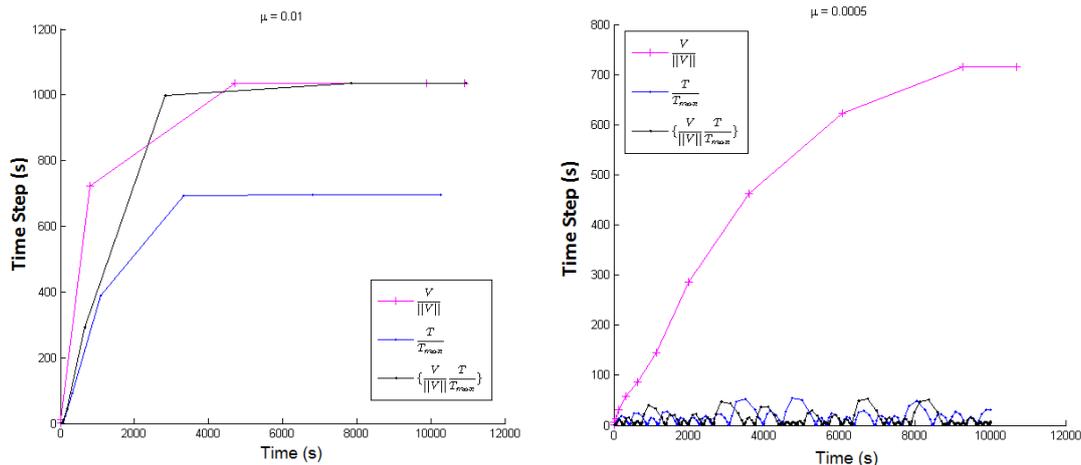


Figure 4.25 – Time-step evolution (s) for the different possible combinations for adaptation.

|                         | Execution time (s) |                | # time-steps |                |
|-------------------------|--------------------|----------------|--------------|----------------|
|                         | $\mu = 0.01$       | $\mu = 0.0005$ | $\mu = 0.01$ | $\mu = 0.0005$ |
| Fixed $\Delta t = 0.01$ | 21,600             | 32,750         | 1,000,000    | 1,000,000      |
| Adaptive $\Delta t$     | 420                | 1800           | 310          | 3,788          |
| Ratio                   | 51.4               | 18.19          | 3225         | 263.99         |

Table 4.6 – Execution time and computational cost

ing on both the temperature and the velocity field as it presented the most accurate results for this problem. It is obvious that for the same level of accuracy, the time-adapted solution requires much less computations and CPU time for both laminar and turbulent cases and thus leads to accelerating the computations and saving time and cost. If a constant time-step should be used for the whole simulation, a question arises regarding the choice of its size. Most of the time this choice is done by trial and error and is not founded on a study of consistency and stability. The originality of this work relies in its automaticity and consistency in choosing the time-stepping sizes with respect to the spatial and temporal errors.

#### 4.3.9 Application to 3D heat transfer and turbulent flow inside an industrial furnace

The last numerical example is dedicated to the simulation of the heating processes inside a 3D industrial furnace. The idea is to assess the capability of the proposed algorithms to deal with a complex 3D geometry and to handle simulating long time heating inside large scale furnaces. For that purpose, we reconsider the same furnace studied in chapter 1 with the same boundary conditions and fluid properties. In order to focus on the capability of the adaptive methods to capture solution's gradients in large and complex geometries and simplify the setup of the problem, we assume that the furnace is empty. The real furnace heating test cases in the presence of workpieces will be treated in the following chapters. The 3D computations aim at simulating an hour of heating and have been conducted in parallel on 32, 2.4Ghz Opteron cores.

Figure 4.26 (top) shows the isothermal distribution at different time-steps. When the hot fluid spreads along the volume of the furnace, it induces a turbulent motion within the geometry. This forced convection is caused by the interaction of the moving stream and the stationary fluid inside the furnace. The numerically obtained temperature distribution clearly reflects the expected flow pattern. A number of small vortices inside different buffer zones can be observed. The latter are due to the turbulence dissipation and mixing of the hot and cold air.

Figure 4.26 (bottom) highlights how well the mesh is adapted to capture the gradients of the solution, the boundary layers and the emerging vortices. All the boundary layers as well as the vortices are sharply captured and identified. The obtained meshes show the stretching of the elements and the high resolution near the walls, in the corners and at the location of steep gradients. The anisotropic adaptive procedure modifies the mesh, according to the fixed number of nodes (100,000), in such a way that the local density is adequately distributed in all directions. The estimates were driven based on the  $L_2$  norm of the interpolation error. Recall again that the algorithm builds up the mesh in a way to maximize the accuracy of the numerical solution. Note that for this forced convection, the mesh is adapted only according to the velocity components.

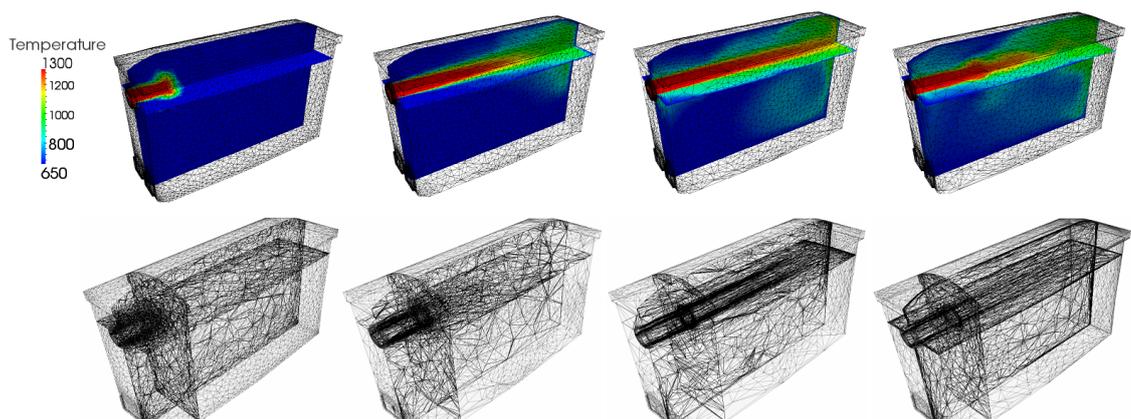


Figure 4.26 – Isotherms inside the furnace (Top) and corresponding adapted meshes (bottom) at three different time-steps.

The results describing only one hour of the heating process required around 58 days of computations on 32 cores with a fixed mesh (of around 100,000 nodes) and time-step size equal to 0.005s whereas it required around 80 days when applying the developed anisotropic mesh adaptation for almost the same number of degrees of freedom and the same time-step size. Both computations were performed using the  $k - \epsilon$  turbulence model. Although a higher computational cost is induced when adapting the mesh, the computed solution is more accurate as the nodes are well distributed to provide an optimal capture of the problem's anisotropic features. However, in both cases, the required computational time is important. This difficulty pointed out the necessity to supply fast algorithms capable of reproducing the full heating sequences in a reasonable time. Significant CPU time and computational cost were saved by applying the time adaptive procedure as it required only 2 days of calculations (see table 4.7)

without the use of a turbulence model and 5.23 hours when using the  $k - \varepsilon$  model. It is important to mention that the anisotropic mesh adaptation algorithm was employed at a frequency of 10 time increments. Qualitative comparisons of the numerical solutions generated will be provided in chapter 7 where real configurations will be considered and experimental data will be used for validation.

|          | With $k-\varepsilon$<br>fixed mesh<br>and $\Delta t = 0.005$ | With $k-\varepsilon$<br>adaptive mesh<br>and $\Delta t = 0.005$ | Without $k-\varepsilon$<br>adaptive | With $k-\varepsilon$<br>adaptive |
|----------|--------------------------------------------------------------|-----------------------------------------------------------------|-------------------------------------|----------------------------------|
| CPU time | 58 days                                                      | 79.5 days                                                       | 52 hours                            | 5.23 hours                       |

Table 4.7 – Statistics on simulating the heating of an empty furnace with non-adaptive, space adaptive and a space/time adaptive methods.

We note that the implementation of the numerical tools gives the choice to the user to use or not a turbulence model. Figure 4.27 depicts the time-stepping history for computing one hour of heating without resorting to a turbulence model. It shows that the evolution of the time-step sizes responds very well to the behavior of the solution. When the air is pumped from the top inlet, the difference between the temperature of the air and that inside the load creates a high temperature gradient. In response, the time-step size decreases to capture this phenomenon. As the air propagates inside the volume, the gradient decreases and hence the time-step size increases allowing a faster diffusion of the flow. Once the air hits the opposite wall, a new high gradient emerges reducing the time-step size. The latter size continues to oscillate around a certain value allowing at the same time a good level of accuracy and an acceleration of the computations. On the other hand, when using the  $k - \varepsilon$  turbulence model as presented in figure 4.28, the generated time-step sizes are much bigger and in accordance with the smoothing of the solution scales. Clearly the level of accuracy is not the same since not all the details of the solution are modelled. However this model can be very useful for the industrials as it provides very rapid testings of *what-if* scenarios and getting an idea of the velocity profile. We highlight the reduced execution time required to simulate an hour of heating with this turbulence model.

It is important to mention that the frequency of remeshing affects the quality and accuracy of the final solution. We compare in figure 4.29 the magnitudes of the velocity fields obtained after 50s on a horizontal plane cut along the z-axis at the burner's level. For these results, and for illustration purposes, we have fixed the time-step size to 0.05s. Clearly, the velocity profile gets more diffusive as the frequency of remeshing increases for the same number of nodes 100,000. This is in accordance with the construction of the anisotropic mesh algorithm. Indeed, the constructed mesh is not predicted for a slab of time i.e. its validity is not controlled. The generated anisotropic mesh is not optimal all over the interval of time between two remeshings and consequently, it is not well adapted to contain all the evolving gradient of the velocity field. When combining the anisotropic mesh adaptation with the time-adaptive algorithm, a pulse graph evolution of the time-step sizes is obtained (as in figure 4.27). At every remeshing, a larger

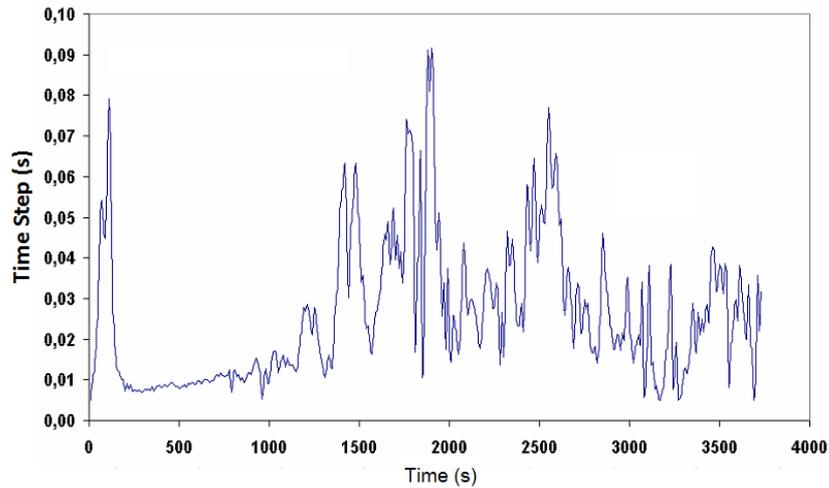


Figure 4.27 – Time-step evolution (s) for the simulation of an hour of the heating process inside an industrial furnace.

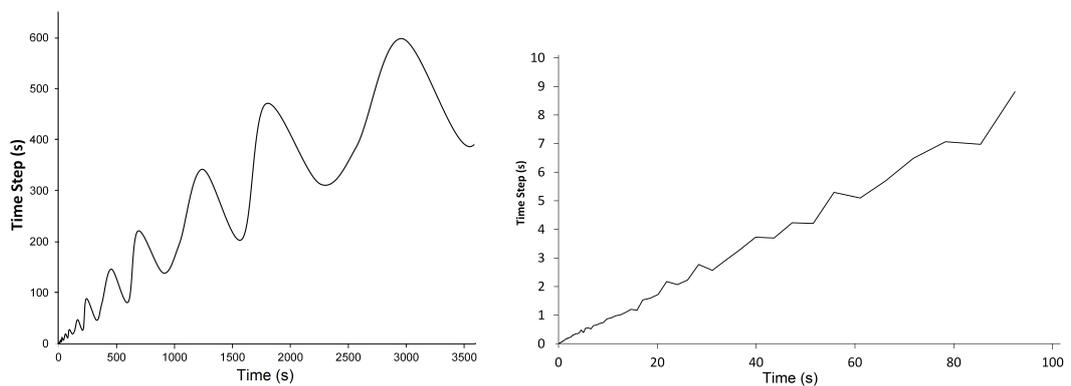


Figure 4.28 – Time-step evolution (s) for the simulation of an hour of the heating process inside an industrial furnace using the  $k - \epsilon$  turbulence model. An hour of fluid flow (left) and zoom on the first 100s (right) are depicted.

time-step size is generated since the mesh is well refined to contain the evolving solution. However, as the solution advances in time between two-remeshings, the induced interpolation error becomes larger and to maintain the temporal error bounded by the equidistributed error, the time adaptive algorithm tends to generate small step-sizes in order to maintain the velocity's sharp gradient within the well refined zones. This problematic of the adequacy between frequency of remeshing and the generated anisotropic meshes will be reconsidered in chapter 6 where we extend the algorithms to generate anisotropic meshes and time-step sizes that account for the solution's evolution over slabs of time.

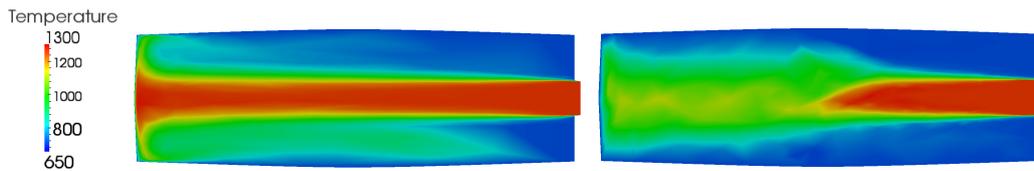


Figure 4.29 – Velocity magnitude on a horizontal plane cut along the  $z$ -axis at the burner's level obtained at time  $t = 50s$  when adapting the mesh every 5 time-increments (left) and every 20 time-increments (right).

## 4.4 CONCLUSION

In this chapter we have developed a time-adaptive algorithm based on the extension of the anisotropic mesh adaptation. It involves error estimation along the temporal edges constructed at each spatial node and orthogonal to the space dimensions. In order to maintain an accurate adaptation in both space and time, the error equi-distribution is also applied in time by not letting the temporal error exceed the spatial one. We recall that one of the main objectives of this work is to provide a method that can be used to simulate diverse industrial applications. For practical reasons, and as noted previously, it is highly desirable to find the most accurate solution using the available computational resources. Inhere the imposed number of nodes is the only parameter that controls at the same time the adaptive meshing and the time-stepping. Consequently, given an idea about the available computational power, the anisotropic mesh adaptation and time-adaptive technique generate the optimal meshes and time-step sizes for advancing the solution of the problem with good levels of accuracy and efficiency. The accuracy and efficiency of the developed algorithm were evaluated on a 2D analytical test case, flow problems, unsteady convection-diffusion problems, and coupled heat transfer and fluid flow problems. In all the cases a significant gain in computational time was highlighted reflecting the importance of the method for the simulation of complex three dimensional problems.

So far, we have tackled the heat transfer and turbulent flow problems and tested the developed space and time adaptive methods on problems involving fluid motion and thermal heat transfer. However, in order to perform real physical simulations, one

has to treat problems exhibiting both solid and fluid components that can arise in a wide variety of multi-disciplinary applications. In the following chapter, we will solve multi-material problems. The modelling should take into account the solid part, the fluid part as well as their interaction resulting in a full thermal coupling framework.

## 4.5 RÉSUMÉ FRANÇAIS

Dans ce chapitre, nous avons développé un algorithme d'adaptation de temps basé sur l'extension de l'adaptation de maillage anisotrope présentée dans le chapitre 2. Il s'appuie sur une estimation d'erreur temporelle sur des arrêtes construites à chaque nœud spatial et en direction orthogonale aux dimensions de l'espace. Afin de maintenir une adaptation précise à la fois dans l'espace et le temps, l'erreur est équi-distribuée également dans le temps en limitant l'erreur temporelle par l'erreur spatiale. Nous rappelons que l'un des principaux objectifs de ce travail est de proposer un procédé qui peut être utilisé pour simuler des applications industrielles diverses en temps réel. Pour des raisons pratiques, et tel que mentionné précédemment, il est hautement souhaitable de trouver la solution la plus précise en utilisant les ressources de calcul disponibles. Le principal avantage des méthodes d'adaptation proposées en espace et en temps est qu'elles dépendent d'un seul paramètre, le nombre de nœuds imposé dans le maillage. Par conséquent, étant donné une idée de la puissance de calcul disponible, l'adaptation de maillage anisotrope et l'approche d'adaptation temporelle génèrent des maillages et des pas de temps optimaux pour faire avancer la solution du problème avec de bons niveaux de précision et d'efficacité.

La précision et l'efficacité de l'algorithme d'adaptation de temps développé ont été évaluées sur un cas test analytique 2D, des problèmes d'écoulement de fluides, des problèmes de convection-diffusion instationnaires et des problèmes de traitements thermiques. Dans tous ces cas un gain significatif de temps de calcul a été souligné reflétant l'importance de la méthode pour la simulation de problèmes complexes en trois dimensions de l'espace.

Jusqu'à ce point, nous avons abordé des problèmes de transfert de chaleur et d'écoulements turbulents et testé les méthodes d'adaptations espace et temps développées sur des problèmes couplant mouvement de fluide et transfert de chaleur thermique. Toutefois, afin de simuler des problèmes physiques réels, il faut prendre en compte à la fois des composants solides et fluides qui peuvent se présenter dans diverses applications multi-disciplinaires. Dans le chapitre suivant, nous allons résoudre des problèmes multiphasiques. La modélisation va prendre en compte les interactions fluides/structures dans un cadre thermique fortement couplé.



# IMMERSED VOLUME METHOD AND MULTI-DOMAIN ADAPTATION

“Heaven’s net is wide, but its mesh is fine”

Lao Tzu

## CONTENTS

---

|       |                                                  |     |
|-------|--------------------------------------------------|-----|
| 5.1   | STATE OF THE ART                                 | 231 |
| 5.2   | IMMERSED VOLUME METHOD                           | 234 |
| 5.2.1 | First component: the levelset function           | 234 |
| 5.2.2 | Second component: anisotropic mesh adaptation    | 238 |
| 5.2.3 | Third component: mixing laws                     | 243 |
| 5.3   | MESH ADAPTATION BASED ON SEVERAL FIELDS          | 246 |
| 5.3.1 | Metric intersection                              | 246 |
| 5.3.2 | Multi-components metric construction             | 247 |
| 5.4   | NUMERICAL VALIDATION                             | 251 |
| 5.4.1 | Flow behind a circular cylinder                  | 251 |
| 5.4.2 | Turbulent flow behind a F1 racing car at 300km/h | 254 |
| 5.4.3 | Forced turbulent convection                      | 256 |
| 5.4.4 | Simulation of a rotating helicopter propeller    | 260 |
| 5.5   | CONCLUSION                                       | 262 |
| 5.6   | RÉSUMÉ FRANÇAIS                                  | 263 |

---

**T**HIS chapter presents advancements toward a monolithic approach with anisotropic mesh adaptation and time-stepping control for the numerical resolution of Fluid Structure Interaction (FSI) problems. The Immersed Volume Method (IVM) will be introduced to embed solid geometries inside fluid domains. The method is combined with the newly developed anisotropic mesh adaptation to ensure an accurate capture of the fluid/solid interface and a precise material distribution over the computational domain. A multi-component metric construction will also be proposed accounting for several fields for adaptation. The proposed method demonstrates its efficiency and accuracy in simulating 2D and 3D unsteady heat transfers and turbulent flows inside complex geometries in the presence of conducting solids.



## 5.1 STATE OF THE ART

Nowadays, the modelling and simulation of fluid-structure interactions (FSI) are considered challenging and intrinsic fields of multi-physical analysis. They involve the study of the interaction between a solid body and its surrounding media of fluid flows. FSI applications are involved in a wide variety of disciplines including aerodynamics, automotive, biomedical engineering, metal forming and heat treatment. The numerical simulations can contribute to the optimization of manufacturing markets, the analysis of biomedical fluids, the tracking of tumor generation and propagation, the development and design of high speed airships and many other fields of multi-physical problems. Clearly, the characteristic scale of applications can range from the micro-scale (blood cells) to the macro-scale (industrial furnaces) or even the combination of both small and large scales. To achieve an accurate capture of all the scales with a good level of robustness and efficiency, several questions need to be addressed regarding the modelling of the fluid and solid counterparts, the partitioning of the computational domain, the discretization schemes, and the computational power.

In the context of fluid mechanics, the resolution of a flow is often sought in a spatial domain delimited by solid boundaries. In other words, a body fitted mesh is used for the simulation. However, for accurate and consistent understanding, coupling between the fluid and its surrounding shall be emphasized in which case their influence on one another is reflected by the numerical results. Therefore, the use of body fitted meshes should be backed up by a numerical modelling or boundary conditions accounting for the effect of the structure on the properties of the flow.

In this thesis, we are interested in the numerical simulation of thermal coupling problems involving multiphase flows. The general objective is to develop powerful and efficient tools capable of reflecting the thermal history of the full conjugate heat transfer process. More precisely, we intend to perform a full heating/cooling simulation of high alloy steel. These results will then be synthesized by the industrial partners in the view of improving the thermo-mechanical properties and final microstructure of the treated workpieces. The problem can therefore be posed as a thermal fluid structure interaction one, the resolution of which necessitates a technique for modelling the exchanges between the solid and the fluid parts.

The most common trend to deal with multi-component domains is issued from the body fitted approach whereby the global domain is partitioned into several local subdomains over each of which a local model is solved. This technique is a sort of multi-level multi-grid method [Aulisa 06]. The global solution is then recovered by gluing together the local solutions. Nevertheless, the assembly may become cumbersome and infeasible because of the communication between the sub-domains. This is due to the complexity of determining closure laws for inter-domain variables. We can find in the literature several methods relying on this principle for the numerical simulation of multiphase flows. In particular, the immersed boundary method [Peskin 77] is a very well-known technique for multi-domain applications. It employs an Eulerian mesh for the fluid

domain and a Lagrangian one for the structure. The Lagrangian coordinate system follows the movement of the local fluid velocity and tracks the positions of the solid over time. The solid's effect on the flow is accounted for by embedding into the problem's formulation an additional source term that accounts for the influence of the immersed interface on the flow. Moreover, a three-field formulation was proposed in [Farhat 95] where in addition to the fluid and the solid subdomains, a dynamic fluid-mesh is resolved. This method improved the efficiency of domain-partitioning approaches.

On the other hand, FSI problems can be treated using immersed methods. We can distinguish in the literature three classes of methods for immersing geometries into a computational domain. The first group gathers the methods that tend to enrich the numerical solution's space in a finite element framework. This method avoids the computational cost needed to construct a mesh that is well representative of the immersed bodies but requires the modification of the finite element solvers. The X-FEM (extended finite element method) employs a local enrichment of the functional space together with the levelset method to render the interfaces, more details about this method are provided in [Moës 99].

The second accounts for the effect of the immersed objects by introducing an additional source term into the governing equations. The immersed interface method [Lee 03] uses the jump function across the interface to determine the approximate solution along it. The ghost fluid method [Fedkiw 99] benefits from a finite difference discretization to define a coordination between subdomains. The fictitious domain method [Glowinski 95] is also commonly known for multi-domain problems whereby the interface continuity between subdomains is enforced through the use of Lagrange multipliers. However, when using these techniques of treating multi-physical domains for the simulation of conjugate heat transfer problems, a heat transfer coefficient needs to be identified and employed as a boundary condition at the level of the interface. In practice, the identification of this coefficient is not straightforward as it requires several experimental studies and is in direct connection with the setup of the problems. That is, if any of the problem's parameters has been modified then a new experiment shall be conducted. Consequently, these methods can rapidly become very expensive and time-consuming.

The interface tracking methods constitute a third class of immersion [Lakehal 02]; they propose a different scenario than the previously mentioned techniques. They benefit from a single fluid formulation with variant material properties. A single set of equations is resolved for the multi-physical domains and their interfaces are tracked by levelset functions. Compared to the multi-level multi-grid methods, the interface tracking methods allow a more subtle and accurate representation of interfaces hence overcoming the drawback of transferring information between subdomains. The immersed volume method (IVM) [Hachem 10a, Bernacki 08, Valette 09, Bruchon 09, Hachem 13] was defined in that sense and can be applied on a wide range of multi-material applications. The key feature of this approach is to retain the advantages of a monolithic formulation and couple it with anisotropic mesh adaptation to provide a high resolu-

tion at the interfaces. In the monolithic framework, a levelset, signed distance function is used to delimit the different sub-domains. The latter are treated as a single fluid with different material properties, and hence the solution is evolved with a single set of equations on the global domain. Indeed, Klaus'Jürgen Bathe [Bathe 09] once said that *“only in the human mind are fluids separated from solids”* as the mechanical principles of both materials are the same but it is the response to these principles that differs. Thus, by accounting for the differences in response, a single system of equations and a single domain can be adopted for both components. Applied in the context of conjugate heat transfer problems, the method bypasses the need for empirical data such as heat transfer coefficients. The exchanges are rather obtained in a natural way by solving a convective equation for the phase indicator field [Lakehal 02]. For the resolution of the governing equations, the methods introduced in chapter 1 can be adopted with a minor modification to account for the non-constant, homogenous aspect of the thermal material properties. The second feature of the IVM method is the use of anisotropic mesh adaptation to well capture the interfaces between the materials. These interfaces are defined as the zero isovalues of a distance function, called the levelset function.

Prior to this work, the anisotropically adapted mesh is generated in the pre-simulation stage. It is constructed to be very fine at the interface level in order to permit a high precision in distributing the physical properties. As the involved solids were considered to be fixed, the same mesh can be maintained all over the computations and yields the same level of accuracy near the interface. However, when moving interfaces shall be addressed, the mesh should be adapted over the computations to conserve the desired accuracy. On the other hand, in addition to the interfaces, good captures of the flow changes and heat transfers with an affordable computational cost are highly desirable. The originality and novelty brought up by this work rely on the capability of the anisotropic mesh adaptation to account for several fields at the same time. The mesh can be automatically adapted to well represent the interfaces as well as the sharp gradients of the problem's variables.

In this chapter, we are interested in explaining how the space and time adaptation techniques can be extended to deal with multiple fields. Furthermore, we intend to consider a multi-domain method for solving coupled heat transfer problems with complex geometries. For that purpose, we resort to the immersed volume method for immersing and distributing the multi-physical properties inside a multi-components domain. The method involves solving a single set of equations for both domains with variable material properties. We start in section 5.2 by introducing the immersed volume method and its three main components: the monolithic formulation, the anisotropic mesh adaptation and the mixing laws. Then we point out its advantages over other methods for multi-material simulations. We move on in the second part of this chapter to elaborate on the multi-component mesh adaptation. We provide afterward some validations of the newly developed numerical tools on fluid/structure interaction problems.

## 5.2 IMMERSED VOLUME METHOD

Recently, immersed methods have been the subject of advanced research for various CFD applications. When simulating physical phenomena with complex 3D geometries, the generation of a mesh fitting the geometry may require a considerable computational time and cost especially because the mesh shall be maintained with the same resolution at the interface level so that to guarantee the same level of accuracy. Hence, a numerical simulation with non-body fitted grid seems a good resort for significantly improving the computational costs. In spite of the advantage of the methods relying on non-body fitted meshes, they require a special non-straightforward interface treatment. For instance, in the context of conjugate heat transfer problems, such as the heating of ingots inside industrial furnaces, heat transfer coefficients between the solid bodies and the surrounding fluid need to be prescribed as boundary conditions at the level of the interface. Recent developments address the immersion of 3D complex geometries, the development of interface detection algorithms as well as the boundary conditions at the interface level [Ilinca 11, Abgrall 14, Quan 14]. The immersed volume method (IVM) [Bernacki 08, Valette 09, Bruchon 09, Hachem 10a, Hachem 13] can be considered as a good tool for reducing the computational cost induced by body-fitted approaches and overcoming the interface treatment step required by non-body fitted techniques. It involves solving a single set of equations for the whole computational domain and accommodates for the different subdomains by considering a single fluid with variable material properties. We can distinguish three main ingredients of the immersed volume method that we will detail in the sequel. As a first step, the levelset, a signed distance function that is used to localize and immerse the solids inside the global domain. The fluid-solid interface is hereby identified as the zero isovalue of the distance function. Then an anisotropic mesh adaptation is employed to provide a good capture of the interfaces between the solid and the fluid parts. The third step consists in mixing the different material properties using a homogenous distribution along the interface. The IVM method can be applied without modification for any geometry and any physical property. It can be very easily implemented and applied with stabilized finite element methods and in the context of moving structure problems.

### 5.2.1 First component: the levelset function

This section is devoted to illustrate on the signed distance function used for defining the interfaces between immersed bodies and their surrounding. In the context of monolithic formulation, we aim at using a single mesh for both solid and fluid subdomains. For simple geometries, we resort to analytical functions to define the solids (like a sphere, a square,  $\dots$ ) and build up their corresponding mesh. However, in general industrial applications, we intend to immerse geometrical objects whose analytical shapes are not known. In these cases, an object is defined in a CAD environment. After cleaning the CAD geometry (eliminating small details, trimming overlapping surfaces,  $\dots$ ), a surface mesh of the object is generated using an STL mesher. The latter is made

up from segments in 2D simulations and from triangles in 3D simulations. The surface mesh is then embedded in the computational domain which is in turn remeshed resulting in a single fluid/solid mesh. An accurate representation of the body is obtained through successive mesh adaptations. More details about this step will be provided in the following section. Figure 5.1 provides a schematic representation of the mesh generation process accounting for immersed bodies.

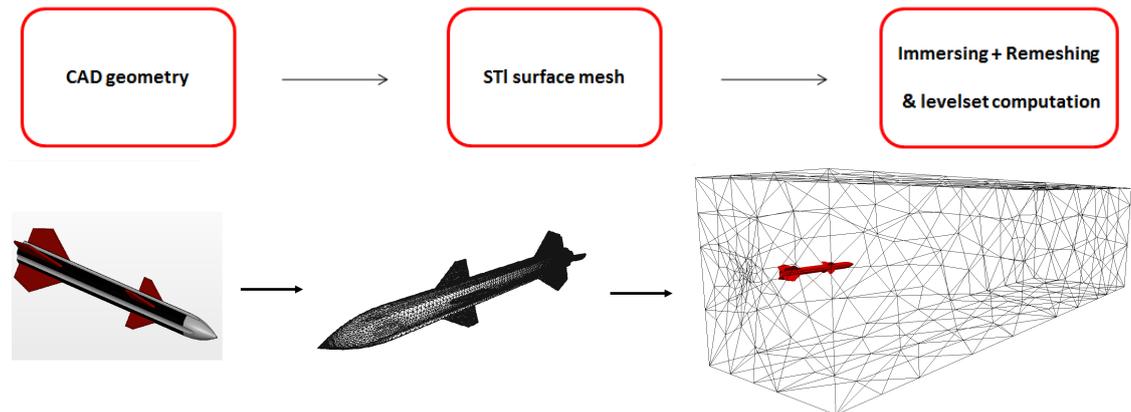


Figure 5.1 – Schematic for immersing objects inside computational domains.

The first step in the immersed volume method is to determine the signed distance from any point in the computational domain to the immersed solid body. The levelset function, first introduced by Stanley and Sethian [Stanley 88], is a real valued Lipschitz function  $\alpha : \mathbb{R}^3 \mapsto \mathbb{R}$  that evaluates the signed distance to the surface of an object. For simple objects, defined by their analytical forms, the distance function is easily determined analytically. For instance, in a 3D computational domain, the distance from a point  $\mathbf{X}^i = (x, y, z)$  in the domain to a spherical object with center  $(x_0, y_0, z_0)$  and radius  $r$ , is computed by:

$$d(\mathbf{X}^i) = \left| r - \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \right| \quad (5.1)$$

Nevertheless, when immersing a complex object, analytical functions representing the distance cannot be found and the distance to the geometry's surface requires a special algorithm. To that regard, we follow the lines of [Bruchon 09] to compute the signed distance function. This function will then be used to delimit the interface of the object. In order to determine the distance between the mesh nodes and the geometry, we iterate over the nodes and find, for each one of them, the distance that separates it from each facet in the surface mesh. The distance that we seek is then identified to be the minimum of the computed distances.

Define  $\Omega$  as the computational domain and let  $\Gamma_s$  be the boundary of the solid body. At any point  $\mathbf{X}^i$  of  $\Omega$ , the function  $\alpha$  corresponds to the signed distance from  $\Gamma_s$  and is

defined by:

$$\alpha(\mathbf{X}^i) = \begin{cases} -\min_{\mathbf{P} \in \Gamma_s} \|\mathbf{X}^i - \mathbf{P}\| & \text{for } \mathbf{X}^i \notin \Omega_s, \\ +\min_{\mathbf{P} \in \Gamma_s} \|\mathbf{X}^i - \mathbf{P}\| & \text{for } \mathbf{X}^i \in \Omega_s. \end{cases} \quad (5.2)$$

where  $\mathbf{P}$  refers to the projection of  $\mathbf{X}^i$  onto the boundary of the solid body  $\Gamma_s$  and  $\Omega_s$  the domain of this solid.

In other words, the distance is positive if the node is located inside the solid, negative outside it, and zero at the level of the interface. A schematic representation of the levelset function on a 2D hat shaped solid object is presented in figure 5.2.

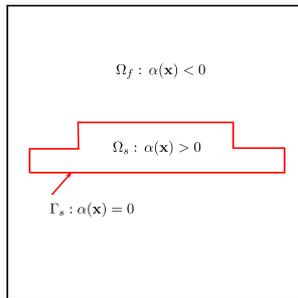


Figure 5.2 – Schematic representation of the levelset function for multi-domain problems.

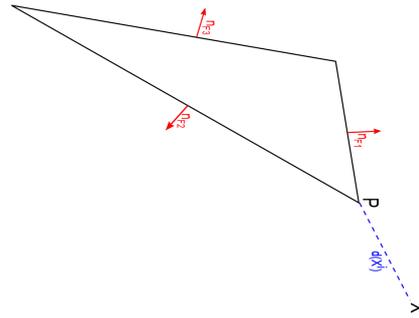


Figure 5.3 – Schematic representation of a node  $\mathbf{X}^i$ 's projection onto an element on the surface mesh.

Two main concerns arise at this level. The first one is related to the sign of the distance function, i.e. how to determine whether a point lies inside or outside the body. Second, as the computational cost induced by the calculation of the distances is linear with respect to the number of faces in the object's surface mesh, the CPU time needed for that process grows very fast and can yield unfeasible complex simulations. To deal with the latter limitation, we resort to a hierarchical representation of the surface mesh. The method consists in packing the elements in the surface mesh into large cells, each of which contains a fixed number of facets, and then evaluating the distances to these cells. The closest cell to the node is retained while the others are eliminated from computations. The selected cell is in turn divided into smaller ones with respect to which the distance computation is now performed. The process of packing and distance evaluation is repeated iteratively in order to determine the location of the projected node. The use of this strategy avoids checking the distances with respect to every face in the surface mesh thereby eliminating the cost induced by such a process. We will omit the details of the hierarchical method for brevity, the interested reader is referred to [Bruchon 09]. As for signing the distance, we define it with respect to the outward pointing normal  $\mathbf{n}_F$  to the closest face in the surface mesh. If the scalar product of the projection vector  $\overrightarrow{\mathbf{X}^i \mathbf{P}}$  by the normal  $\mathbf{n}_F$  is negative then so does the distance and vice versa. When the projection coincides with a mesh node (see fig. 5.3), the normal to the closest face is used that is if  $|\overrightarrow{\mathbf{X}^i \mathbf{P}} \cdot \mathbf{n}_{F_2}| > |\overrightarrow{\mathbf{X}^i \mathbf{P}} \cdot \mathbf{n}_{F_1}|$  then we investigate the sign of  $\overrightarrow{\mathbf{X}^i \mathbf{P}} \cdot \mathbf{n}_{F_1}$ . The algorithm and further details about signing the distance function are provided in [Bruchon 09].

Now that the signed distance has been computed, the fluid/solid interface  $\Gamma_s$  is given by the zero isovalue of the function  $\alpha$ :

$$\Gamma_s = \{\mathbf{X}, \alpha(\mathbf{X}) = 0\}. \quad (5.3)$$

The maximum absolute value of the levelset function is reached in the vicinity of the interface. It's quite useless and time-consuming to compute the signed distance function far away from the immersed object. The key idea in this work for defining the levelset function stems in reducing the computational time by reducing the computational domain. In other words, we construct a rectangular computational domain enclosing the object of interest and whose dimensions are a bit larger than the body itself. The signed distance function is then determined over this subdomain. In figure 5.1 we show a surface mesh of a missile (*center*) and the corresponding mesh that is well adapted to the geometry (*right*). The aim now is to define the signed distance function over the simulation's computational domain. This is achieved by interpolating the levelset function from the fitting domain that is constructed to fit the solid onto the global computational domain as depicted in figure 5.4. The interpolation process relies on the hierarchical packing method that rapidly identifies for each node in the fitting domain the corresponding element in the global domain. Since there will be nodes in the global domain where the levelset field is not assigned, we set its value at these nodes to its maximum (in absolute value) over the fitting domain. Clearly as these nodes are located outside the object then the sign of the levelset function is negative.

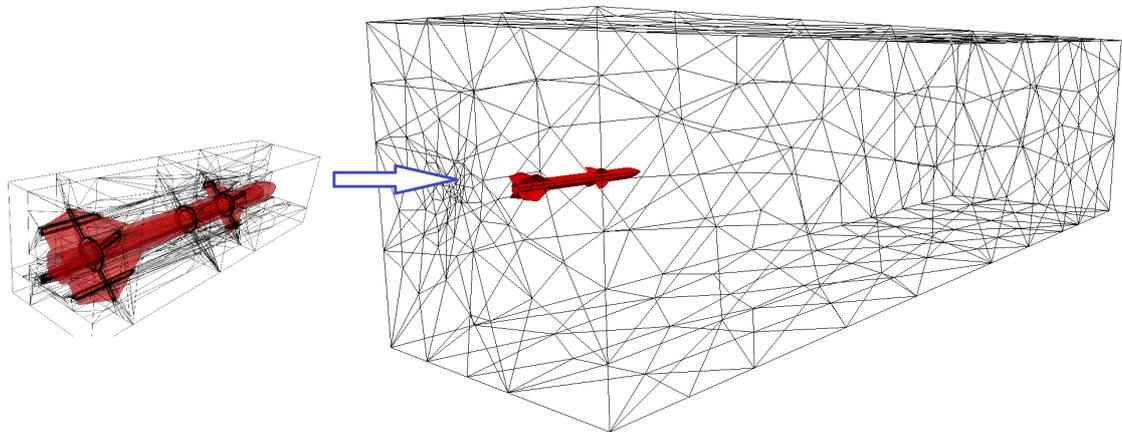


Figure 5.4 – Zero-isovalues of the levelset function of a missile object in the fitted domain (left) and the computational domain (right).

The advantage of this interpolation technique resides in its flexibility and rapidity. It permits the creation of several instances of the same object at different locations in the global domain. To do so, we just need to change the position of the fitting domain with respect to the global one. It allows, as well, moving the solid over the computations by simply modifying the position of the fitting domain.

As explained earlier, the signed distance function is employed to position the interface of the immersed object inside the global computational domain. This function

can also be used to initialize the desirable material properties on the single mesh representing the fluid and the solid parts as will be discussed in the following subsection.

It is important to mention that in the case of a geometry's immersion using its corresponding STL file, the quality and accuracy of the defined surface mesh limits the precision of the described immersed structure. Another approach for immersion is currently under development in the work of Veysset [Veysset 14b]. This method bypasses the generation of surface meshes by defining Non Uniform Rational B-Splines (NURBS) curves or surfaces, to represent simple or complex geometries. The distance function is now computed with respect to the NURBS functions and the interfaces of the immersed solids are defined as the zero-isovalues of these functions.

### 5.2.2 Second component: anisotropic mesh adaptation

In order to provide an accurate configuration of the physical problem, immersed objects should be properly defined and material properties should be well distributed. Difficulties might arise due to the discontinuities in the material properties between the solid and the fluid parts. Indeed, when the discontinuities intersect the mesh elements arbitrarily, they will not be properly captured and hence they might result in the deterioration of the solution's accuracy. Motivated by the lack of fields' fidelity at the level of interfaces that can severely limit the applicability of the immersed volume method, a mesh refinement around the interfaces stems as a good remedy to this issue.

The objective is therefore to combine the levelset representation with local mesh refinement around the zero isovalue of the levelset function yielding an accurate capture of a fluid-solid interface at a low computational cost. The key to these regards is anisotropic mesh adaptation that generates highly stretched and well oriented elements allowing a good capture of sharp gradients.

In the past few years, progress has been made in the CIMLIB library to build up an anisotropic mesh adaptation technique that is locally refined in the vicinity of the interface [Gruau 05, Coupez 09, Bernacki 08]. In what follows, we retrace the steps used prior to this work in order to construct an anisotropic mesh suiting this kind of applications.

We recall that the anisotropic mesh adaptation procedure can be interpreted as a metric construction problem where a metric is a symmetric positive definite tensor whose eigenvalues and eigenvectors can be used to prescribe respectively the mesh sizes and the directions along which these mesh sizes are prescribed. In the framework of mesh adaptation at the interface of an immersed object, the mesh shall be driven by the gradient of the levelset function

$$\mathbf{x} = \frac{\nabla \alpha}{\|\nabla \alpha\|}.$$

To ensure a good capture, small mesh sizes shall be imposed along the unit normal to the interface that is in the gradient direction while maintaining a fixed default or a

background size  $h_d$  in the orthogonal direction to the interface and in the zones that are far from it.

A likely definition for the mesh size evolution is:

$$h = \begin{cases} h_d & \text{if } |\alpha(\mathbf{x})| > \varepsilon/2 \\ \frac{2h_d(m-1)}{m\varepsilon}|\alpha(\mathbf{X})| + \frac{h_d}{m} & \text{if } |\alpha(\mathbf{X})| \leq \varepsilon/2 \end{cases} \quad (5.4)$$

This means that, at the interface, the mesh size is  $m$  times smaller than the background size  $h_d$ . As we get far away from it this size increases gradually till equaling  $h_d$  at a distance  $\varepsilon/2$  with  $\varepsilon$  being a predefined thickness.

The levelset's gradient direction  $\mathbf{x}$  and the mesh size  $h$  defined above, imply the following metric:

$$\mathcal{M} = C(\mathbf{X} \otimes \mathbf{X}) + \frac{1}{h_d^2} \mathbb{I} \quad \text{with} \quad C = \begin{cases} 0 & \text{if } |\alpha(\mathbf{X})| \geq \varepsilon/2 \\ \frac{1}{h^2} - \frac{1}{h_d^2} & \text{if } |\alpha(\mathbf{X})| < \varepsilon/2 \end{cases} \quad (5.5)$$

where  $\mathbb{I}$  denotes the identity tensor. This metric yields an anisotropic mesh near the interface, with a mesh size  $h$  in the levelset's gradient direction and  $h_d$  in the other directions, and an isotropic mesh far from the interface, with a constant mesh size  $h_d$  in all the directions. We present in figure 5.5(left) the mesh obtained using the metric  $\mathcal{M}$  constructed with the layer based approach. We can identify how the elements are anisotropic in the vicinity of the interface and how they become isotropic and increase progressively in size as we move away from the interface. The obtained mesh is made up of 4,000 nodes distributed over the domain with a higher concentration near the interface.

On the other hand, as discussed earlier, in order to provide a good capture of the object's interface, we need to create a sharp gradient in its surrounding. However, relevant information about the levelset function are condensed in a narrow layer around its zero-isovalue so as to determine the signed distance field, the gradient direction and refine the mesh in this zone. Outside this layer, one just needs to know whether a node is inside or outside the solid object. Therefore, it is not necessary to compute the levelset function far from the interface. For that reason, we only retain the information provided at a narrow layer around the interface by applying a hyperbolic tangent filter on the signed distance function as described in [Coupez 14]. The filtered function is then defined by:

$$\varphi = \varepsilon \tanh\left(\frac{\alpha}{\varepsilon}\right). \quad (5.6)$$

This filtering process yields a sharp gradient of  $\varphi$ . The parameter  $\varepsilon$  controls the steepness of the gradient and thus the anisotropy of the mesh in the vicinity of the interface. The smaller  $\varepsilon$ , the sharper the gradient of  $\varphi$  and hence the higher the stretching of the elements near the interface. The resulting function  $\varphi$  is continuous and differentiable,

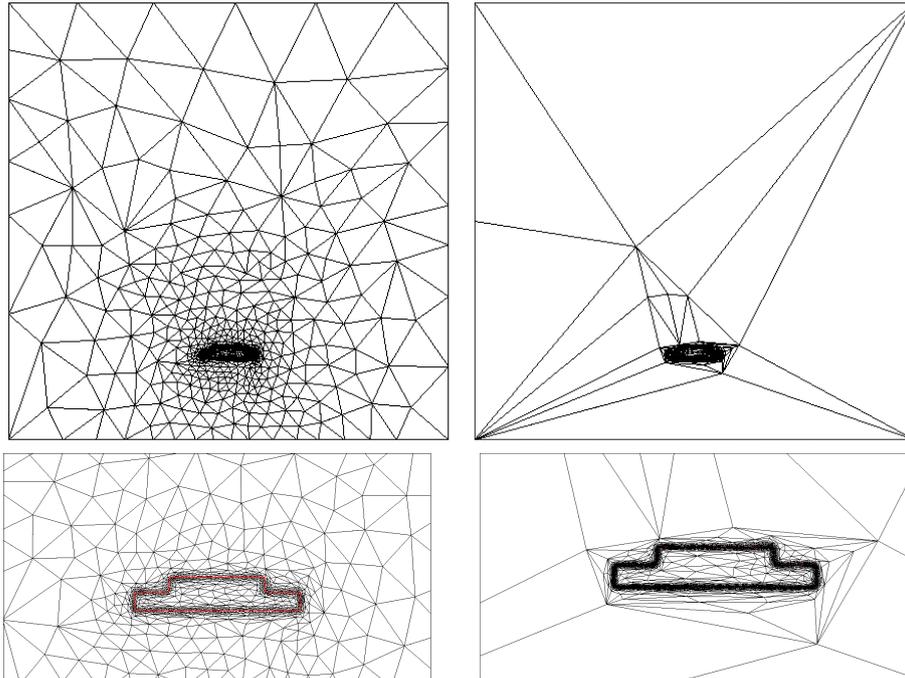


Figure 5.5 – Anisotropic meshes adapted on the immersed hat shaped solid obtained using the layer based approach (left) and the edge-based metric construction developed in this work (right).

and in the narrow layer around the interface, is equal up to the first order to  $\alpha$  and grows up to size  $\pm\epsilon$  outside the layer of size  $2\epsilon$ .

Although the anisotropic meshing technique driven by the metric defined in equation (5.5) seems simple and easy to employ, it does not provide a control on the total number of degrees of freedom in the constructed mesh. Therefore when complex industrial applications are involved, a drastic increase in the mesh complexity can be expected yielding a considerable computational cost.

In chapter 2, we have proposed a novel method for generating anisotropic meshes. Starting with an arbitrary mesh, the developed approach automatically detects the locations of sharp solution/function gradients from an a posteriori edge-based error estimator. At each node in the mesh, a metric tensor is constructed without prior information from the element, neither considering any underlying interpolation. The metric assigns optimal mesh sizes in the different directions issued from that node under a constraint of a fixed number of nodes. It is performed by introducing a statistical concept: the length distribution tensor that gathers information from all the edges at the node. Then the error is computed along and in the direction of each edge. These error estimates would then be used to prescribe stretching factors along the edges in the mesh. Taking into account the stretching of the edges, a modified metric tensor is constructed and in turn is given to the MTC mesher to generate an optimal mesh in terms of minimum interpolation error. The method has been theoretically and numerically validated and achieves second order convergence rates.

In the framework of the immersed volume method, the goal is to provide a good representation of the fluid/solid interfaces for a fixed number of nodes in the mesh.

The procedure consists in applying the mesh adaptation algorithm on the normalized filtered levelset function. Indeed, as the gradient of the levelset function has a unit norm over the mesh, the error is equi-distributed over the domain and consequently applying an anisotropic mesh adaptation based on this error would not lead to a sharp capture of the interfaces. In order to condense the nodes around the zero-isovalue of the levelset function, we resort to the filtered levelset function which exhibits strong gradients at that level.

We repeat the anisotropic mesh adaptation on the 2D hat shaped body using the new anisotropic mesh adaptation tools with the same number of nodes 4,000. We present in figure 5.5(right) the obtained mesh when adapting on the filtered levelset function. We can clearly see how stretched the elements are far from the interface as the gradient of the function does not vary at this level. The nodes are highly condensed in the vicinity of the interface favoring its accurate capture. This validates how the developed algorithm optimizes the distribution of the nodes to produce a sharp anisotropic mesh that is well adapted based on a given variable.

We move on now to evaluate the performance of the developed method on 3D complex geometries. Figure 5.6 depicts the surface plots of the levelset and the filtered levelset functions on a 2D horizontal cut along the x-axis and y-axis of the missile object. Clearly, the filtered levelset function provides a better description of the object at hand.

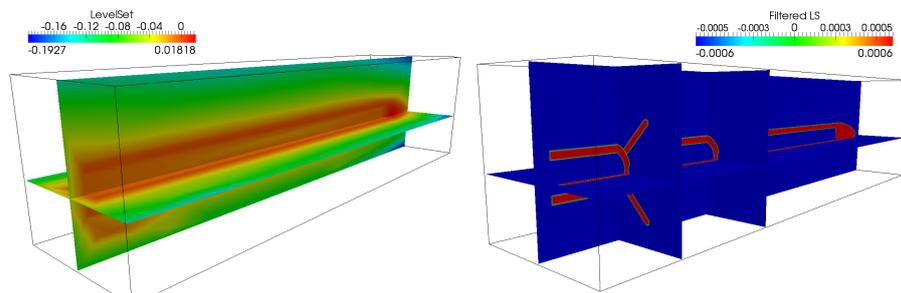


Figure 5.6 – Levelset (left) and filtered levelset (right) functions corresponding to an immersed missile object.

For illustration on the performance of the anisotropic mesh adaptation tool, figures 5.7 and 5.8 present different snapshots on the zero isovalues of the levelset functions corresponding to a missile object and a F1 car. Inhere, the meshes were adapted only on the filtered levelset function describing the geometry. The close-up snapshots depict the layers of elements near the objects boundaries. We can clearly detect the bandwidth across the interface where the elements are highly condensed. One can also observe the extremely stretched elements along the boundary of the immersed objects whereas the rest of the domain keeps almost the same background mesh size. Note that this background size can be controlled by the prescribed minimum value for the mesh sizes. These results reflect how well, for a fixed number of nodes (40,000 for the missile and 150,000 for the F1 car), the developed algorithm captures complex geometries in terms of curvatures, sharp angles and singularities.

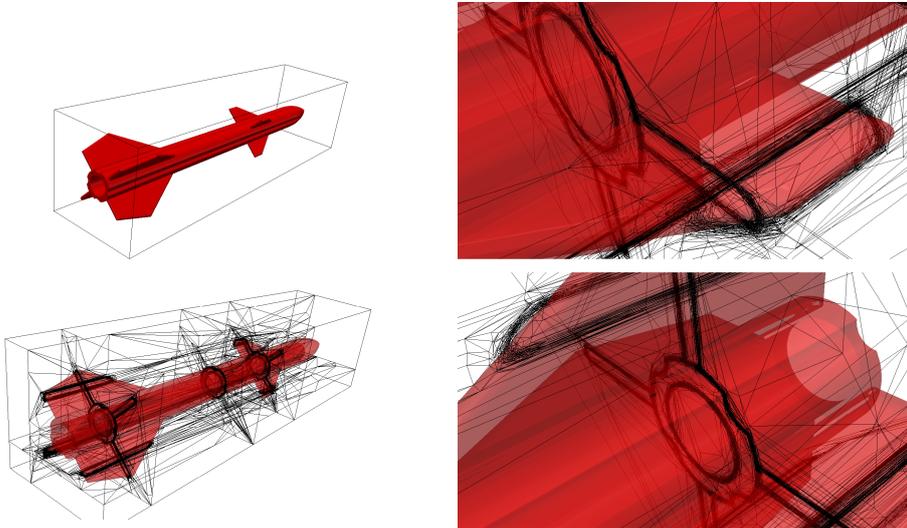


Figure 5.7 – Immersed missile object with its corresponding anisotropic mesh.

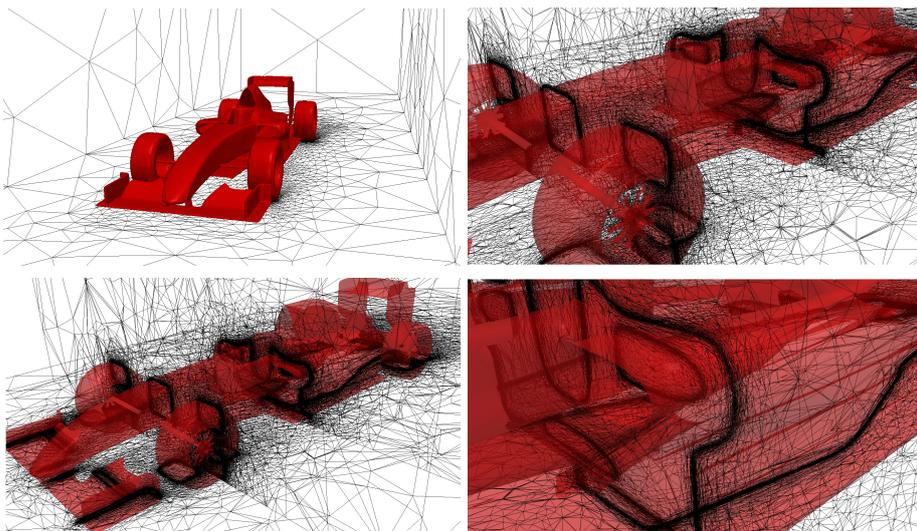


Figure 5.8 – Immersed F1 car with its corresponding anisotropic mesh.

Nevertheless, generating anisotropic meshes based only on the geometry of the immersed objects is not the optimal choice for CFD simulations. These meshes, by controlling the element sizes, stretching and orientation, will tend to generate fine elements in the surrounding of the fluid/solid interfaces and coarse ones far away. Consequently, they will ignore the characteristic features of the flow around the objects as well as the evolution of the thermo-dynamic properties. Hence, this choice of adaptation turns out to be insufficient and may yield an ill-conditioned stiffness matrix incapable of describing the heat transfers and the complex dynamics of the flow. Therefore, two possible attempts can be adopted to deal with this issue. The first one relies on imposing a relatively fine background mesh that can be obtained by prescribing a relatively large minimum mesh size. The second one consists in generating a mesh that is adapted to several fields of interest. This can be achieved by generating a metric tensor for every field and then applying a metric intersection procedure followed by a mesh generation. However this process necessitates a considerable computational cost. In this work, we propose another way to account for multi-components mesh adaptation.

The extension to deal with multi-component fields (tensors, vectors, scalars) is provided in the following section. Rather than considering several metric intersections, we propose an easy way to account for different fields (e.g. velocity norm, velocity components, levelset function, ...) in an a posteriori analysis while producing a single metric field. Note also that the proposed algorithm is implemented in the context of adaptive meshing under the constraint of a fixed number of nodes. With such an advantage, we can provide a very useful tool for practical CFD and FSI problems while preventing a drastic increase of mesh complexity and hence in computational cost. The generality, automatism and efficiency of the proposed method make it an important tool for the resolution of complex CFD applications.

### 5.2.3 Third component: mixing laws

Once the mesh has been well adapted around the zero-isovalue of the levelset function providing an accurate capture of the fluid/solid interface, we resort to the use of mixing laws in order to well distribute the material properties on the different subdomains. The assignment of the physical properties can be specified on the basis of the levelset function's sign by means of a Heaviside function given by:

$$H(\alpha) = \begin{cases} 1 & \text{if } \alpha \geq 0 \\ 0 & \text{if } \alpha < 0 \end{cases}. \quad (5.7)$$

However, sharp variations across the interface might be devastating for numerical solvers. To improve the stability and continuity of the numerical solution, the sharp discontinuities can be relaxed by artificially thickening the interface layer. This can be achieved by smoothing the transition of material properties along the interface. To do so, a smoothed Heaviside function is employed over a characteristic thickness of  $2\varepsilon$

[van der Pijl 05]:

$$H_\varepsilon(\alpha) = \begin{cases} 1 & \text{if } \alpha > \varepsilon \\ \frac{1}{2} \left( 1 + \frac{\alpha}{\varepsilon} + \frac{1}{\pi} \sin \left( \frac{\pi\alpha}{\varepsilon} \right) \right) & \text{if } |\alpha| \leq \varepsilon \\ 0 & \text{if } \alpha < -\varepsilon \end{cases} \quad (5.8)$$

with  $\varepsilon = O(h_d)$  being a thickness parameter for smoothing the Heaviside function and  $h_d$  the mesh size in the normal direction to the interface. In the numerical experiments we set  $\varepsilon = 2h_d$ . In the vicinity of the interface, the latter parameter can be determined according to the following expression:

$$h_d = \max_{j,l \in K} \nabla \alpha \cdot \mathbf{X}^{jl}, \quad (5.9)$$

where  $K$  is a mesh element in the tight sublayer around the interface and  $\mathbf{X}^{jl} = \mathbf{X}^l - \mathbf{X}^j$  is an edge of  $K$ .

The objective now is to benefit from the regularity provided by the smoothed Heaviside function defined in equation (5.8) to define the physical properties on one side or the other of the interface. When treating heat transfer and fluid flow problems, linear variation of the properties is prescribed for the density  $\rho$ , the initial temperature  $T_0$ , the dynamic viscosity  $\mu$ , the specific heat capacity  $C_p$  and the mean absorption coefficient  $\kappa$ , as follows:

$$\begin{aligned} \rho &= \rho_f H_\varepsilon(\alpha) + \rho_s (1 - H_\varepsilon(\alpha)), \\ \mu &= \mu_f H_\varepsilon(\alpha) + \mu_s (1 - H_\varepsilon(\alpha)), \\ \rho C_p &= (\rho_f C_{pf} H_\varepsilon(\alpha) + \rho_s C_{ps} (1 - H_\varepsilon(\alpha))), \\ \rho C_p T_0 &= \rho_f C_{pf} T_{0f} H_\varepsilon(\alpha) + \rho_s C_{ps} T_{0s} (1 - H_\varepsilon(\alpha)), \\ \kappa &= \kappa_f H_\varepsilon(\alpha) + \kappa_s (1 - H_\varepsilon(\alpha)), \end{aligned} \quad (5.10)$$

where the solid properties are referred to using the subscript 's' and the fluid ones using the subscript 'f'.

However, the use of linear distribution for thermal conductivity surrenders to abrupt changes along the interfaces and yields inaccurate results as pointed out in [Patankar 80, Batkam Hemo 02, Hachem 09]. Harmonic variations were proposed to ensure the conservation of the heat flux:

$$\lambda = \left( \frac{H_\varepsilon(\alpha)}{\lambda_f} + \frac{1 - H_\varepsilon(\alpha)}{\lambda_s} \right)^{-1}. \quad (5.11)$$

Figure 5.9 depicts the distribution of the density over the domain, obtained using the above defined mixing law, on a gradually refined mesh made up of 4,000 nodes obtained using the layer-based approach and the one obtained for the same mesh res-

olution using the developed anisotropic mesh adaptation. We can clearly see how precisely the transition from the solid subdomain to the fluid one is captured using the anisotropic meshing tools. We note that the sharp discontinuity of the material properties across the interface, if not properly represented, can affect the quality of the numerical solution.

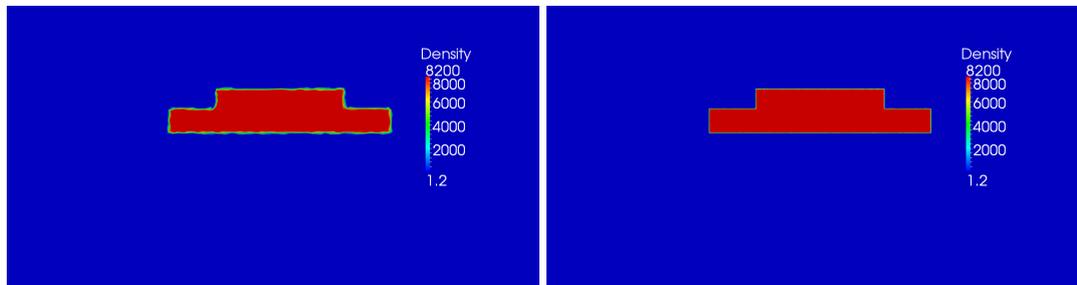


Figure 5.9 – Density distribution in a multi-domain simulation on adapted meshes of 4,000 nodes based on a gradual refinement (left) and on an anisotropic refinement (right).

We recall that the fluid/solid interface is no longer an explicit geometric boundary of the domain and consequently one cannot impose boundary conditions (such as radiative heat transfers) at its level. The alternative proposed by the IVM approach accounting for heat transfers is to employ a monolithic formulation of the multi-materials problem whereby a single set of equations is solved. The coupled problem involves simultaneously resolving the momentum equations, energy equation, the turbulent kinetic and dissipation energy equations, and the radiative transport equations over the entire domain with variable material properties.

The power of the proposed direct thermal coupling analysis resides in the natural treatment of the heat transfers without resorting to experimental investigations in order to determine heat transfer coefficients. Obviously the computation of these coefficients requires repeating the whole experiment at every change in the material properties or in the setup of the problem which can rapidly become unfeasible, especially when complex simulations with complicated geometries are involved. The immersed volume method can be applied to any heat transfer problem and the only needed parameters are the physical properties of the different materials.

When a fixed mesh is maintained all over the computations, the immersed volume method is applied only once prior to the simulation. The associated computational cost, due to the calculation of the levelset function and the initial mesh generation is relatively small with respect to the total cost of the simulation. However, if the mesh has to be dynamically adapted over time, then the IVM algorithm shall be applied at every update of the mesh. This will induce a considerable cost related to the computation of the levelset function, the redistribution of the material properties, and the anisotropic well adapted mesh. In complex fluid/structure interaction applications, mesh adaptation is considered as a powerful tool for improving the accuracy of the numerical results and accelerating the computations. However, adapting the mesh only on the levelset function is not sufficient to provide a good capture of the physical phenomena

that are involved. The objective of the adaptive methods is to lower the gap between the real physical problem and the numerical one. This is achieved by increasing the grid's resolution at the locations of solution discrepancies in the aim of reducing the estimated errors. Therefore the mesh shall be adapted on the different features of the problem including the immersed objects' interfaces. This concern rises the need to construct a single metric tensor that accounts for different error indicators which will be discussed in the following section.

### 5.3 MESH ADAPTATION BASED ON SEVERAL FIELDS

When simulating complex physical phenomena involving turbulent flows and heat transfers, it is highly desirable to accurately capture all the characteristics of the problem, including but not limited to, the flow field, the temperature variations, and the fluid/solid interfaces. For instance when simulating turbulent flows, the boundary layer and the flow separation need to be modeled accurately. Two strategies are generally adopted: the 'a priori' and 'a posteriori' adaptations. The former approach departs from a mesh that is pre-adapted around the boundaries, the main vortices and in the wake region. However to design such a mesh, a priori assumptions on the profile of the solution shall be considered (i.e. Reynolds number,  $y^+$  value, ...). In this case the mesh adaptation criteria are geometric and do not evolve with the solution. In the second approach, the mesh adaptation problem is cast as a dynamic optimization one suited to control the interpolation error over time. The anisotropic decisions are driven by 'a posteriori' error estimates on the desired indicator fields. In the framework of the anisotropic mesh adaptation technique developed in chapter 2, the objective of the error estimates is to survey the spatial variations of the fields of interest over the mesh edges. The gathered estimates will then be synthesized into a metric tensor that will incorporate information about the locations of anisotropic refinement. Finally the optimal mesh is generated by a single metric tensor based on the control of the desired sensors' interpolation errors.

#### 5.3.1 Metric intersection

The common technique to derive a single metric at each node in the mesh accounting for several fields of interest relies on the rigor of computing the metrics corresponding to each of the sensor fields then performing a metric intersection operation [Alauzet 10a]. The constraint yielding the smallest mesh size imposed in each direction by the set of metrics is maintained. The geometric interpretation of the intersection operator corresponds to the largest ellipsoid contained in the intersection of the ellipsoids associated with the concerned metrics. At a node  $\mathbf{X}^i$  in the mesh, when two metrics  $\mathcal{M}_1$  and  $\mathcal{M}_2$  shall intersect, a simultaneous reduction is performed to ensure a common diagonalizable basis for the metrics. According to [Alauzet 10a] the basis is made up from the normalized eigenvectors  $e_i$  ( $i=1,2,3$  in  $3d$ ) of the metric  $\mathcal{N} = \mathcal{M}_1^{-1}\mathcal{M}_2$ .

The corresponding diagonal components associated to  $\mathcal{M}_1$  and  $\mathcal{M}_2$  in this basis are determined by:

$$\lambda_i = e_i^t \mathcal{M}_1 e_i \quad \mu_i = e_i^t \mathcal{M}_2 e_i \quad \text{for } i = 1, 2, 3. \quad (5.12)$$

The metric reflecting the intersection is thus given by:

$$\mathcal{M}_{1 \cap 2} = \mathcal{R}^{-t} \begin{pmatrix} \max(\lambda_1, \mu_1) & 0 & 0 \\ 0 & \max(\lambda_2, \mu_2) & 0 \\ 0 & 0 & \max(\lambda_3, \mu_3) \end{pmatrix} \mathcal{R}^{-1}. \quad (5.13)$$

where  $\mathcal{R}$  is a rotation matrix whose columns are the eigenvectors of the metric  $\mathcal{M}_{1 \cap 2}$ . In practice, following the lines in [Alauzet 10a], the eigenvalues of  $\mathcal{N}$  are computed using a Newton algorithm and accordingly the eigenvectors are deduced. The metric tensor representing the intersection of  $k$  metrics is defined by:

$$\mathcal{M}_{(1 \dots k)} = \mathcal{M}_{(((1 \cap 2) \cap 3) \dots) \cap k)}. \quad (5.14)$$

However as the intersection operation is not commutative, the resulting mesh is not unique and depends on the order of intersection when the number of intersected metrics is higher than 2.

An alternative for finding the intersection of  $k$  metrics is proposed in [John 48, Loseille 09] and consists in resolving the following optimization problem:

$$\mathcal{M}_{\cap_i M_i} = \underset{\mathcal{M}}{\operatorname{argmin}} \ln \left( \det(\mathcal{M})^{-1} \right), \quad (5.15)$$

under the constraints that the ellipsoid associated with  $\mathcal{M}_{\cap_i M_i}$  is contained in each of the ellipsoids of the  $k$  metrics. This solution is computationally more expensive but produces a unique metric tensor.

### 5.3.2 Multi-components metric construction

In this work, we omit the metric intersection process and we construct directly one metric that accounts for several variables [Coupez 13a, Coupez 13b]. Based on the theoretical analysis conducted in the previous chapter, we extend the scalar field representing the edge error estimation of a single variable into a vector whose components are the error estimates on the different variables under consideration. That is, instead of considering each variable at a time, constructing its corresponding metric tensor then intersecting the different metrics, we perform the analysis on a vector field  $\mathbf{u}$ , recover its gradient then estimate the corresponding vector of error estimates on the edges of the mesh. The vector field components are normalized to give equal weights to the different variables. An example of such a vector, at a node  $X^i$ , can be:

$$\mathbf{u}(X^i) = \left\{ \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|}, \frac{\|\mathbf{v}^i\|}{\max_j \|\mathbf{v}^j\|}, \frac{\alpha}{\max(\alpha)}, \dots \right\},$$

where  $\mathbf{v}$  is the velocity field,  $\frac{\|\mathbf{v}^i\|}{\max_j \|\mathbf{v}^j\|}$  refers to the normalized velocity magnitude, and  $\alpha$  denotes a scalar field (filtered levelset function, temperature, turbulent viscosity,  $\dots$ ). Consequently, the adaptivity accounts for the change of direction of the velocity, its magnitude and the variation of the scalar  $\alpha$ . The resulting vector of error estimates is:

$$\mathbf{e}_{ij} = \{e_{ij}^1, e_{ij}^2, \dots, e_{ij}^n\}. \quad (5.16)$$

Therefore, by defining a suitable  $L_p$  norm  $\|\cdot\|_p$  the corresponding stretching factors become:

$$s_{ij} = \left( \frac{\|\mathbf{e}(N)\|_p}{\|\mathbf{e}_{ij}\|_p} \right)^{\frac{1}{2}}. \quad (5.17)$$

To illustrate on the performance of the multi-criteria metric construction, let us consider 4 geometric objects (2 circles and 2 squares) and adapt the mesh on the iso-zero value of their levelset functions. Figure 5.10 depicts the meshes obtained after 10 adaptive iterations, using the metric intersection, based on the simultaneous reduction method (left) and the multi-components method developed in this work while adapting on the  $L_2$  norm of the edge error vector field (right). In both cases, 4,000 nodes were prescribed to drive the mesh. It is clear from the obtained meshes that the mesh converges faster when using the developed multi-criteria adaptation. The elements are well localized over the regions of high gradients permitting a good capture of the geometries' isovalues. In the case of metric intersection, the nodes in the mesh are concentrated at the zones of geometrical intersections providing an important precision at this level while a weaker capture is associated to the other locations of steep gradients. We have reported in table 5.1 the needed time to perform 10 mesh adaptations on the geometries and the number of adaptive iterations to converge the mesh to the optimal one. We can clearly identify the advantage of the proposed multi-component metric construction. Implementation-wise, the method is simpler as it requires only to fill in a vector of several components and compute its corresponding gradient field.

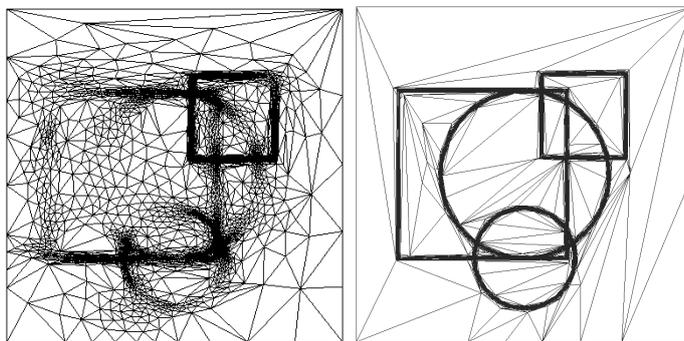


Figure 5.10 – Anisotropic mesh adaptation around different shapes obtained after 10 adaptive iterations using the metric intersection (left), the multi-components metric construction based on the  $L_2$  norm of the edge error (right).

|                                     | Metric intersection | Multi-criteria |
|-------------------------------------|---------------------|----------------|
| CPU time (s)                        | 34.65               | 24.77          |
| # of iterations to mesh convergence | 35                  | 10             |

Table 5.1 – Statistics on the CPU time and the number of mesh adaptations to converge the mesh toward optimality using the metric intersection and the multi-criteria techniques.

We move on now to highlight the importance of adapting the mesh on several fields of interest, we reconsider the 2D driven cavity problem described in the previous chapters. The computations were performed on the domain  $[0,1] \times [0,1]$ . A fluid motion inside the cavity is driven by shear forces resulting from a moving top wall at a velocity of  $1m/s$  and no-slip conditions on the other walls. The density of the fluid is set to  $1kg/m^3$  and its dynamic viscosity to  $2 \times 10^{-4}Pa \cdot s$  yielding a Reynolds number of 5,000. We have conducted the same test case while adapting the mesh on the norm of the velocity field and on the combination of the norm and direction of that field using 10,000 nodes. As shown in figure 5.11, the normalized velocity field reflects better the locations where the velocity field sustains strong variations. The corresponding anisotropic meshes are depicted in figure 5.12. We can clearly identify that the anisotropic mesh adaptation accommodating for the intensity of the velocity field and its changes of directions allows a better capture of the emerging small vortices, the boundary layers and detachment points.

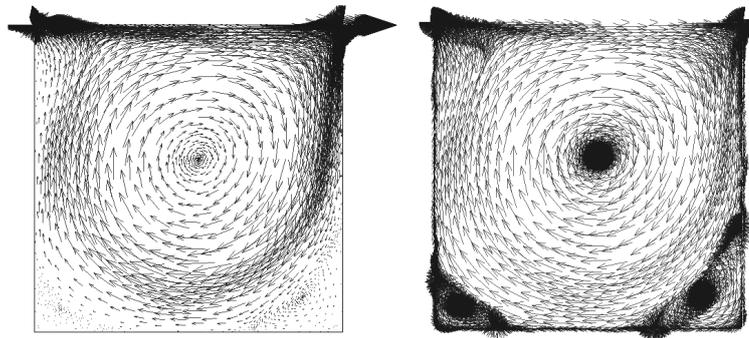


Figure 5.11 – Normalized velocity field (left) the velocity vector (right).

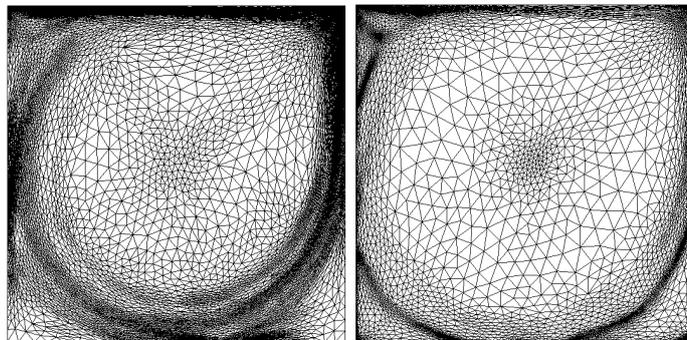


Figure 5.12 – Anisotropic mesh obtained when adapting on the velocity norm (left) and on its norm and direction (right).

**Remark 38.** There are two possibilities for defining mesh adaptation criteria. The first one, ‘sensor based’, consists in minimizing the interpolation error for selected solution fields (such

as the levelset function, the temperature field, the velocity norm, and the velocity direction, etc.). This technique for deriving the anisotropic mesh is independent from the problem at hand. It tends to generate an optimal capture of the desired fields but does not provide a full representation of the equations being solved. The sensor based choice for adaptation seems interesting from a computational standpoint as it does not require a data analysis before mesh adaptation. Moreover, it does not necessitate an a priori analysis of the equations and parameters at hand. The second criterion for anisotropic mesh adaptation focuses on the problem being solved and not on particular sensor fields. This approach is applied when the objective is to get as close as possible to the equations at hand. It is known as 'problem oriented' mesh adaptation [Alauzet 12, Belme 12, Venditti 02, Venditti 03] and intends to derive the anisotropic mesh that optimally reflects the features of the problem. The approach consists in performing an a priori analysis based on a goal oriented rationale and controlling the error with respect to a specified output functional. This method takes into account the error propagation along the computational domain through the resolution of the dual, or adjoint functional. Several techniques for reducing the error correction of the output function can be found in the literature, the interested reader can consult the following references [Dwight 08, Müller 01, Venditti 02, Venditti 03].

As the objective of this work is to conduct anisotropic adaptation decisions entirely from the a posteriori error estimates without resorting to a priori assumptions nor resolving an additional problem, we employ a "sensor based" adaptation technique.

In the following section, we are interested in evaluating the performance of the immersed volume method associated with a dynamic anisotropic mesh adaptation in resolving two and three dimensional physical problems. We recall that prior to this work, the IVM was applied once, in the problem's preparation phase, and the anisotropic mesh was generated based only on the levelset function and respecting a fixed small background mesh size. The resulting mesh that is well adapted to the immersed objects' interfaces was maintained all over computations. Clearly, the grid was not adapted to the characteristic features of the physical phenomena (e.g. vortices, boundary layers, temperature gradients) whose locations are not known a priori. Therefore, in order to obtain a good level of precision, a fine mesh shall be considered over the whole domain resulting in a high execution time and memory storage. In the context of problems involving a moving geometry, all the region traversed by the solid needs to be refined in order to maintain a sharp capture of the moving interface. This would require a clustering of elements in this zone of the domain and consequently will induce a considerable computational cost. Another problematic arises when the path of the solid is not known prior to the simulation. As it will be shown by the numerical results, the use of dynamic mesh adaptation overcomes these difficulties. The mesh will automatically follow the moving interface ensuring its accurate capture without drastically increasing the grid resolution and consequently the simulation's CPU time.

## 5.4 NUMERICAL VALIDATION

We have conducted several numerical experiments in the view of validating the IVM with a dynamic anisotropic mesh adaptation. In this section, we will start by showing that the IVM compares well with the regular body fitted method. We move on next to present the results obtained using the approach on a 2D forced convection problem. Then the performance and scalability of the adaptive procedure will be evaluated on 3D simulations with complex geometries.

### 5.4.1 Flow behind a circular cylinder

We reconsider the same test case presented in Section 4.3.4 and apply it in an IVM framework. This example consists in the study of a flow past a circular cylinder and was taken from [Schäfer 96]. It has been tested by many authors [Volker 04b, Volker 01, Frochte 09] using the body fitted technique. In this application, we intend to compare the results obtained with the IVM to these references and to our results obtained using a body fitted approach. We conduct two simulations with exactly the same setup using the anisotropic mesh adaptation and time adaptive technique. The number of nodes in the mesh was fixed to 20,000 and the mesh was adapted on the filtered levelset function and the velocity's norm and direction every 5 time-steps. Figure 5.13 depicts the velocity field and adapted mesh at time  $t = 6s$  obtained using the IVM and the body fitted approach. One can clearly identify the sharp representation of the immersed cylinder and the good capture of the boundary layers and vortex shedding in the former case. It can be detected from the close-up snapshots around the cylinder how the elements are symmetrically and isotropically distributed in that region which agrees with the symmetric profile of the velocity gradient field and hence with the error estimations. Anisotropic well oriented elements are generated away from the cylinder following the direction of the velocity field. From a qualitative point of view, both techniques are capable of rendering a good representation of the flow characteristics and generate a well adapted anisotropic mesh. Moreover, we point out the presence of mesh elements inside the cylinder, typical of the immersed volume approach. However, as the velocity field does not change in this region, these elements are very few, and consequently the computational cost due to the additional degrees of freedom is not significant.

To give a quantitative comparison, the evolutions of the drag and lift coefficients over time have been studied and the results are reported in figure 5.14. To generate a reference solution, we apply the body fitted approach on a fixed mesh of around 120,000 nodes and a time-step size  $\Delta t = 0.0025$ . A slight difference between the results can be spotted. Compared with the reference solution provided in [Schäfer 96], the IVM yields a better capture of the drag and lift coefficients than the body fitted technique (BF). Table 5.2 summarizes the statistics on the maximum drag and lift coefficients obtained using a body fitted method on a fixed mesh of around 120,000 nodes and a time-step size  $\Delta t = 0.0025$ , the body fitted approach and the IVM with

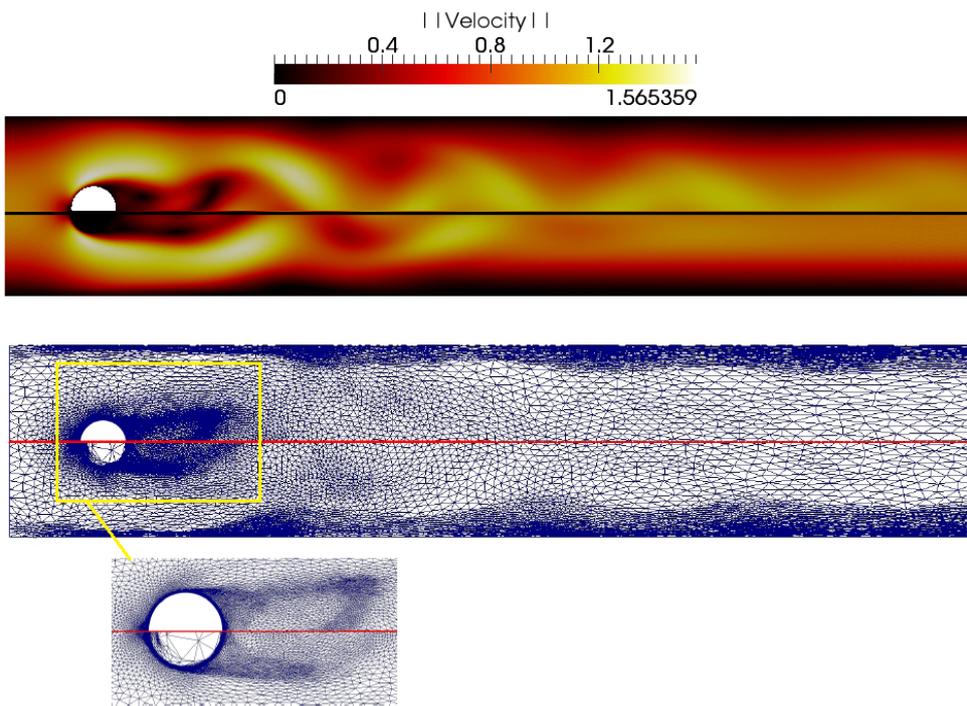


Figure 5.13 – Flow behind a circular cylinder: comparison of the velocity profiles (top) obtained using the body fitted and the IVM and their corresponding meshes (bottom) at time  $t = 6s$ .

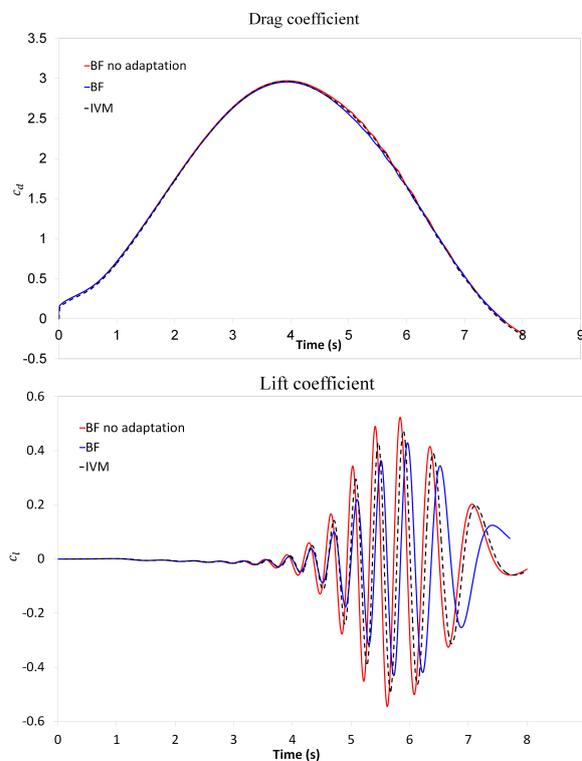


Figure 5.14 – Flow behind a circular cylinder: time history of the drag (top) and lift (bottom) coefficients.

|                                  | $c_{d,max}$     | $c_{l,max}$     |
|----------------------------------|-----------------|-----------------|
| Schäfer & Turek [Schäfer 96]     | $2.95 \pm 0.02$ | $0.48 \pm 0.01$ |
| Schäfer & Turek FEM [Schäfer 96] | 2.92            | 0.491           |
| Volker [Volker 04b]              | 2.95            | 0.382           |
| Present work (BF no adaptation)  | 2.9703          | 0.523           |
| Present work (BF)                | 2.9562          | 0.428996        |
| Present work (IVM)               | 2.9605          | 0.48215         |

Table 5.2 – Statistics on the maximum drag and lift coefficient values.

|                              | CPU time (s) |
|------------------------------|--------------|
| Present work (no adaptation) | 83201        |
| Present work (Body fitted)   | 16442        |
| Present work (IVM)           | 17307        |

Table 5.3 – CPU time needed for the simulation of a flow behind a circular cylinder.

anisotropic mesh and time adaptivity ( 20,000 nodes). All these results are compared with reference values found in the literature. The second row in the table reports the data in [Schäfer 96] obtained using a P2/P1 (Taylor-Hood) finite element method on unstructured grid made up from 21,508 degrees of freedom and a time-step 0.005. The reference values taken from [Volker 04b] correspond to an implicit time-marching scheme and the  $Q_2/P_1^{\text{disc}}$  finite element formulation on a mesh made up of 159,744 nodes and a time-step  $\Delta t = 0.00125$ . When using a body fitted approach, the failure to achieve the reference values especially for the lift coefficient can be attributed to the use of a first order time discretization as compared to the second order scheme employed in the reference paper. These discrepancies were also highlighted in [Volker 04b] when using a first order scheme. In the wake region behind the cylinder as shown in figure 5.13, the elements are well oriented and highly condensed in the vicinity of the interface providing a good level of accuracy and achieving the reference values.

As for the execution time needed to perform these simulations, table 5.3 provides statistics on the CPU time needed by the body fitted approach and the the immersed volume method with space and time adaptations. Compared with the solution obtained without adaptation, both adaptive approaches tested in the present work show considerable gain in efficiency. We notice that the body fitted approach is faster than the IVM. It is worth mentioning that the reported CPU time for the body fitted method did not account for the pre-simulation mesh generation phase. In this test case, the immersed solid has a simple geometry and hence this phase requires a negligible time. However, when the immersed object's geometry is complicated, the needed time for generating the initial mesh will be important and shall be accounted for.

**Remark 39.** It was reported in [Schäfer 96] that the solution obtained using a P2/P1 (Taylor-Hood) finite element method on unstructured grid made up from 21,508 degrees of freedom and a time-step 0.005 required 44000s on an 8.3 Linpack1000 machine, however comparison with our work can not be made due to the differences in computer hardware.

### 5.4.2 Turbulent flow behind a F1 racing car at 300km/h

In this test case, we aim at evaluating the performance of the IVM combined with anisotropic adaptation and stabilized flow solvers in resolving complex fluid-structure interaction problems. It is also employed to test the limit of the developed VMS solver. Indeed, beyond 300km/h, an extension of the solver to deal with compressible flows is needed. For that purpose, we consider a turbulent flow past an immersed highly sophisticated shape of a modern Formula One (F1) car (of dimensions  $6 \times 1.31 \times 2.07$ ) with several deflectors and additional winglets essential for enhancing the aerodynamic efficiency and the performance. The computational domain is a plane channel of dimensions  $20 \times 4 \times 6$ . The 3D computations have been obtained using 64 2.4Ghz Opteron cores. This example can be considered as the three-dimensional extension of the previous one with a complex geometry. It clearly highlights the utility and the advantages of the proposed immersed computational framework (Immersing  $\rightarrow$  Adapting  $\rightarrow$  Solving) as it points out the ease of changing/modifying/adjusting/adding geometrical details to the immersed car and study their impact on the car's performance. Each new immersed appendix may be represented easily by a new levelset function. If a body fitted mesh was to be used for this kind of simulation then the mesh construction phase will take a significantly important time. On the other hand, in the case of a moving geometry, this time will be even more expensive.

From a design and performance optimization standpoint, an important motivation for modelling the flow behind the F1 car is the capability to study the aerodynamic forces, improve the shape, brake/engine cooling, and control the wake structures. In here, we will not conduct such an analysis as the aim is only to show the capability of the developed IVM together with the space and time adaptations and the available stabilized solvers to perform such a study.

An inlet air velocity of  $300 \text{ km/h}$  was imposed on the left wall and no-slip conditions were prescribed at the other walls and the F1 car. We start the computations with an isotropic initial mesh made up of 4 million elements. Preserving half of this density, the mesh was then dynamically and anisotropically adapted every 10 time-increments based on the filtered levelset function, the velocity norm and directions. Figures 5.15 and 5.16 depict the velocity streamlines and pressure profile's evolution over time at different regions of the domain. The snapshots reflect the complex nature of the flow that is highly manipulated by the front wings. A high fidelity solution is needed to ensure an accurate representation of the wake at the level of the front wing, rear wings, sidepods and other small vertical winglets added as vortex generators in order to boost the boundary layers and increase the velocity streams. Clearly the wakes are highly dependent on the vehicle model that is being studied. A complicated bluff body aerodynamic scenario is identified, where vortices of different sizes emerge and re-energize the flow motion.

The structures of the generated anisotropic meshes (figure 5.17) reflect how well they capture the anisotropy of the solution. One can see how the mesh is dynamically

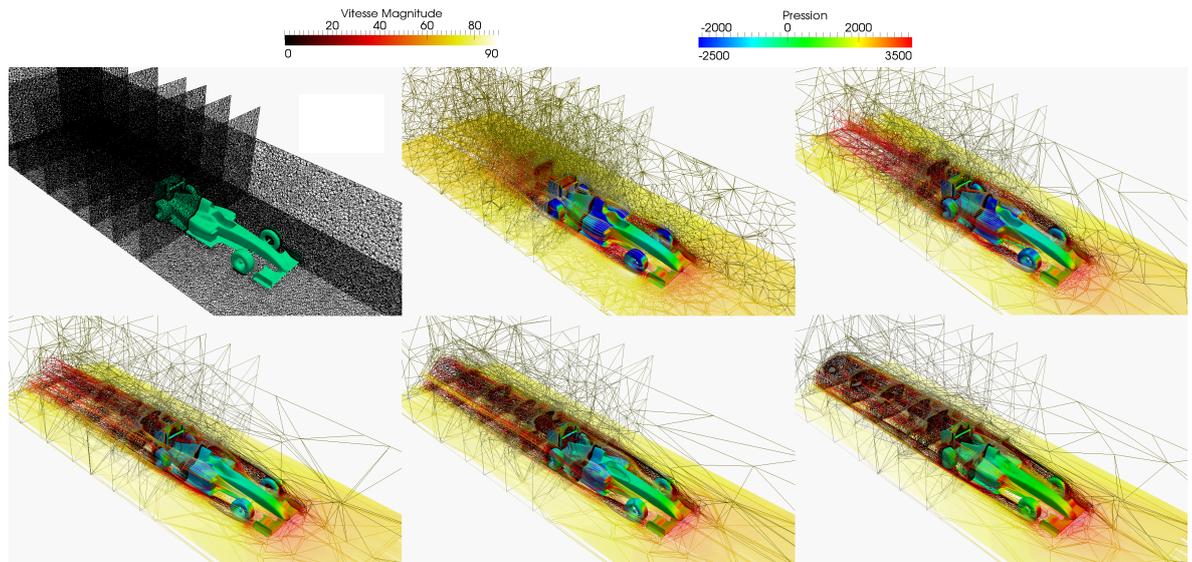


Figure 5.15 – Flow behind a F1 racing car: Velocity streamlines, pressure field and their corresponding dynamically adapted meshes at different time instances.

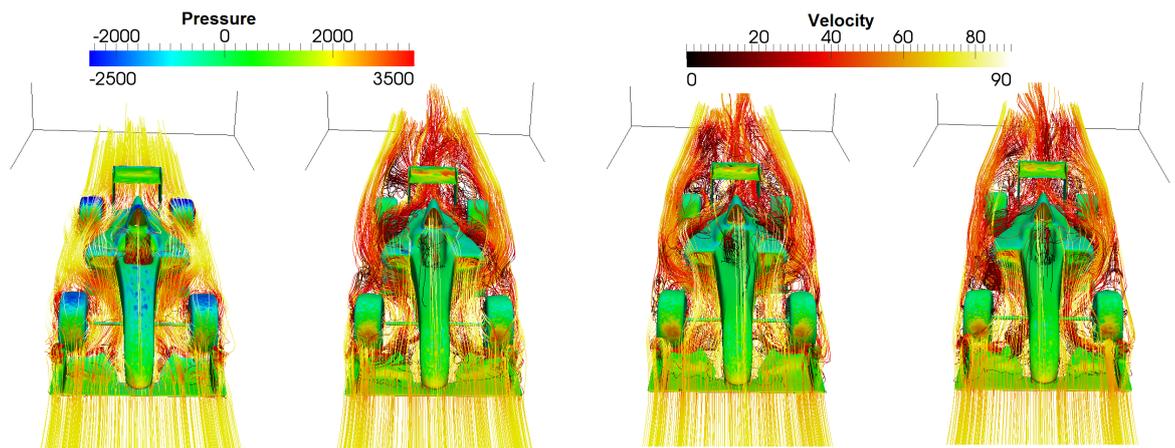


Figure 5.16 – Frontal view on the flow behind the F1 racing car at different time instances.

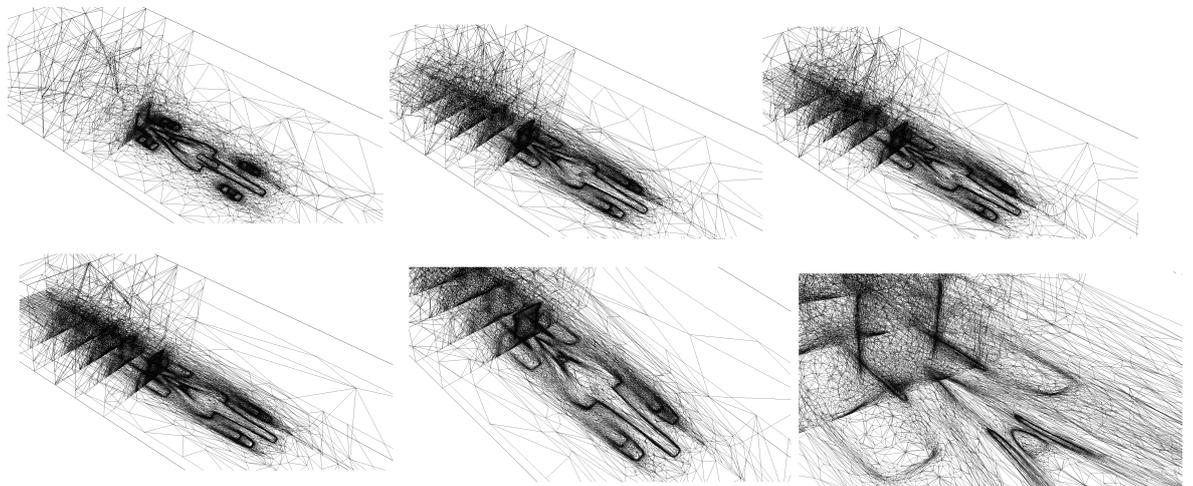


Figure 5.17 – Anisotropic meshes describing the characteristics of the flow behind the F1 racing car.

and progressively evolving as the flow develops. With time, a sharp layer of elements delimits the mushroom shape of the wake behind the geometry and yields a good precision at that level. Note also the high resolution that is following the development of the boundary layers and the detachments behind the tires and wheels. This reflects well the anisotropy of the solution caused by the discontinuity of the boundary conditions and the nature of the flow. Indeed, taking a closer look at the mesh near these interfaces, we can detect the good orientation of the elements with the stretching in the right direction. Although it is difficult to observe, a Venturi effect, closely related to the underground aerodynamics, develops absorbing a good amount of mesh nodes. In the far gradients' zone, the mesh is being systematically and automatically de-refined and the nodes are being redistributed over the domain. This demonstrates the ability of the algorithm to work under the constraint of a fixed number of nodes and to effectively control the element sizes, orientations and locations. If a pre-adapted fixed mesh were to be used over the whole simulation, then a mushroom shaped box that is highly refined needs to be maintained behind the geometry as well as along the wing, tires, sidepods and other locations of interest. Therefore isotropic fixed mesh is far from the optimal choice for efficient computations.

### 5.4.3 Forced turbulent convection

We aim in this numerical example at assessing the performance of the IVM in simulating conjugate heat transfer and turbulent air flow inside a 2D cavity. This test case can be viewed as a simplified model of a gas fired furnace. We consider a pre-heated rectangular domain  $[0, 23] \times [0, 12]m^2$  where we immerse two steel alloy bodies at positions  $(5, 0)$  and  $(13, 0)$ . For illustration purposes on the multi-criteria feature of the proposed adaptation algorithm, we assume constant material properties and focus only on the dynamic adaptation and response to the changes in the characteristics of the physical problem. The material properties affected to the fluid and the solid bodies are reported in table 5.4. We note that in this example the radiation effects are neglected. The initial temperature of the domain is set to  $250^\circ C$  and that of the solids to  $50^\circ C$ . Heated air is pumped into the enclosure through an inlet of size  $0.7m$  positioned at  $(0, 2)$  with a velocity of  $2m/s$  and a temperature of  $1000^\circ C$ . The air is vented out of the domain through two outlets of size  $0.7m$  centered at  $(23, 4.5)$  and  $(23, 10.5)$ . A schematic representation of the domain is provided in figure 5.18. Adiabatic temperature and slip boundary conditions are prescribed on the other walls. Before starting the computations, a levelset function is computed to define and position the immersed solids. A subsequent anisotropic mesh adaptation on the filtered levelset function is applied permitting a precise capture of the solid interfaces. The rest of the domain is discretized while respecting a fixed background size. Now, that the mesh has been created, we distribute the material properties according to equation 5.10 where we handle the sharp discontinuities in the thermal conductivities along the solid/air interfaces using harmonic mean formulation. It is important to mention that a high kinematic viscosity is

| Properties                                            | Air          | Steel |
|-------------------------------------------------------|--------------|-------|
| Density ( $\text{kg}/\text{m}^3$ )                    | 1.25         | 7850  |
| Specific heat ( $\text{J}/\text{kg} \cdot \text{K}$ ) | 1000         | 1499  |
| kinematic viscosity ( $\text{Pa} \cdot \text{s}$ )    | $1.95e^{-5}$ | -     |
| Conductivity ( $\text{W}/\text{m} \cdot \text{K}$ )   | 0.0262       | 48.6  |

Table 5.4 – Material properties of the air and the steel alloy solids.

assigned to the solid part and yields a negligible velocity field at the level of the solid bodies. Consequently, the energy equation reduces to a conduction equation at that level as the convection term vanishes. Moreover, in order to preclude possible numerical oscillations incurred by the sharp gradient in the temperature field at the locations of high convection, the SCPG stabilization is used. On the other hand, the stabilization of flow inside the enclosure is accounted for by using the variational multiscale (VMS) approach. As the Reynolds number corresponding to this simulation reflects a turbulent behavior of the airflow, a turbulence model will therefore be used. We solve the Reynolds-averaged Navier-Stokes problem with the  $k - \epsilon$  model to account for the turbulence inside the cavity as explained in chapter 1.

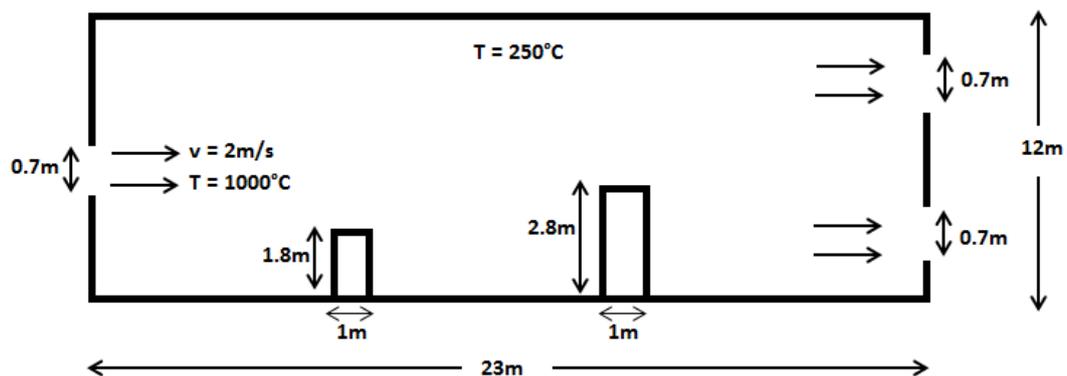


Figure 5.18 – Forced convection: sketch of the geometry and initial conditions.

Along the simulation time, the mesh is adapted on the  $L^2$  norm of the interpolation error of the temperature field, the filtered levelset function, the velocity norm and the velocity direction using 20,000 nodes.

The temperature distribution over time and the velocity field are depicted together with the corresponding anisotropically adapted meshes in figure 5.19. We can infer from the temperature profile that as the heated air enters the enclosure and gets in contact with the colder stationary fluid present there, an important gradient occurs. In parallel, two main vortices are formed and advected by the flow. Turbulent chaotic fluctuations are induced breaking up the steady nature of the flow inside the domain. As a response, the dynamic anisotropic mesh tool detects these gradients and removes elements from other locations and insert them in the vicinity of the sharp variations. Furthermore, as the flow advances small eddies are generated and developed. We can observe how the mesh follows their movement and ensures their good capture. The elements are very well oriented and highly stretched to better capture the solid

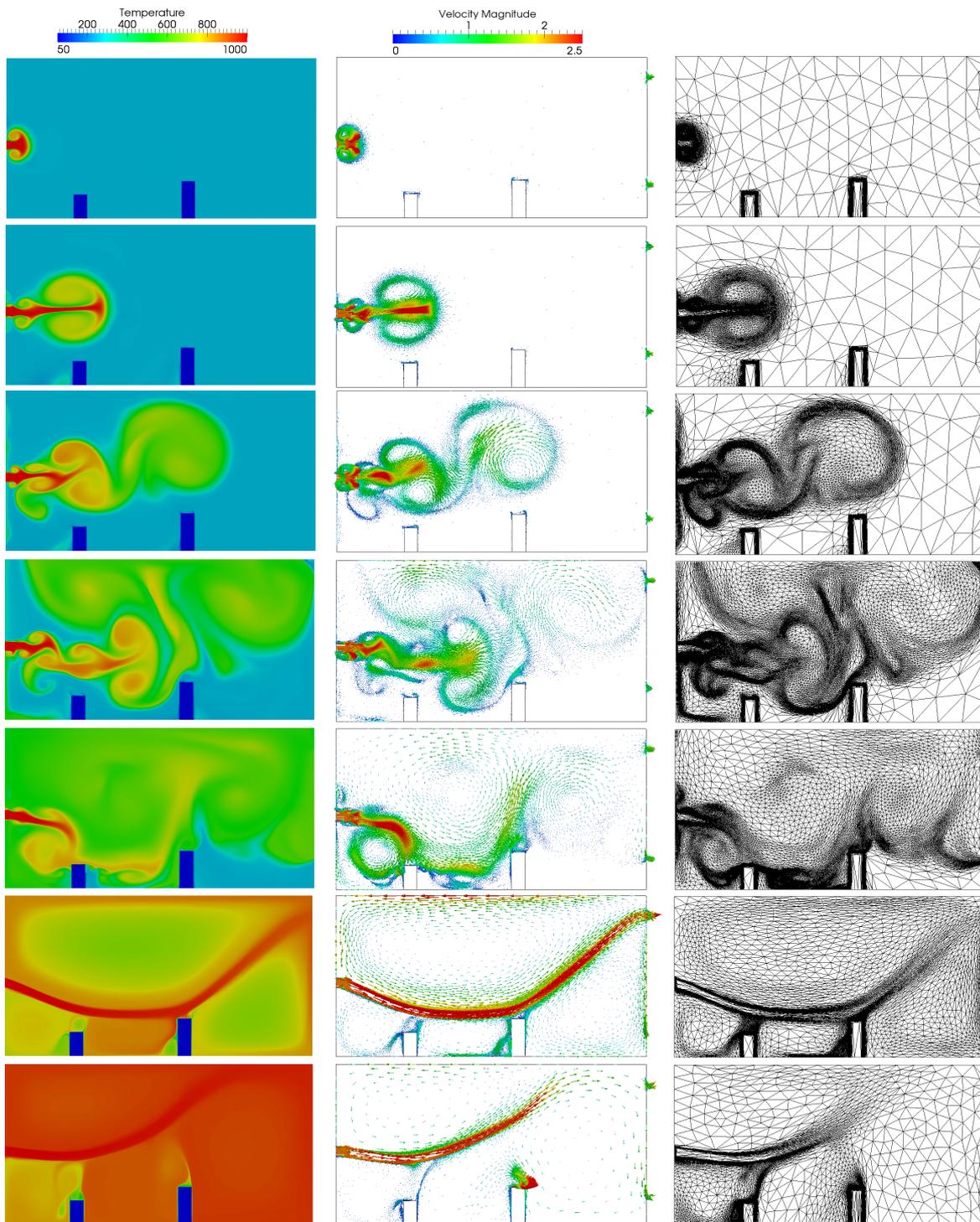


Figure 5.19 – Forced convection: temperature distribution, velocity profiles and the corresponding meshes over time.

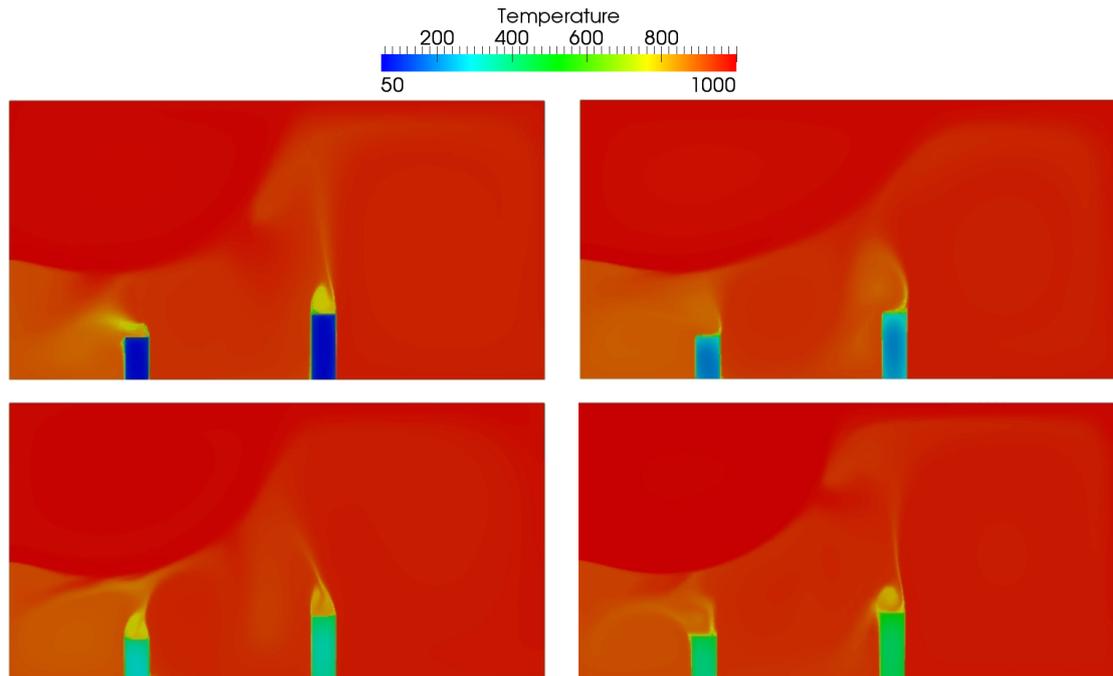


Figure 5.20 – Forced convection: thermal heating of immersed ingots.

interfaces as well as the evolution of the velocity and temperature fields. The boundary layers are also very well rendered by highly elongated elements in the flow direction.

On the other hand, we can clearly notice that the presence of ingots slows down the injected air from the burner and slightly influences the main air circulation inside the domain.

Once the chaotic flow settles down and a warm temperature is spread inside the cavity, the mesh becomes finer at the level of the solid bodies ensuring their accurate heating (figures 5.19 and 5.20). This example illustrates well the dynamic nature of the anisotropic adaptation. We can detect how the elements are refined, stretched and displaced over time and in accordance with the flow characteristics.

Regarding the time-stepping sizes over the simulation time, we present in figure 5.21 the history of the time-steps generated by the time-adaptive algorithm to advance the solution while maintaining the temporal error below the spatial one which is in turn controlled by a fixed number of nodes in the mesh  $N = 20,000$ . We observe that over the first 20s of the simulation, the time-step sizes are small allowing a precise capture of the newly emerging physics. As the flow settles down, and the temperature field in the cavity does not show much variations, the generated time-steps become larger accelerating the computations. Obviously, this is much more advantageous than evolving the solution with a fixed time-step size.

Finally, this test case can serve as a furnace design optimization one, whereby based on the profile of the flow's characteristic functionals, one can determine the best configuration for positioning the workpieces.

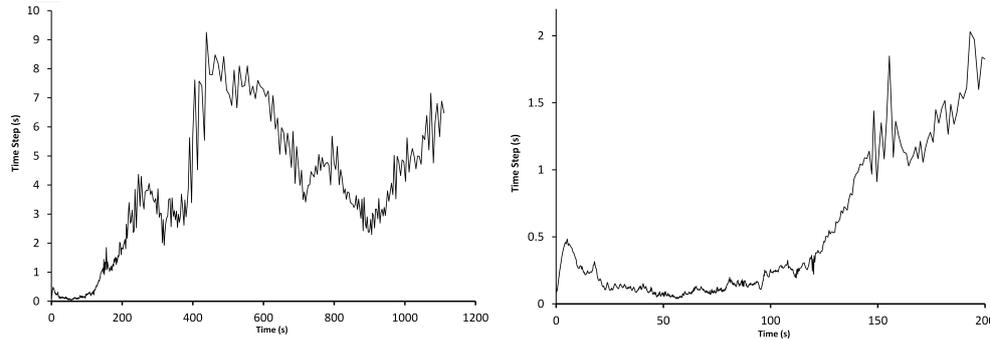


Figure 5.21 – Forced convection: time-steps evolution over time (left) and a zoom on the first 200s (right).

#### 5.4.4 Simulation of a rotating helicopter propeller

Now that the performance of the IVM with the anisotropic mesh adaptation and time adaptivity have been validated on heat transfer and flow problems, we move on to illustrate its performance in capturing moving structures. The aim is to investigate the capability of the developed tools to simulate airflows behind moving bodies. We now immerse a helicopter geometry in a large computational domain and let the propeller rotate with a constant angular velocity  $\theta = \frac{2\pi}{3}$  rad/s which is 3 full turns per second. We fix the number of nodes in the mesh to 200,000 and the meshing frequency to 5. The main challenge is to preserve the precise anisotropic and sharp capture of the blades over the simulation. If a constant time-stepping were to be employed together with the anisotropic mesh adaptation, then a very small size should be used in order to maintain the rotating interface of the propeller inside the refined zone. Indeed, when anisotropic mesh adaptation is employed, in the view of optimizing the mesh with the available number of nodes, the method tends to create a very sharp layer of condensed elements in the vicinity of the interface allowing its good capture. This layer has a very small thickness as shown by the 2D horizontal cut along the blade in figure 5.22. The mesh is highly refined around the Naca profile representing the zero-isovalue of the blade's levelset allowing its good capture. However, when a frequency of mesh adaptation is used and the propeller is rotating, the time-step size shall be set to a very small value in order to maintain the moving geometry in the highly refined zone. Otherwise, the refined mesh remains behind the rotating blades implying a very poor representation of the geometry as depicted in figure 5.23. The selected time-step size is equal to 0.001. One can clearly notice how the interface, the zero-isovalue of the levelset function, is very poorly represented. Therefore, it is hard to determine the fixed time-step size to be imposed in a way that guarantees the good capture of the moving interface. On the other hand, using the adaptive time-stepping approach developed in this work, the selection of the time-step size is automatic and in accordance with the mesh adaptation frequency. Figure 5.24 presents the propeller's displacement over time. We highlight the good capture of the geometry at the different instances. This is rendered by the appropriate selection of the time-step sizes. However, as reflected in the history of the first 0.05s of time-stepping (fig. 5.25), the generated sizes oscillate around a small

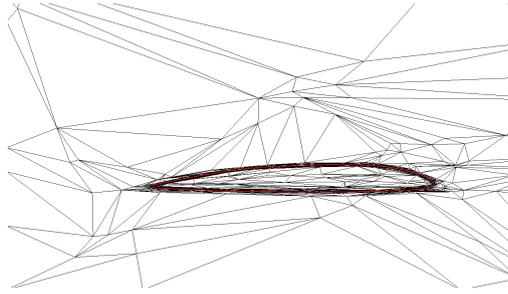


Figure 5.22 – 2D mesh of the horizontal cut along the blade and the zero-isovalue of the blade's levelset (in solid red line).

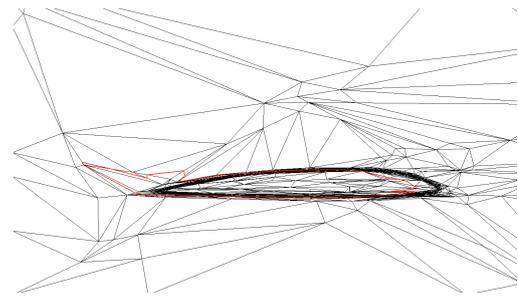


Figure 5.23 – Three time-steps after remeshing: mesh and zero-isovalue of the blade's levelset.

value  $\Delta t = 0.0005$ . This oscillatory profile is in good agreement with the nature of the movement as the blades perform a steady rotation over time.

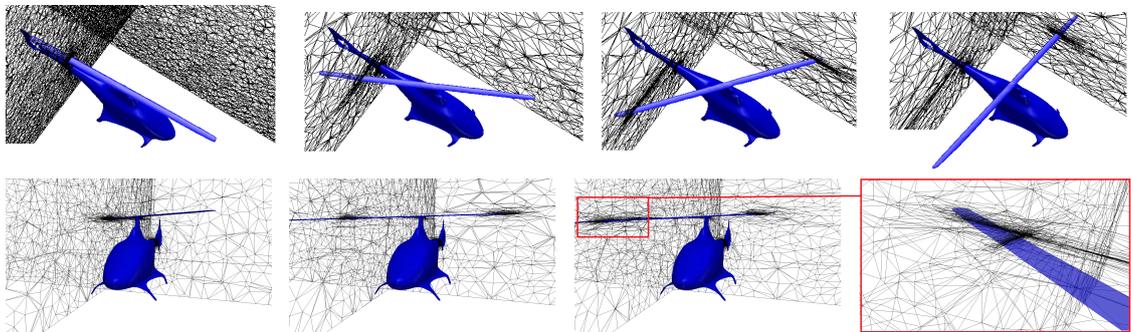


Figure 5.24 – Different snapshots on the mesh capture of the blade's trajectory over time and a zoom on the blade's mesh.

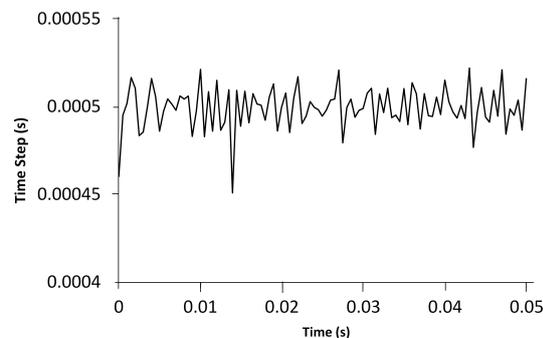


Figure 5.25 – History of the time-step sizes for the first 0.05s of the propeller's rotation.

For illustrative purposes, we re-consider a mesh of 500,000 nodes that is well refined around the immersed geometry and in a rectangular box that would contain the moving propeller over time and set the time step size to  $\Delta t = 0.0005$ . The simulations were run on 64 cores from a GNU/LINUX cluster of 2.4 Ghz Opteron cores connected by an InfiniBand network. Table 5.5 reports the CPU times needed using a fixed fine mesh and a small time-step size, an anisotropically adapted mesh and a fixed time-step and an anisotropically adapted mesh and an adaptive time-stepping for the simulation of 1s rotating immersed propeller inside a cavity. We can clearly identify an improve-

|                                      | CPU Time (h) |
|--------------------------------------|--------------|
| Fixed mesh & $\Delta t = 0.0005$     | 22.65        |
| Adaptive space & $\Delta t = 0.0005$ | 14.21        |
| Adaptive space & time                | 12.53        |

Table 5.5 – CPU time needed for the simulation of 1s propeller’s rotation.

ment in the execution time as a result of the space-time adaptation. However, the obtained CPU time is still significantly large for this kind of problems.

Consequently, in this numerical test case, the anisotropic mesh adaptation has provided a good level of accuracy but showed to be blocking the efficiency of computations. That is, as the mesh is being dynamically and anisotropically adapted, it necessitates a tiny time-step size in order to advance the moving interface in time. The reason for this limitation is related to the imposed frequency of adaptation. Two possible remedies can be adopted: the first consists in adapting the mesh at every time increment and the second would be to construct a mesh that is well adapted for several time-steps. The former solution is computationally expensive and accumulates interpolation error. The latter seems to be more accurate and efficient method and will be the subject of the following chapter.

## 5.5 CONCLUSION

In this chapter, we have introduced the Immersed Volume Method (IVM) and coupled it with the anisotropic mesh adaptation and time adaptive algorithms. This method is employed in the context of multi-material applications where the interfaces between the solid and the fluid subdomains are rendered by the zero-isovalue of a levelset function. A single mesh is constructed for both subdomains whereby a single set of equations is being solved. Mixing laws have been used to define the material properties in the different subdomains. A harmonic mean formulation is applied to handle abrupt changes in the material properties. In order to impose the no-slip boundary condition at the solid body’s level, a high value for kinematic viscosity is considered in that region making the velocity component in the momentum equation negligible. The dynamic anisotropic mesh adaptation permitted a sharp capture of the interfaces and an accurate material distribution over the different subdomains. The method has been tested on examples of flow behind solid bodies and a heat transfer problem and proved to be a powerful tool for simulating fluid/structure interaction problems. A  $P_1$  radiative formulation, developed and implemented in the CIMLIB library, was used to account for the radiative heat transfers. In chapter 7, we intend to validate the developed tools brought forth in this thesis work on industrial applications and confront the results to experimental data.

We have also tested the IVM and the developed dynamic adaptations on a moving solid problem. Although the results were accurate, the method showed to be computationally expensive in that context. Indeed, when a frequency of remeshing is being

used, the constructed mesh is not well prepared to contain the evolving solution over the interval of no adaptation. Consequently, the time-adaptive algorithm, in the view of limiting the computational error tends to generate very small time-steps thus limiting the advancement of the solution over time. We intend, in the following chapter, to derive an adaptation technique that takes into account a frequency of remeshing. The method generates meshes and time-step sizes that are adequate to several time-steps and hence the solution would be able to evolve more smoothly in time.

## 5.6 RÉSUMÉ FRANÇAIS

Dans le cadre de simulation de problèmes d'interaction fluide-structure, deux approches peuvent être utilisées: les approches de partitionnement des domaines et les approches monolithiques. Les premières considèrent les domaines de fluides et solides séparément puis le couplage s'effectue par des conditions aux limites. Néanmoins, la détermination de ces conditions est compliquée et des instabilités peuvent apparaître lorsque les ratios des propriétés physiques sont importants. En revanche, les approches monolithiques considèrent un seul domaine de calcul avec des propriétés physiques variées pour les différents fluides et solides. Un seul maillage est alors utilisé et ne nécessite pas sa coïncidence avec la frontière fluide-solide. Dans cette thèse nous avons adopté la méthode d'immersion de volume, une approche monolithique, par laquelle les objets solides sont immergés et localisés dans la partie fluide en utilisant une fonction distance signée, la levelset. Ensuite, les propriétés physiques et thermiques des différentes composantes sont distribuées à l'aide d'une fonction Heaviside lissée. L'avantage de cette méthode réside dans sa capacité de s'affranchir de la nécessité de déterminer les coefficients d'échanges thermiques au niveau de l'interface. Cependant, une bonne précision du maillage à ce niveau permet une bonne distribution des propriétés.

Dans ce chapitre, nous nous sommes intéressé au couplage de la méthode d'immersion de volume et la méthode d'adaptation anisotrope du maillage. L'idée était non seulement de générer un maillage raffiné au niveau de l'interface mais aussi d'avoir un maillage qui s'adapte dynamiquement et automatiquement aux caractéristiques de l'écoulement autour. Afin de répondre à ces besoins, une extension de l'adaptation anisotrope à une approche multi-composante a été introduite. L'originalité de cette méthode réside dans sa simplicité et sa capacité d'adapter le maillage sur plusieurs champs en une seule métrique.

Le couplage a été testé sur plusieurs applications d'écoulement de fluide et de transfert de chaleur et s'est avéré être un outil puissant pour simuler des problèmes d'interaction fluide/structure. En effet, les maillages obtenus par la méthode d'adaptation anisotrope permettent de capturer précisément les interfaces des objets immergés, les différents vortexes, les couches limites et les gradients de température. L'adaptation en espace et en temps a démontré encore une fois sa capacité d'améliorer

les temps de calcul ce qui rentre dans l'objectif général de la thèse d'arriver à des simulations réalistes de traitement thermiques dans des fours industriels.

Nous avons également testé la méthode d'immersion de volume et les méthodes d'adaptations dynamiques développées sur un problème de mouvement d'interface. Dans ce contexte, bien que les résultats furent précis, la méthode s'est avérée coûteuse en temps de calcul. En effet, quand une fréquence de remaillage est utilisée, le maillage construit n'est pas bien préparé pour contenir l'évolution de la solution le long de l'intervalle de temps au cours duquel le maillage est maintenu. Par conséquent, l'algorithme d'adaptation de temps, dans la vue de limiter l'erreur commise, a tendance à générer de très petits pas de temps limitant ainsi l'avancement de la solution au fil du temps. Nous avons l'intention, dans le chapitre suivant, de développer une technique d'adaptation qui prend en compte une fréquence de remaillage. Le procédé génère des maillages et des tailles de pas de temps qui vont être utilisés durant plusieurs pas de temps et par conséquent la solution serait capable d'évoluer plus facilement dans une région bien adaptée au cours du temps.

# SPACE-TIME SLAB ADAPTIVE MESHING

# 6

"How wonderful that we have met with a paradox.  
Now we have some hope of making progress."  
Niels Bohe

## CONTENTS

---

|       |                                                                          |     |
|-------|--------------------------------------------------------------------------|-----|
| 6.1   | STATE OF THE ART                                                         | 267 |
| 6.2   | SINGLE TIME-STEP SPACE-TIME ADAPTIVE REMESHING                           | 269 |
| 6.3   | PARADOXICAL MESHING: FULL ADAPTIVITY ALGORITHM                           | 270 |
| 6.3.1 | Validity of the generated mesh and frequency of remeshing                | 271 |
| 6.3.2 | A predictor-corrector approach                                           | 272 |
| 6.3.3 | Generating time-slabs for adaptation                                     | 273 |
| 6.3.4 | Solution sampling                                                        | 274 |
| 6.3.5 | Edge-based error estimation                                              | 275 |
| 6.3.6 | Metric construction for a slab of time                                   | 275 |
| 6.3.7 | Temporal stretching factors                                              | 276 |
| 6.3.8 | Space and time remeshing                                                 | 277 |
| 6.3.9 | Convergence of the space-time adaptive algorithm                         | 278 |
| 6.4   | PARADOXICAL MESHING ALGORITHM                                            | 279 |
| 6.5   | 1D TEMPORAL MESHER                                                       | 280 |
| 6.5.1 | Application of the Time mesher                                           | 290 |
| 6.6   | NUMERICAL EXAMPLES                                                       | 291 |
| 6.6.1 | A two-dimensional rotating Circle                                        | 291 |
| 6.6.2 | A two-dimensional analytical test case                                   | 294 |
| 6.6.3 | A two-dimensional analytical test case with sinusoidal evolution in time | 295 |
| 6.6.4 | Fluid flow for different Reynolds numbers                                | 297 |
| 6.6.5 | Flow past a wind turbine                                                 | 298 |
| 6.6.6 | Conjugate heat transfer inside an industrial furnace                     | 308 |
| 6.7   | CONCLUSION                                                               | 310 |
| 6.8   | RÉSUMÉ FRANÇAIS                                                          | 311 |

---

**I**N the previous chapters, we have developed space and time adaptive algorithms for the resolution of steady and unsteady CFD applications. The methods proved to yield a good level of accuracy and reduced the computational cost of the simulations. Nevertheless, when applied in the context of problems involving rapid solution's evolution, these adaptations induced a deceleration in the computations. One way to preserve the good level of accuracy, is to apply the mesh adaptation at every solver iteration. Another possibility is to generate very small time-steps to advance the solution within the refined zones. Adapting the mesh at every solver iteration would not only yield an important execution cost but also accumulate interpolation errors. These problems can be treated by lowering the frequency of mesh adaptation. However, in the case of an imposed fixed time-step size, as the mesh is being predicted only for the very next solver iteration, it remains behind the advancing interface yielding a loss of accuracy. When applying the time-adaptive algorithm together with the anisotropic mesh adaptation, the moving interface is precisely captured but the generated time-step sizes are too small and thus the computations are very slow. To overcome the above mentioned bottlenecks, a new fully adaptive method is proposed in this chapter. It intends to automatically generate a mesh and a set of time-step sizes. This mesh/time step sizes couple takes into account the solution's evolution over the concerned period of time.

## 6.1 STATE OF THE ART

The numerical simulation of time-dependent problems constitutes an important tool for the understanding of physical phenomena. Nevertheless the latter are usually characterized by a multiscale nature; then in order to properly handle and accurately capture complex phenomena a considerable computational effort is needed. A very fine mesh, employing an unnecessarily large number of vertices and hence a large computational cost, is required everywhere on the domain so that an accurate numerical solution is computed. Therefore mesh adaptation is crucial when dealing with time dependent problems especially if the solution is not known a priori and presents an arbitrary behavior all over the computations. On the other hand, the time-step size should be carefully chosen so that to reproduce the time multiscale features of the problem.

When simulating 3D complex physical phenomena and industrial applications, it is highly desirable to provide a good level of accuracy within a reasonable computational cost. Therefore, in order to avoid excessive computational expenses due to mesh adaptation, a frequency of adaptation is generally adopted. However, this procedure may yield a delay of the mesh with respect to the solution. Assuming that we adapt the mesh based on the solution computed at time  $t^n$  using the algorithm introduced in chapter 2, when the solution advances in time, the mesh will no longer provide an error control at time  $t^{n+1}$ . The time adaptive algorithm devised in chapter 4 tends to limit the induced error by generating small time-step sizes. In other words, as the solution at time  $t^{n+1}$  is computed on the new mesh which is optimal based on the solution at time  $t^n$ , this mesh will not necessarily be the optimal one at time  $t^{n+1}$ . This is the case when propagating a discontinuity, or moving a geometry with time.

Dynamic body movement inside a computational domain can be investigated using the body fitted or the immersed/embedded techniques. The authors in [Hassan 07] provided a summary on the different techniques developed in the literature to enable the simulation of flows involving moving boundaries. In the body-fitted framework, we can distinguish several approaches to handle simulations with moving structures [Farhat 98, Johnson 99, Saksono 07, Hassan 07, Compère 10, Alauzet 13]. The general idea proposed in [Farhat 98] is to follow the boundary by moving the nodes surrounding the objects in the mesh while maintaining their connectivity. We omit the details of these methods for brevity and refer the reader to [Hassan 07] for more details. In this work, to handle geometry displacement, we are interested in adopting the immersed volume method (IVM), discussed in chapter 5, for its flexibility, ease of use and monolithic formulation. In order to account for the moving objects, it suffices to let the corresponding levelset function vary in time. We couple the IVM technique with the developed dynamic anisotropic mesh adaptation and time adaptive algorithm. We recall that during the error analysis performed prior to remeshing, a uniform error  $\varepsilon(N)$  is computed and intends to equi-distribute the error over the edges in the mesh under the constraint of a fixed number of nodes  $N$ . This error is then used to control the temporal error. The latter is limited by  $\frac{\varepsilon(N)}{f}$ , where  $f$  is the frequency of mesh adapta-

tion. Hence, the error will also be equi-distributed over the temporal edges. However, in the case of simulations involving discontinuity propagation or moving geometries, as shown in example 5.4.4, the generated time-steps are very small thus blocking the advancement of the solution over time.

One possible remedy to this problematic involves adapting more frequently the mesh so that the solution's gradient remains in the refined zones. Nevertheless, another problematic needs to be addressed in this case. It is related to the interpolation error induced at each remeshing as the solution should be transferred from the old mesh to the new optimal one. Consequently, the frequency of mesh adaptation affects the global accuracy of the numerical solution. In order to avoid the excessive interpolation error due to frequent remeshing, it was suggested in [Löhner 92, Rausch 92] to adapt the mesh based only on refinement/coarsening without vertex displacement. A safety region around critical zones are introduced to maintain the solution's gradient in the refined locations between two adaptations. However, as a result of creating safety regions, the generated meshes are no longer optimal in the anisotropic sense. Moreover, these methods provide no control on the temporal error induced over the computations.

Another approach for dealing with the mesh/solution compatibility would be to employ a dynamic frequency of adaptation [Zienkiewicz 87, Zienkiewicz 92, de Sampaio 93, Micheletti 08a, Picasso 09]. That is, one defines a certain threshold value on the global interpolation error and when the latter exceeds the specified value, the mesh adaptation algorithm is applied. These methods are derived from on strong mathematical analysis of the equations at hand and provide space-time error estimates for unsteady problems. In practice, the mesh is adapted only when a threshold value is exceeded, i.e. a degradation of mesh/solution compatibility is permitted up to a certain value. However the result of this degradation is not tracked and can affect the global accuracy of the simulation. An extension of these methods consists in conserving the same mesh connectivity while moving the nodes using an R-adaptation technique. The equations are then solved in a fully Arbitrary Lagrangian Eulerian (ALE) way. The process is repeated until the mesh elements' quality deteriorates. At that level, an H-adaptation is performed creating a new mesh topology on which the ALE resolution is applied. Although this approach reduces the interpolation error as a result of the ALE computations, in practice when simulating real physical applications, the number of remeshing can drastically increase affecting negatively the CPU time. For instance, the authors in [Saksono 07] adopted this approach to a turbo-machinery problem whereby in order to preserve a good precision, the mesh was adapted at every solver iteration. This is in addition to the generation of badly shaped elements as a result of the R-adaptation, thus affecting the convergence of the solvers and the quality of the solution. In the immersed volume framework, the mesh adaptation is more straightforward, as the geometry's interface is defined as the zero-iso-value of a levelset function. As the geometry advances in time, the levelset will be recomputed and the mesh will be re-generated accordingly.

Using the time-adaptive technique proposed in chapter 4, the error is bounded

all over the simulation thus bypassing the need to move the mesh nodes during the no-adaptation interval. However, the time-step sizes are considerably reduced thus increasing the total CPU time of the simulation.

In this chapter, we are interested in developing a new approach, called the '*paradoxical meshing*', that guarantees the compatibility of the mesh/solution couple without accumulating excessive interpolation errors due to the transfer of the solution from an old mesh to a new one. The objective is to build up a mesh that controls the space and time interpolation error over a time interval  $[t^n, t^{n+f}]$ . For this purpose, we revisit the theory of classical adaptation developed in chapters 2 and 4 on anisotropic meshing and extend it to contain an analysis for several solver iterations. The method will be based on a predictive-corrective formulation. It is referred to as paradoxical as it permits to overcome the paradox of predicting a mesh, based on a posteriori error analysis, that will contain the upcoming unknown solution. The developed space-time adaptive algorithm is independent from the properties of the problem at hand and significantly reduces the computational cost and improves the global accuracy of the calculations. In the context of the immersed volume method, with applications involving moving geometries, it should be capable of following the time-evolving interfaces.

This chapter is divided into 6 sections. We start in section 1 by defining the general structure of the new space-time adaptive algorithm. Then we move on in section 2 to extend the edge-based error estimator to account for intervals of time. The optimal metric controlling the global  $L_p$  norm of the interpolation error under the constraint of a fixed number of nodes over the interval of time will then be presented in section 3. Section 4 discusses the temporal error control in accordance with the new spatial equidistributed error. A 1D time mesher will be developed in section 5 allowing the redistribution of the temporal nodes according to the predicted optimal time-slab sizes. Finally, section 6 provides some numerical examples validating the performance and the potential of the new method.

## 6.2 SINGLE TIME-STEP SPACE-TIME ADAPTIVE REMESHING

When dealing with steady state problems, the single time-step mesh adaptation technique, also called the 'classical adaptation' technique presented in chapter 2, performs very well for converging the mesh-solution couple. Nevertheless, this method is no longer optimal, from an accuracy/efficiency standpoint, when applied with a certain frequency to unsteady problems as the physical solution evolves in time. The validity of a mesh cannot be determined a priori. So if the mesh is not adapted at every solver iteration, the time-step size might be greater than the length of the mesh's validity interval; hence the solution may propagate into a non pre-adapted region of the domain resulting in a mesh/solution lag and an increase in the numerical error.

Together with the time-adaptive method introduced in chapter 4, the single time-step mesh adaptation can be efficiently and accurately adapted to time dependent problems. The general structure of the simulation flow using a single time-step space-time

adaptive method is summarized in algorithm 4. In this algorithm, two parameters are to be supplied by the user: the number of nodes desired in the mesh, which is usually set in accordance with the available computational resources, and the frequency of adaptation, that can be regarded as a calibration between efficiency and accuracy. For more details on the anisotropic mesh adaptation and time-adaptive procedure, the reader can consult algorithm 3. We note that this algorithm can also be applied in the case of moving geometries where the solution is considered as a levelset function that is advancing in time.

---

**Algorithm 4:** Single time-step space-time adaptive algorithm based on a frequency of adaptation for unsteady problems.

---

**Input:** An initial mesh-solution couple  $(\mathcal{H}_0, \mathcal{S}_0^0)$ .

An initial time-step size  $\Delta t_0^0$ .

A user-prescribed mesh complexity  $N$ .

A user-prescribed frequency of adaptation  $f$ .

**Output:** At each time  $t$ , the mesh-solution couple  $(\mathcal{H}_k, \mathcal{S}_k^j)$  and the time-step size  $\Delta t_k^j$  for the following time increment.

**begin**

Set the mesh counter  $k$  and the time  $t$  to 0.

**while**  $t < T$  *i.e. the final time of the simulation is not reached*, **do**

Construct the metric tensor  $\mathcal{M}_k$ .

Generate the optimal mesh  $\widetilde{\mathcal{H}}_k$ .

Interpolate the solution  $\mathcal{S}_k^j$  from the mesh  $\mathcal{H}_k$  onto the mesh  $\widetilde{\mathcal{H}}_k$ .

**for**  $j : 1 \rightarrow f$  **do**

    Compute the solution  $\mathcal{S}_k^j$  on the mesh  $\widetilde{\mathcal{H}}_k$ .

    Determine the time-step size  $\Delta t_k^j$  for the following time-increment.

    Advance in time:  $t \rightarrow t + \Delta t_k^j$ .

Update the counter  $k$ :  $k \rightarrow k + 1$ .

---

### 6.3 PARADOXICAL MESHING: FULL ADAPTIVITY ALGORITHM

In this chapter, we develop a new method for full adaptation in the case of unsteady problems or applications involving moving geometries. As a first step, the computation time interval is divided into several subintervals  $[t^n, t^{n+n_{\text{freq}}}]$ . The technique relies on a predictor corrector approach whereby a first run on an interval  $[t^n, t^{n+n_{\text{freq}}}]$  is conducted with relatively large time step sizes on the same initial mesh. Inhere, the time interval  $[t^n, t^{n+n_{\text{freq}}}]$  is constituted of  $n_{\text{freq}} + 1$  temporal nodes with  $n_{\text{freq}}$  being the frequency of adaptation. On each of the spatial edges of the initial mesh, the maximum estimated error of interpolation over the time interval is stored. Based on these estimations, an equi-distributed error is then determined for the whole interval of time. Stretching factors are also defined in the same way as in the single time-step formulation. Then a metric construction follows yielding an optimal mesh adapted, for a specific number

of nodes, to the solution's evolution over a certain time interval. On the other hand, as in the single time-step approach, temporal errors are determined along the temporal edges. Afterwards, using the computed equi-distributed error, and the estimated temporal errors, stretching factors are evaluated on the temporal edges. The process is repeated for the different subintervals  $[t^n, t^{n+n_{\text{freq}}}]$  constituting the simulation time interval. Once all the estimations have been gathered, spatial mesh adaptation based on the estimated metrics are generated using the MTC mesher and a time-adaptation is performed based on a newly developed 1D temporal mesher. All the details of this approach will be provided in the following. In this chapter, we develop a new method for full adaptation in the case of unsteady problems or applications involving moving geometries. As a first step, the computation time interval is divided into several subintervals  $[t^n, t^{n+n_{\text{freq}}}]$ . The technique relies on a predictor corrector approach whereby a first run on an interval  $[t^n, t^{n+n_{\text{freq}}}]$  is conducted with relatively large time step sizes on the same initial mesh. Inhere, the time interval  $[t^n, t^{n+n_{\text{freq}}}]$  is constituted of  $n_{\text{freq}} + 1$  temporal nodes with  $n_{\text{freq}}$  being the frequency of adaptation. On each of the spatial edges of the initial mesh, the maximum estimated error of interpolation over the time interval is stored. Based on these estimations, an equi-distributed error is then determined for the whole interval on time. Stretching factors are also defined in the same way as in the single time-step formulation. Then a metric construction follows yielding an optimal mesh adapted for a specific number of nodes to the solution's evolution over a certain interval of time. On the other hand, as in the single time-step adaptation approach, temporal errors are determined along the temporal edges. Afterwards, using the computed equi-distributed error, and the estimated temporal errors, stretching factors are computed on the temporal edges. The process is repeated for the different subintervals  $[t^n, t^{n+n_{\text{freq}}}]$  constituting the simulation time interval. Once all estimations have been gathered, spatial mesh adaptation based on the estimated metrics are generated using the MTC mesher and a temporal time-adaptation is performed based on a newly developed 1D temporal mesher. All the details of this approach will be provided in the following.

Before proceeding into the details of the newly developed paradoxical meshing method, we illustrate on the validity interval of the meshes generated using the single time-step adaptation technique.

### 6.3.1 Validity of the generated mesh and frequency of remeshing

The validity of a mesh is defined as the size of the largest time interval during which the error remains bounded and the solution is accurately captured. It is worth mentioning that the frequency of remeshing is in direct correlation with the duration of applicability of the mesh. Note that this parameter depends on the desired error, the time-step sizes, the speed of propagation and the evolution of the desired solution/geometry. As the time-dependent problems exhibit arbitrary progression with time, the duration of applicability of a mesh cannot be known a priori. When the time-step size is greater

than the length of the mesh's time interval, the solution may propagate into a non pre-adapted region of the domain resulting in a mesh/solution lag. Hence, a remarkable increase in the numerical error is induced. A possible solution to this problem is to adapt the mesh at each solver iteration guaranteeing that the spatial error remains bounded. However this approach increases drastically the computational cost and thus induces an important CPU time. On the other hand, when adapting the mesh at every time-step a transfer of the variables through interpolation is required, and this causes the accumulation of interpolation errors that affects the solution's accuracy on the long run.

The above described bottlenecks of the single time-step mesh adaptation method when applied to time-dependent problems raise the need for an extension of the developed tools to anticipate the solution's evolution over a period of time and generate the corresponding mesh.

### 6.3.2 A predictor-corrector approach

A predictor corrector principle, called the fixed point method, was investigated in the literature for developing space and time adaptive algorithms [Zohdi 07, Alauzet 07]. The approach proposed in [Zohdi 07] focuses on the prediction of the solution's behavior over a time interval and based on the induced error, an optimal time-step size is deduce. Another variant of this method for the resolution of unsteady problems was proposed in [Alauzet 07]. The main idea consists in dividing the simulation time  $[0, T]$  into subintervals on each of which a single adapted mesh is used and intends to minimize the global space-time error. The generation of the adapted meshes is preceded by a prediction phase whereby the solution's evolution over the time interval is predicted. However in this work, the time-step sizes are either user specified, or controlled by the CFL condition. In the former case, when the problem's characteristics are not known a priori, the time-step size is maintained at a small value to guarantee the convergence of the computations. In the latter case, the size is proportional to the smallest altitude in the mesh. Consequently, a single small-altitude element, which is very likely in the case of anisotropic meshing, can considerably reduce the time-step size and thus drastically increase the simulation's CPU time. Therefore, the fixed point algorithm, provides a good level of accuracy by constructing meshes well adapted to the displacements of the solutions over time. It reduces the computational cost when compared to methods involving frequent remeshing. However, the time-step size is limited either by a CFL condition or a user-prescribed value which in turn may increase the CPU time of the simulation. Moreover, a control only on the  $L_\infty$  norm of the space-time interpolation error is provided. A metric is constructed at every time-increment in the time-subinterval. The set of constructed metrics are then intersected yielding a single metric and thus a mesh controlling the maximum space and time errors. The process is repeated until the convergence of the mesh/solution couple. In addition to the above mentioned bottlenecks of the algorithm, the convergence criterion is defined in terms of a fixed

number of iterations  $n_{\text{nptfx}}$ . Nevertheless, the choice of this number is pre-defined and independent of the problem's features which is not optimal as a convergence indicator. Furthermore, the fixed point algorithm, by controlling the space-time error in a single metric that will be used to generate the anisotropic mesh for the intended interval of time, might not capture a sharp spatial discontinuity that does not evolve in time. This is the case of a stationary shock or a non-moving object's interface.

In this work, we are interested in developing a space and time fully adaptive algorithm. The latter aims at anticipating the solution's progress over a period of time and generating the optimal mesh that is adequately adapted, for a fixed number of nodes, to the evolving solution along that time interval. The analysis is carried out on a (3D+1D) mesh, in other words the computations are performed synchronously on a three dimensional spatial mesh and a one dimensional temporal mesh. So the time is considered as an additional dimension taken to be orthogonal to the spatial one. We aim at generating a mesh that remains valid for several time-increments together with the corresponding optimal set of time-step sizes. Note that, in this case, the user can assign a frequency of adaptation and the algorithm will accordingly adapt the meshes. The scheme predicts the transient behavior and the displacement of the solution over an interval of time and produces the optimal mesh and set of time-step sizes that give the most accurate capture of the physical phenomena.

The method combines the advantages of the fixed point algorithm, with the anisotropic mesh adaptation and time-adaptive techniques developed in chapters 2 and 4. The objective is to address and answer several problematics including: the control of the space and time interpolation errors, the generation of anisotropic meshes that are optimal for slabs of time, and the optimal definition of time-step sizes. Unlike the fixed point algorithm that employs a metric intersection to generate a single metric for the time-slab, we intend to use the developed multi-components construction that accounts for several fields in a single metric. Moreover, in our approach, the space and time dimensions are decoupled in the sense that the control provided by the generated anisotropic mesh depends entirely on the spatial error estimates whereas the stretching of the time-step sizes will be generated in such a way to bound the temporal error by the computed equi-distributed error. In that sense, a discontinuity that is not evolving in time will be compared to other spatial discontinuities/locations of solution's high gradient and accordingly the mesh will be anisotropically adapted to the regions of sharpest evolutions.

### 6.3.3 Generating time-slabs for adaptation

The first step in the paradoxical meshing algorithm, as in the fixed point algorithm, consists in dividing the simulation time interval  $[0, T]$  into  $n_{\text{SI}}$  subintervals, time-slabs,

such that:

$$[0, T] = [0, T^1] \cup [T^1, T^2] \cup \dots \cup [T^{k-1}, T^k] \cup [T^k, T^{k+1}] \cup \dots \cup [T^{n_{SI}-2}, T^{n_{SI}-1}] \cup [T^{n_{SI}-1}, T^{n_{SI}}]$$

each will in turn be divided into  $n_{\text{freq}}$  subintervals where  $n_{\text{freq}}$  is the frequency of adaptation assigned by the user. A single adapted mesh will be generated in the sequel for each of these subintervals. We call the adaptation method a paradoxical meshing as the resulting mesh is being adapted to  $n_{\text{freq}}$  time-steps while adapting every  $n_{\text{freq}}$  steps. It is important to mention that, unlike the fixed point algorithm, neither the length of the time-slabs nor their number is fixed over the computations. On the contrary, these two parameters will be dynamically updated based on the induced temporal error.

The mesh and the set of time-step sizes are computed through an iterative process along which we try to converge both meshes (the spatial and the temporal one) to the optimal configurations that give the most accurate solution for the corresponding interval of time. At every iteration, we consider each of the  $n_{SI}$  intervals at a time and divide it into  $n_{\text{freq}}$  subintervals as shown in figure 6.1:

$$[T^k, T^{k+1}] = [T^k = t_0^k, t_1^k] \cup [t_1^k, t_2^k] \cup \dots \cup [t_l^k, t_{l+1}^k] \cup \dots \cup [t_{n_{\text{freq}}-1}^k, t_{n_{\text{freq}}}^k] = T^{k+1}]$$

such that the temporal node  $t_{l+1}^k$  is defined by:

$$t_{l+1}^k = t_l^k + \Delta t_{k,l} = T^k + l \times \Delta t_{k,l}, \quad (6.1)$$

with

$$\Delta t_{k,l} = \frac{\delta t_k}{n_{\text{freq}}}, \quad (6.2)$$

being the time-step size at the temporal node  $t_{k,l}$  and  $\delta_k$  is the time-slab size.

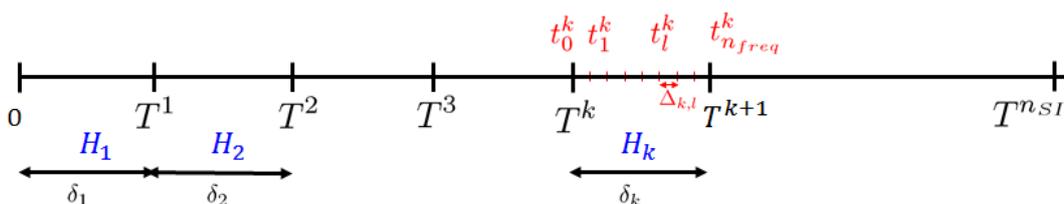


Figure 6.1 – Time slabs  $[T^{k-1}, T^k]$  and their corresponding initial meshes.

This discretization will result in a 1D temporal mesh of the time interval  $[0, T]$ . The elements in this mesh are the time-slabs  $[T^k, T^{k+1}]$  of length  $\delta t_k$ . The latter slabs of time are in turn split into  $n_{\text{freq}}$  sub-intervals of the same length  $\Delta t_{k,l}$ .

### 6.3.4 Solution sampling

Without loss of generality, we assume that we are at a certain iteration of the paradoxical meshing algorithm. Each time slab  $[T^k, T^{k+1}]$  is considered at a time. The physical problem's solution  $\mathcal{S}_{k,l}$  is predicted at each of the  $\{t_l^k\}$  temporal nodes in this sub-interval using the numerical scheme. We note that the solution can be a vector

of several components such as the velocity, pressure, temperature, levelset function, etc. In practice, the vector will include only the variables of interest for the anisotropic adaptation. An example of such a multi-components normalized vector can be:

$$\mathcal{S}_{k,l} = \underbrace{\left\{ \frac{\mathbf{v}^i}{\|\mathbf{v}^i\|}, \frac{\|\mathbf{v}^i\|}{\max_j \|\mathbf{v}^j\|}, \frac{\alpha}{\max(\alpha)}, \dots \right\}}_{n_{\text{comp}} \text{ components}} \quad (6.3)$$

where  $\mathbf{v}^i$  denotes the velocity field at node  $X^i$ , and  $\alpha$  stands for any scalar field (temperature, pressure, levelset function,  $\dots$ ).

The collected solutions are then gathered in a vector field associated with the time slab:

$$\mathcal{V}_k = \left\{ \mathcal{S}_{k,0}, \mathcal{S}_{k,1}, \mathcal{S}_{k,2}, \dots, \mathcal{S}_{k,l}, \dots, \mathcal{S}_{k,n_{\text{freq}}-1} \right\} \quad (6.4)$$

### 6.3.5 Edge-based error estimation

In the view of controlling the interpolation error for several time-increments constituting the time slab  $[T^k, T^{k+1}]$ , we propose an  $L_\infty(t; L_p(\Omega))$  approach for defining the error estimates on the edges of the mesh. In other words, we define the spatial error estimate at every temporal node  $\{t_l^k\}$  in an  $L_p$  interpolation sense using the constructed field of gathered data:

$$e_{ij}^s(t_k^l) = \left\| e_{ij}^{k,l} \right\|_p = \left\| \left\{ |\mathbf{G}_1^{ij} \cdot \mathbf{X}^{ij}|, |\mathbf{G}_2^{ij} \cdot \mathbf{X}^{ij}|, \dots, |\mathbf{G}_m^{ij} \cdot \mathbf{X}^{ij}|, \dots, |\mathbf{G}_{n_{\text{comp}}}^{ij} \cdot \mathbf{X}^{ij}| \right\} \right\|_p \quad (6.5)$$

with  $\mathbf{G}_m^i$  being the reconstructed gradient at node  $X^i$  of the  $m^{\text{th}}$  component of the solution's vector field  $\mathcal{S}_{k,l}$ .

Then for each edge in the mesh, the maximum of the collected error estimates among the different temporal nodes will be considered during the construction of the metric tensor, i.e. the error on edge  $\mathbf{X}^{ij}$  will be set to:

$$e_{ij}^s([T^k, T^{k+1}]) = \max_{t_k^l \in [T^k, T^{k+1}]} e_{ij}^s(t_k^l) \quad (6.6)$$

Therefore, during the remeshing process, the edges, on which the highest magnitude of the errors over time is spotted, will be given more importance than the ones with the lower magnitudes. We note the use of the  $L_p$  norm in the computation of the interpolation errors along the edges at every temporal node, which is calibrated to capture the multi-scale aspects of physical problems.

### 6.3.6 Metric construction for a slab of time

Now that we have estimated the errors on the edges of the mesh over the slab of time, we move on to construct the appropriate metric tensor for a fixed number of nodes  $N$ . To do so, we follow the lines in section 2.4 and determine the modified equidistributed

error as follows:

$$\varepsilon_{[T^k, T^{k+1}]}(N) = \left( \frac{N}{\sum_i n_{[T^k, T^{k+1}]}^i(1)} \right)^{-\frac{4}{d}}, \quad (6.7)$$

where,

$$n_{[T^k, T^{k+1}]}^i(1) = \sqrt{\det \left( \frac{1}{d} (\mathbb{X}^i)^{-1} \left( \sum_{j \in \Gamma(i)} \left( \frac{1}{e_{ij}^s([T^k, T^{k+1}])} \right)^{-\frac{1}{2}} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right) \right)} \quad (6.8)$$

denotes the number of created nodes along the different directions of the edges connected to  $X^i$ , and

$$\mathbb{X}^i = \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \quad (6.9)$$

is the length distribution tensor at a node  $X^i$ . We note that it is also possible to employ the privileged length distribution tensor at that level.

Consequently, the metric tensor  $\mathcal{M}_{[T^k, T^{k+1}]}^i$  controlling the  $L_\infty(t; L_p(\Omega))$  spatial interpolation error along the edges connected to node  $X^i$  is determined by:

$$\mathcal{M}_{[T^k, T^{k+1}]}^i = \frac{1}{d} \left( \frac{1}{|\Gamma(i)|} \sum_{j \in \Gamma(i)} s_{ij}^2 \mathbf{X}^{ij} \otimes \mathbf{X}^{ij} \right)^{-1} \quad (6.10)$$

with,

$$s_{ij} = \left( \frac{\varepsilon_{[T^k, T^{k+1}]}(N)}{e_{ij}^s([T^k, T^{k+1}])} \right)^{\frac{1}{2}} \quad (6.11)$$

being the scaling factors associated with stretching, or shrinking, the edges  $\mathbf{X}^{ij}$  in their own direction. The resulting metric will then be used to generate an anisotropic mesh well adapted at the locations of sharp solutions' gradients over the time slab  $[T^k, T^{k+1}]$ .

### 6.3.7 Temporal stretching factors

The second goal of the spatio-temporal algorithm that we are developing is to provide a control on the temporal error. As discussed in chapter 4, this is achieved by equidistributing the error in space and time. In other words, the time-step sizes will be stretched/coarsened in a way not to allow the temporal error along a temporal edge to exceed the global spatial equidistributed error. Therefore, the stretching factors to be imposed on a temporal edge  $[t_l^k, t_{l+1}^k]$  will be defined by:

$$\tau_l^k = \left( \frac{\varepsilon_{[T^k, T^{k+1}]}(N)}{\max_{X^i} (e_{l, l+1}^k)^i} \right)^{\frac{1}{2}} \quad (6.12)$$

where the temporal errors  $(e_{l,l+1}^k)^i$  are evaluated in exactly the same way as in chapter 4, i.e.

$$(e_{l,l+1}^k)^i = (\mathcal{S}_{k,ll+1})^i \Delta t_{k,l} \quad (6.13)$$

with

$$(\mathcal{S}_{k,ll+1})^i = (\mathcal{S}_{k,l+1})^i - (\mathcal{S}_{k,l})^i, \quad (6.14)$$

being the change in the solution's temporal gradient along the temporal edge  $[t_l^k, t_{l+1}^k]$  at a spatial node  $X^i$  and

$$(\mathcal{S}_{k,l})^i = \frac{(\mathcal{S}_{k,ll+1})^i \Delta t_{k,l} - (\mathcal{S}_{k,ll-1})^i \Delta t_{k,l-1}}{\Delta t_{k,l}^2 + \Delta t_{k,l-1}^2} \quad (6.15)$$

being the reconstructed temporal gradient at a spatial node  $X^i$  and temporal node  $t_l^k$ .

As a result, one obtains a set of stretching factors for the temporal edges  $[t_l^k, t_{l+1}^k]$  and consequently a stretching factor for the slab of time can be deduced:

$$\tau^k = \sum_{l=0, \dots, n_{\text{freq}}-1} \tau_l^k, \quad (6.16)$$

implying an optimal size of the time interval  $[T^k, T^{k+1}]$ :

$$\widetilde{\delta t}_k = \tau^k \delta t_k. \quad (6.17)$$

### 6.3.8 Space and time remeshing

We have discussed in the previous section the main steps constituting the prediction phase of the paradoxical meshing algorithm. Metric tensors  $\mathcal{M}_{[T^k, T^{k+1}]}^i$  have been generated prescribing at each spatial node  $X^i$  stretching factors along the different orientation (edges' directions) issued from that node for the time interval  $[T^k, T^{k+1}]$ . Moreover, a temporal error analysis accounting for the equi-distribution of the errors in space and time permitted the derivation of the optimal time-slab's stretching. The process of metric construction and computation of the temporal edges' stretching factors is carried out on the different time-slabs yielding a set of metric tensors  $\left\{ \mathcal{M}_{[T^k, T^{k+1}]}^i \right\}_k$  together with the corresponding stretching factors of the time slabs.

Based on the constructed metric tensors, a 2D/3D remeshing using the MTC mesher is performed for each slab of time resulting in a new mesh anisotropically adapted to control the  $L_\infty(t; L_p(\Omega))$  spatial error. In turn, the optimal lengths of the time-slabs  $\widetilde{\delta t}_k$ , which are in accordance with the optimal anisotropic spatial mesh, will be given to a 1D time mesher yielding a new discretization of the interval  $[0, T]$ . It is important to mention that the number of subintervals  $n_{\text{SI}}$  is automatically changed due to the 1D remeshing. The data structure of the temporal mesh will also be updated where the temporal nodes  $T^k$  will be redistributed and their number can be increased or reduced depending on the features of the problem. More details on the 1D time mesher will be

provided in the following section. Figure 6.2 portrays a possible optimal distribution of the temporal nodes.

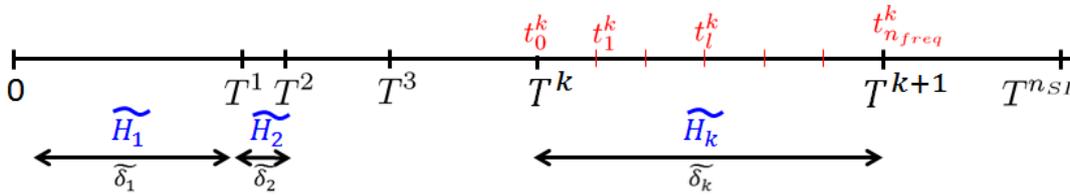


Figure 6.2 – Updated time slabs  $[T^{k-1}, T^k]$  and their corresponding optimal meshes.

### 6.3.9 Convergence of the space-time adaptive algorithm

To sum up, in this section, we have described the skeleton of the paradoxical meshing technique. We recall that the objective of this algorithm is to build up meshes that would hold for slabs of time. Each of these slabs will be constituted of  $n_{\text{freq}} + 1$  temporal nodes. Starting with an initial mesh  $\mathcal{H}_{\text{init}}$ , an initial time-slab size  $\delta_{\text{init}}$  and a prescribed frequency of adaptation, the method will generate, for a fixed number of nodes  $N$ , a set of optimal meshes  $\{\mathcal{H}\}_k$  and time-slab sizes  $\{\delta_k\}_k$ . We note that since an implicit time discretization is being used, the choice of the initial time-slab size will not affect the global stability of the simulation.

Given the initial mesh, time-slab size, and frequency of remeshing, the starting point consists in considering the whole simulation time, splitting it into  $n_{\text{SI}}$  subintervals using the prescribed initial time-slab size. These intervals will in turn be divided into  $n_{\text{freq}}$  subintervals delimited by temporal nodes  $t_k^l$ . The physical unsteady problem is solved on  $\mathcal{H}_{\text{init}}$  at the different temporal nodes  $t_k^l$ . Variables of interest for adaptation are collected over the computations. Afterward, each time-slab  $[T^k, T^{k+1}]$  is considered at a time, and edge-based error estimation is performed associating to each edge in the mesh  $\mathcal{H}_{\text{init}}$  a spatial error  $e_{ij}^s(t_k^l)$ . These errors will then be used to determine a global equi-distributed error  $\varepsilon_{[T^k, T^{k+1}]}(N)$  for a fixed number of nodes  $N$  and a set of stretching factors. Metric tensors will subsequently be constructed reflecting at each node  $X^i$  in  $\mathcal{H}_{\text{init}}$  the optimal stretching of the edges connected to it in their own direction. Moreover the sampled data will also be used to evaluate temporal errors  $(e_{i,l+1}^k)^i$ . The latter together with the estimated equi-distributed error yield stretching factors on the temporal edges and thus stretching factors on the time-slab  $[T^k, T^{k+1}]$ . The same computations of data sampling, metric construction and temporal stretching estimation are carried on the different time-slabs. Given the collected metrics, a new mesh will then be generated on each time-slab  $[T^k, T^{k+1}]$  controlling the  $L_\infty(t; L_p(\Omega))$  spatial interpolation error. We recall that the adaptation on the  $L_p(\Omega)$  norm of the interpolation errors raises the possibility to control the multiscales of the problem's characteristics. Furthermore, the set of generated optimal time-slab sizes will be provided to the time mesher yielding a 1D remeshing of the time-slabs.

Consequently, a first prediction of meshes and time-slabs has been conducted. Start-

ing now from the set of generated meshes and time-slabs a solution sampling is predicted followed by an error estimation, a metric construction, and an estimation of optimal time-slab sizes. Finally, a new set of meshes and time-slab sizes is regenerated. The whole process of prediction/remeshing/correction is iteratively repeated until the space and time meshes provide an optimal representation of the variables of interest.

Therefore, we have defined all the blocks that constitute the new adaptation algorithm. The only missing piece on the derivation checkerboard is the definition of a convergence criterion for the mesh/solution couple. Several possible convergence detectors can be adopted. The first one is based on a fixed number of iterations of the algorithm which is the choice made in the fixed point algorithm. Another possibility, which is adopted in this work, would be to consider a fixed tolerance on the equidistributed error. More precisely, convergence is reached when the relative change in the equidistributed error is less than a certain prescribed tolerance. Therefore, the convergence criterion reads as:

$$\frac{\varepsilon_{[T^k, T^{k+1}]}(N)^\zeta - \varepsilon_{[T^k, T^{k+1}]}(N)^{\zeta-1}}{\varepsilon_{[T^k, T^{k+1}]}(N)^\zeta} \leq \epsilon_{\text{TOL}} \quad \forall k = 1, \dots, n_{\text{SI}} - 1 \quad (6.18)$$

where  $\zeta$  is the algorithm's iteration number. Indeed, the convergence of the global equidistributed error means that the mesh has converged to the optimum for a given fixed number of nodes. Furthermore, since this error controls as well the time-step sizes, this means that the time-adaptive algorithm converged as well. At convergence, computations are resumed on the predicted optimal set of meshes  $\rightarrow \widetilde{\mathcal{H}}_k^\zeta$  with the corresponding set of time-step sizes  $\widetilde{\mathcal{T}}^{\zeta-1}$ . In the following numerical computations, we set the convergence parameter to 0.01. In other words, we assume the the couple mesh/solution has converged when the change in the mesh is less than 1%.

## 6.4 PARADOXICAL MESHING ALGORITHM

Algorithm 5 retraces the steps of anisotropic mesh and time-step generation. The method can be easily implemented and parallelized. This technique ensures a good level of accuracy in particular for problems involving steep variations and sharp evolution in time.

**Remark 40.** *We note that the proposed approach can also be applied in the context of physical problems involving moving geometry using the immersed volume formulation. The solid's interface is rendered by the zero-isovalue of the levelset function that is moved over time. Using the paradoxical meshing, the object's interface will emerge into a pre-adapted zone and therefore will be well represented all over the simulation.*

Using the single time-step adaptation, the mesh is dynamically adapted at a specific frequency. Nevertheless, it will not be predicted for an interval of time, thus yielding small time-step sizes in order to maintain the evolving interface in a well adapted

region. This will imply a considerable computational cost, as it will be shown in the numerical results. The paradoxical meshing, on the other hand, permits a smoother movement of the geometry as the well refined zones of the mesh form a wide layer in the movement direction.

We have therefore established all the steps for the development and implementation of the paradoxical meshing algorithm. We note that the algorithm does not require to work on the whole interval of time. It can be applied on subintervals of the simulation time. That way, there is no need to wait for the end of the iterative process on the different slabs of time in order to perform the computations on the generated optimal meshes and evolve the solution with the optimal time-step sizes. On the other hand, the method can also be applied in parallel with the single time-step adaptation approach. When the computed global error is relatively large, reflecting large estimated errors along the edges of the spatial mesh which in turn highlight important variations in the solution between two successive spatial remeshing. At that stage, the paradoxical technique can be activated to construct the appropriate optimal mesh and time-step sizes over the time interval.

As it was already mentioned a keystone step in the paradoxical meshing algorithm is the time remeshing. We point out that it is possible to use any mesher in order to remesh the time-interval of simulation. However, the algorithm for time-remeshing that we develop in the following section is well devised for time-advancing, simple and easy to implement.

## 6.5 1D TEMPORAL MESHER

In this chapter, we have developed a new algorithm for anisotropic space and time adaptation. The method relies on a predictor-corrector formulation whereby a set of meshes  $\widetilde{\mathcal{H}}_k$  and their corresponding optimal time-step sizes  $\widetilde{\delta t}_k$  are generated. These predicted meshes are designed to contain the evolving solution over slabs of time. The most trivial approach for running the computations is to consider the meshes  $\widetilde{\mathcal{H}}_k$  one after the other and to advance the solution over the predicted slab  $\widetilde{\delta t}_k$  at the predefined frequency. However, as illustrated in figure 6.3, this approach is inadequate. Indeed, a mesh that is optimally generated to contain the solution emerging from time  $T^k$  and can optimally hold for a time-slab of size  $\widetilde{\delta t}_k$ , may not be optimal at time  $T^p$  for an interval of size  $\widetilde{\delta t}_k$ . For instance, we can clearly see in the example presented in figure 6.3 that letting the mesh  $\widetilde{\mathcal{H}}_{k-1}$  valid on an interval of length  $\widetilde{\delta t}_k$  would make the mesh  $\widetilde{\mathcal{H}}_k$  be used as of time  $T^p$  whereas this mesh was designed to contain the solution's evolution starting from time  $T^k$  and ending at  $T^k + \widetilde{\delta t}_k$ . Therefore careful generation shall be developed yielding appropriate temporal nodes' distribution.

We aim in this section at developing a new 1D mesher that is well-suited for time adaptation. Given the initial temporal mesh configuration, the algorithm shall be able to redistribute the nodes, insert new ones and remove other ones based on the estimated optimal time-step sizes.

---

**Algorithm 5:** Paradoxical space-time adaptive algorithm based on a frequency of adaptation for unsteady problems.

---

**Input:** An initial mesh  $\mathcal{H}_{\text{init}}$ .

An initial time-slab size  $\delta t_{\text{init}}$ .

A user-prescribed mesh complexity  $N$ .

A user-prescribed frequency of adaptation  $n_{\text{freq}}$ .

**Output:** A set of meshes lasting  $n_{\text{freq}}$  time-steps with their corresponding time-slab sizes.

**begin**

Initialization: temporal mesh  $\mathcal{T}$ , time slabs, spatial meshes on these slabs.

**while** *not converged* **do**

**for**  $k : 0 \rightarrow n_{\text{SI}} - 1$  : *loop over the slabs of time* **do**

**for**  $l : 0 \rightarrow n_{\text{freq}} - 1$  : *loop over the temporal nodes* **do**

Consider the mesh  $\mathcal{H}_k$ .

Advance in time with  $\Delta t_{k,l} = \frac{\delta t_k}{n_{\text{freq}}}$ .

Solve the physical problem and collect the variables of interest  $S_{k,l}$ .

**if**  $(k > 0 \ \& \ l == 0)$  **then**

Determine the temporal error at the previous temporal node:

$$\max_{X^i} \left( e_{n_{\text{freq}}-1, n_{\text{freq}}}^{k-1} \right)^i.$$

Deduce the temporal stretching factor  $\tau_{n_{\text{freq}}-1}^{k-1}$ .

Update  $\widetilde{\delta t}_{k-1}$ .

Compute the edge-based errors  $e_{ij}^s(t_k^l)$  and track the  $L_\infty(t; L_p(\Omega))$  interpolation error's estimates.

**if**  $((k, l) > (0, 1))$  **then**

$$\left[ \text{Compute the temporal error } \max_{X^i} \left( e_{l, l-1}^k \right)^i. \right.$$

Compute the equidistributed error  $\varepsilon_{[T^k, T^{k+1}]}(N)$ .

Determine the corresponding stretching factors for the mesh edges.

Construct the metric tensors  $\mathcal{M}_{[T^k, T^{k+1}]}^i$  on the nodes of the mesh.

Determine the stretching of the temporal edges  $\tau_{l=1, \dots, n_{\text{freq}}-1}^k$ .

Update  $\widetilde{\delta t}_k$ .

Spatial remeshing  $\rightarrow \widetilde{\mathcal{H}}_k^\zeta$ .

Check for convergence.

**if** *not converged* **then**

Temporal remeshing  $\rightarrow \widetilde{\mathcal{T}}^\zeta$ .

Update the set of optimal meshes.

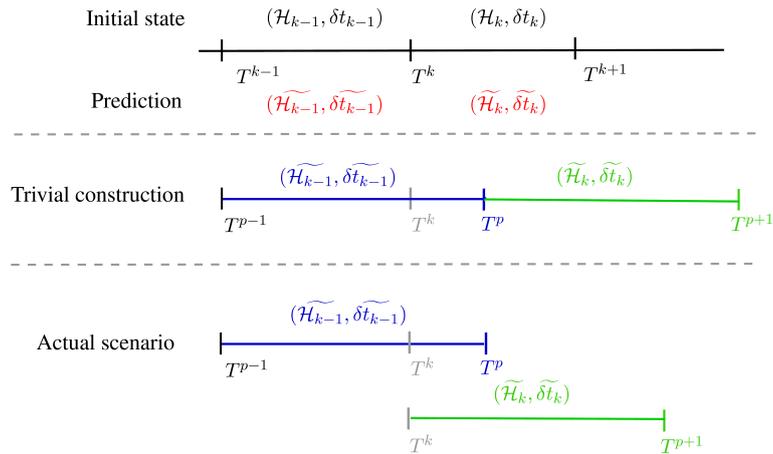


Figure 6.3 – Example on the failure of the trivial time advancing.

The algorithm will iterate over the time-slabs in the initial temporal mesh and decide for each one of them whether to shrink it, stretch it or keep it the same.

For illustration purposes, we consider two successive time-slabs  $[T^{k-1}, T^k]$  and  $[T^k, T^{k+1}]$  with their optimal meshes and time-slab sizes  $(\widetilde{\mathcal{H}}_{k-1}, \widetilde{\delta t}_{k-1})$  and  $(\widetilde{\mathcal{H}}_k, \widetilde{\delta t}_k)$ . We assume that the algorithm is considering the slab  $[T^{k-1}, T^k]$  in the initial temporal mesh. Without loss of generality, we assume also that the node  $T^{k-1}$  will be maintained in the new mesh and corresponds to node  $T^{p-1}$ . Several possible scenarios can be identified. In what follows, we will detail how to treat each one of them. We denote by  $\{T^k\}_k$  the nodes belonging to the old temporal mesh and  $\{T^p\}_p$  the nodes of the new temporal mesh.

- **Scenario 1** In this scenario, we consider that the temporal error is larger than the equi-distributed error over the time-slab  $[T^{k-1}, T^k]$ . In that case, the optimal time-slab size will be smaller than the initial one. To control the temporal error on the following iteration of the paradoxical meshing algorithm, new nodes will be inserted in this interval yielding new time-slabs as shown in figure 6.4.

The principle of temporal mesh slab insertion follows the sub-routine defined in algorithm 6. According to this algorithm, new time-slabs of size  $\widetilde{\delta t}_k$  will be added

---

**Algorithm 6:** Mesh slab insertion subroutine.

---

**Data:** The interval  $[T^{k-1}, T^k]$  and the generated optimal time-slab size  $\widetilde{\delta t}_{k-1}$

**begin**

```

Set  $T^{p-1} = T^{k-1}$ 
while  $(T^{p-1} + \widetilde{\delta t}_k) \leq T^k$  do
   $T^p \leftarrow T^{p-1} + \widetilde{\delta t}_k$ .
  Update  $p$ :  $p \leftarrow p + 1$ .
if  $(T^{p-1} < T^k)$  then
   $T^p \leftarrow T^k$ .

```

---

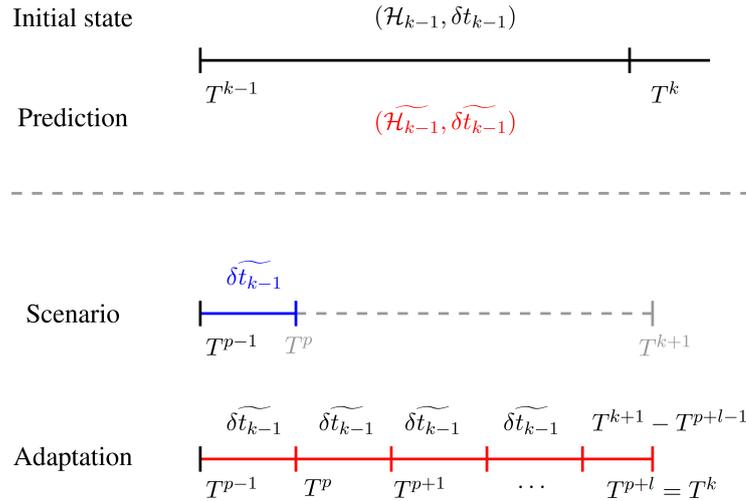


Figure 6.4 – Scenario 1: the optimal time-slab size is smaller than the initial one. Consequently, new slabs will be generated in the mesh.

to the temporal mesh and it is also possible to have an additional subinterval of size  $T^k - T^p$  where  $T^p$  is the last inserted node before  $T^k$ . Depending on the size of this interval, it is either kept the same or modified. The modification can be done through merging it with the previous subinterval or summing both intervals and then splitting the resulting interval into two equal subintervals. The optimal mesh  $\widetilde{\mathcal{H}}_{k-1}$  will be maintained over each of these slabs in the following iteration of the paradoxical algorithm.

- **Scenario 2** The second scenario, as illustrated in figure 6.5, takes into account the cases when the initial time-slab size  $\widetilde{\mathcal{H}}_{k-1}$  is optimal, i.e.  $\widetilde{\delta t}_{k-1} = \delta t_{k-1}$ . Thus, no change needs to be made on the nodes. Consequently, we set  $T^{p-1} = T^{k-1}$  and  $T^p = T^k$ .

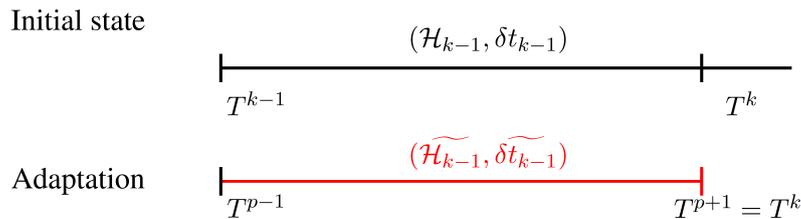


Figure 6.5 – Scenario 2: the optimal time-slab size is equal to the initial one.

- **Scenario 3** The predicted validity interval of mesh  $\widetilde{\mathcal{H}}_{k-1}$ , starting at node  $T^{p-1} = T^{k-1}$ , contains the node  $T_k$ . Two possibilities arise in that case as shown in figure 6.6.

In the first possibility, only  $T^k$  is contained inside  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ . Then, we examine the interval  $[T^k, T^{k+1}]$ . Depending on  $\widetilde{\delta t}_k$  either this interval should be shrunk, stretched or remain the same. If it should remain the same or be stretched i.e.  $\widetilde{\delta t}_k \geq \delta t_k$ , or even be shrunk with its endpoint remaining outside  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ , an additional node is added in the new mesh and corresponds to  $T^p = T^{p-1} + \widetilde{\delta t}_{k-1}$ , whereas the node  $T^k$  will not be introduced into the

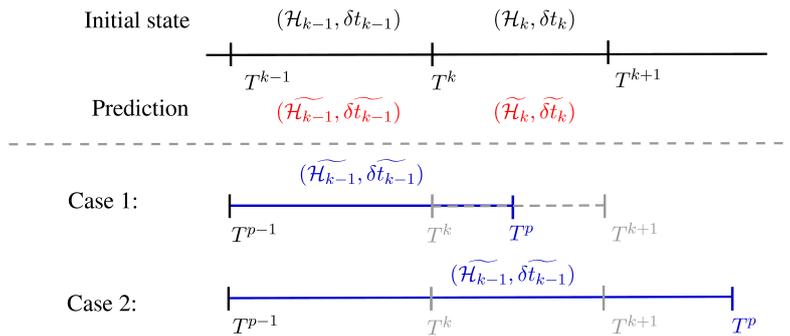


Figure 6.6 – Scenario 3: The time-slab  $[T^{k-1}, T^k]$  will be stretched.

new temporal mesh. The spatial mesh  $\widetilde{\mathcal{H}}_{k-1}$  will be employed to evolve the solution at the following iteration of the paradoxical meshing algorithm over the time slab  $[T^{p-1}, T^p]$ . Figure 6.7 sketches this situation and the adequate remeshing.

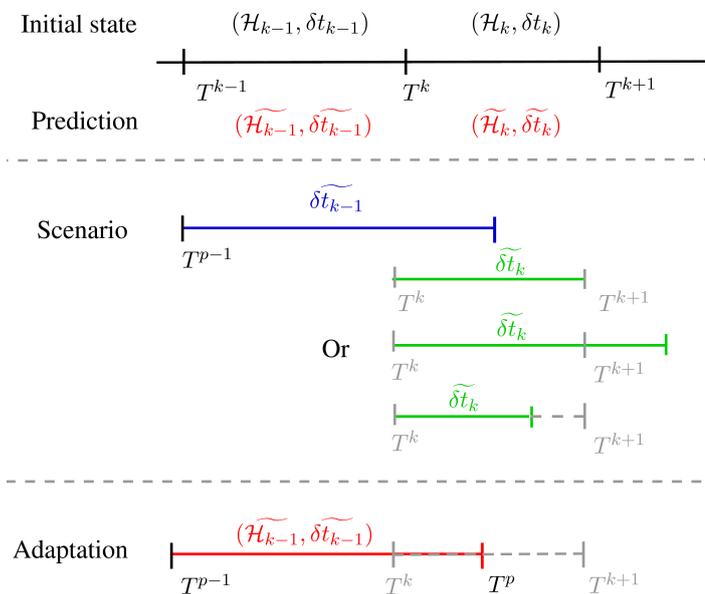


Figure 6.7 – Scenario 3 (a): The time-slab  $[T^{k-1}, T^k]$  will be stretched with  $T^{k+1}$  and the new endpoint of the predicted optimal slab  $[T^k, T^k + \widetilde{\delta t}_k]$  lying outside  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ .

On the other hand, as illustrated in figure 6.8, if the time-slab size is to be shrunk with  $(T^k + \widetilde{\delta t}_k) < (T^{p-1} + \widetilde{\delta t}_{k-1})$ , that means a considerable temporal error is induced over  $[T^k, T^{k+1}]$ , the node  $T^p = T^k$  will be added to the new temporal mesh. The spatial mesh  $\widetilde{\mathcal{H}}_{k-1}$  will be considered to evolve the solution over  $[T^{p-1}, T^p]$  in the following iteration of the paradoxical meshing algorithm. However, if  $(T^k + \widetilde{\delta t}_k) \geq (T^{p-1} + \widetilde{\delta t}_{k-1})$ , a new node  $T^p = (T^{p-1} + \widetilde{\delta t}_{k-1})$  will be inserted.

In the second case, the mesh  $\widetilde{\mathcal{H}}_{k-1}$  is well prepared to contain the evolving solution for a  $\widetilde{\delta t}_{k-1}$  time-slab, therefore, there is no need to change the mesh at time  $T^k$ . In this scenario, both  $T^k$  and  $T^{k+1}$  belong to the predicted optimal interval  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ . If additionally, the interval  $[T^k, T^{k+1}]$  is to be stretched with  $(T^k + \widetilde{\delta t}_k) \geq (T^{p-1} + \widetilde{\delta t}_{k-1})$ , the nodes  $T^k$  and  $T^{k+1}$  will not be inserted in the new mesh and the analysis is carried out for the following interval  $[T^{k+1}, T^{k+2}]$ .

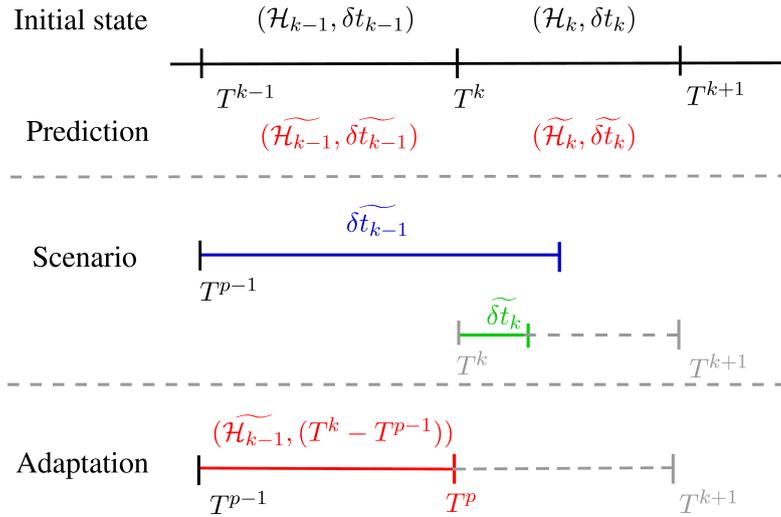


Figure 6.8 – Scenario 3 (b): The time-slab  $[T^{k-1}, T^k]$  will be stretched with  $T^k$  and the new endpoint of the predicted optimal slab  $[T^k, T^k + \widetilde{\delta t}_k]$  lying inside  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$ .

All the intervals falling inside  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$  and that are predicted to be stretched and going beyond  $T^{p-1} + \widetilde{\delta t}_{k-1}$  will be omitted from the new temporal mesh together with their corresponding optimal meshes. The process will stop at index  $m$  for which  $T^{k+m}$  is within  $[T^{p-1}, T^{p-1} + \widetilde{\delta t}_{k-1}]$  whereas  $T^{k+m}$  is beyond this interval. In this case, the same judgment described in the first possibility applies. That is if the time-slab  $[T^{k+m}, T^{k+m+1}]$  is to be shrunk with  $(T^{p-1} + \widetilde{\delta t}_{k-1}) > (T^{k+m} + \widetilde{\delta t}_{k+m+1})$ , then the node  $T^p = T^{k+m}$  will be added to the new temporal mesh and the mesh  $\widetilde{\mathcal{H}}_{k-1}$  will be used to evolve the solution on the time-slab  $[T^{p-1}, T^p]$  at the following iteration of the paradoxical algorithm. The time remeshing then resumes at time-slab  $[T^{k+m}, T^{k+m+1}]$ . On the other hand, if the time-slab  $[T^{k+m}, T^{k+m+1}]$  is to remain the same, be stretched or even be shrunk with  $(T^{p-1} + \widetilde{\delta t}_{k-1})$  lying inside the predicted optimal interval, a node  $T^p = (T^{p-1} + \widetilde{\delta t}_{k-1})$  is added to the new temporal mesh and the spatial mesh  $\widetilde{\mathcal{H}}_{k-1}$  will be used to evolve the solution on the time-slab  $[T^{p-1}, T^p]$  at the following iteration of the paradoxical algorithm. Then the time remeshing resumes at time-slab  $[T^{k+m}, T^{k+m+1}]$ . Figure 6.9 shows an example of this scenario with  $m = 2$ .

Another possibility to stop the process is to have a time-slab  $[T^{k+m}, T^{k+m+1}]$  that will be contained inside  $T^{p-1} + \widetilde{\delta t}_{k-1}$  after adaptation. That is, in the error estimation, the time adaptation algorithm detected a temporal error higher than the estimated equi-distributed error. Consequently, all the previous temporal nodes in the old mesh will not be included in the new one and their corresponding meshes will be omitted at the next iteration of the paradoxical meshing algorithm. The node  $T^p = T^{k+m}$  will be added to the new temporal mesh and the mesh  $\widetilde{\mathcal{H}}_{k-1}$  will be used to evolve the solution on the time-slab  $[T^{p-1}, T^p]$  at the following itera-

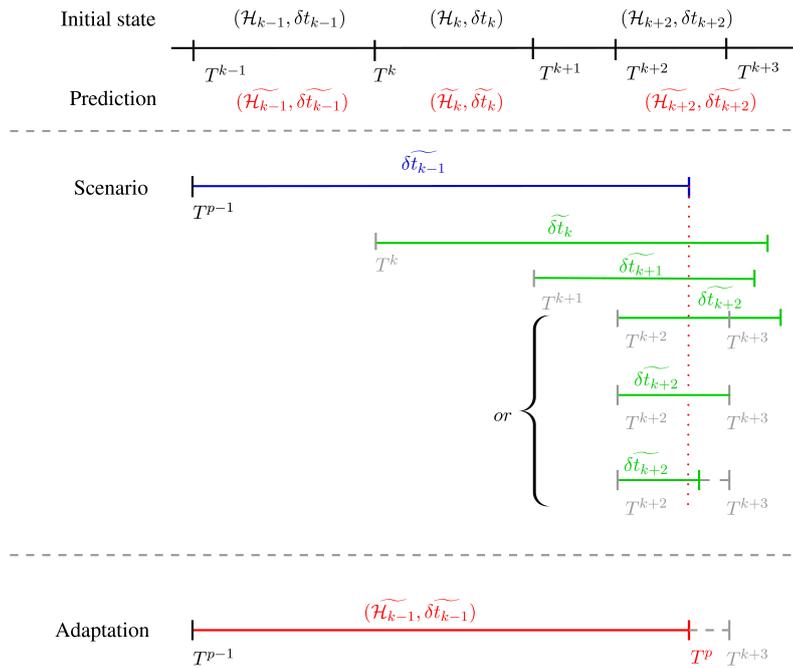


Figure 6.9 – Scenario 3 (c): The time-slab  $[T^{k-1}, T^k]$  will be stretched with  $T^k, T^{k+1}, T^{k+2}$  lying inside  $[T^{p-1}, T^{p-1} + \delta \widetilde{t}_{k-1}]$ .

tion of the paradoxical algorithm. The time remeshing then resumes at time-slab  $[T^{k+m}, T^{k+m+1}]$ .

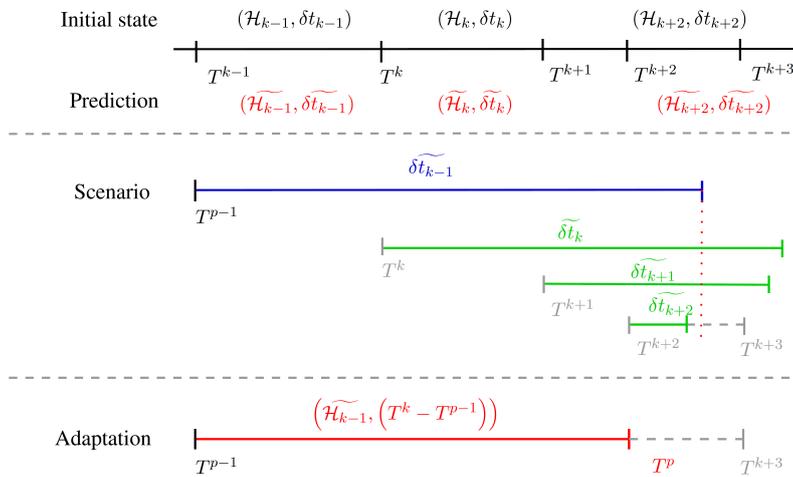


Figure 6.10 – Scenario 3 (d): The time-slab  $[T^{k-1}, T^k]$  will be stretched with  $T^k, T^{k+1}, T^{k+2}$  and  $T^{k+2} + \delta \widetilde{t}_{k+2}$  lying inside  $[T^{p-1}, T^{p-1} + \delta \widetilde{t}_{k-1}]$ .

- **Scenario 4** This scenario, sketched in figure 6.11, examines the case when the generated optimal mesh is valid from  $T^{p-1} = T^{k-1}$  to the end of the simulation time. This occurs when the solution of the problem has converged to a steady state. In that case, all the following nodes in the temporal mesh will be removed and  $T^p$  is set to the final simulation time  $T$ . Furthermore, the remaining meshes will also be removed and will not be used in the second run of the paradoxical algorithm. We note that this scenario is a particular case of scenario 3.

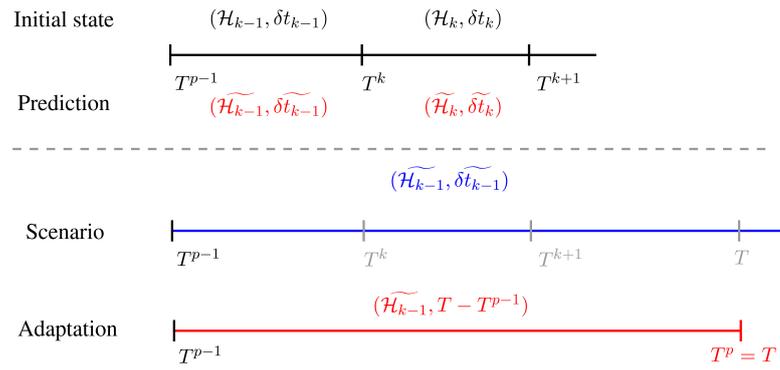


Figure 6.11 – Scenario 4: the generated optimal mesh  $\widetilde{\mathcal{H}}_{k-1}$  up to the end of the simulation.

**Remark 41.** For simplicity of elucidation, we have considered that  $T^{k-1}$  is inserted in the new mesh nevertheless in practice  $T^{p-1}$  need not coincide with  $T^{k-1}$ . For this case, instead of considering an optimal time-slab size  $\widetilde{\delta t}_{k-1}$ , we use a slab size  $(\widetilde{\delta t}_{k-1} - (T^{p-1} - T^k))$  and apply the same analysis suggested previously.

Finally, in this section we have evoked the different possibilities addressed in the temporal mesher developed in this work. We recall that this algorithm permits the redistribution, the insertion and the removal of nodes in the 1D time mesh. We summarize in algorithm 7 the general structure of the temporal mesher.

---

**Algorithm 7:** Time meshing of the interval  $[0, T]$  in accordance with the estimated optimal slab sizes.

---

**Input:** The vector of initial time slab sizes  $\mathcal{T}_{\text{initial}} = \{\delta t_k\}_k$ .  
The vector of optimal time slab sizes  $\mathcal{T}_{\text{optimal}} = \{\widetilde{\delta t}_k\}_k$ .  
The vector of previous temporal mesh nodes  $\mathcal{T}_{\text{initial}} = \{T^k\}_k$ .  
The set of corresponding optimal meshes  $\mathcal{H}_{\text{optimal}}$

**Output:** The vector of new temporal mesh nodes  $\mathcal{T}_{\text{new}} = \{T^k\}_k$ .  
The vector of new time slab sizes  $\mathcal{T}_{\text{new}} = \{\delta t_k\}_k$ .  
The set of corresponding meshes  $\mathcal{H}_{\text{new}}$

**begin**

Initialization:

Define the first temporal node:  $\mathcal{T}_{\text{new}}[0] = \mathcal{T}_{\text{initial}}[0]$ .

Define a counter, *count*, that stands for the position in the vector of temporal nodes.

Initialize the counter: *count* = 1.

**for**  $k : 0 \leftarrow n_{SI} - 1$  : loop over the slabs of time, **do**

- Check for Scenario 1 using algorithm 8,
- Check for Scenario 2 using algorithm 9,
- Check for Scenario 3 and 4 using algorithm 10.
- Adapt the temporal mesh accordingly.

**end**

---

---

**Algorithm 8:** Subroutine representing scenario 1 in the main time meshing algorithm 7.

---

**Scenario 1:** treats the case where the optimal time slab size is smaller than the old one.

```

if ( $\mathcal{T}_{\text{optimal}}[k] < \mathcal{T}_{\text{initial}}[k]$ ) then
  for  $i : 1 \leftarrow \lfloor \frac{\mathcal{T}_{\text{initial}}[k]}{\mathcal{T}_{\text{optimal}}[k]} \rfloor$  do
    Fill in the large non-optimal slab of initial size  $\mathcal{T}_{\text{initial}}[k]$  time with smaller
    slabs  $\mathcal{T}_{\text{optimal}}[k]$ .
     $\mathcal{T}_{\text{new}}[\text{count}] = \mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k]$ .
     $\mathcal{T}_{\text{new}}[\text{count} - 1] = \mathcal{T}_{\text{optimal}}[k]$ .
    Associate the corresponding mesh  $\mathcal{H}_{\text{new}}[\text{count} - 1] = \mathcal{H}_{\text{optimal}}[k]$ .
    Increment the counter:  $\text{count} = \text{count} + 1$ .
  // If it cannot be evenly filled with slabs of size  $\mathcal{T}_{\text{optimal}}[k]$ : define the last one of size
   $\mathcal{T}_{\text{initial}}[k + 1] - \mathcal{T}_{\text{new}}[\text{count} - 1]$ 
  if ( $\mathcal{T}_{\text{new}}[\text{count} - 1] < \mathcal{T}_{\text{initial}}[k + 1]$ ) then
     $\mathcal{T}_{\text{new}}[\text{count}] = \mathcal{T}_{\text{initial}}[k + 1]$ .
     $\mathcal{T}_{\text{new}}[\text{count} - 1] = \mathcal{T}_{\text{new}}[\text{count}] - \mathcal{T}_{\text{new}}[\text{count} - 1]$ .
    Optimize the last interval if needed by merging with the previous one or
    by summing them and then dividing by two.
    Associate the corresponding mesh  $\mathcal{H}_{\text{new}}[\text{count} - 1] = \mathcal{H}_{\text{optimal}}[k]$ .
    Increment the counter:  $\text{count} = \text{count} + 1$ .

```

---



---

**Algorithm 9:** Subroutine representing scenario 2 in the main time meshing algorithm 7.

---

**Scenario 2:** treats the case where the optimal time slab size is equal to the old one.

```

if ( $\mathcal{T}_{\text{optimal}}[k] == \mathcal{T}_{\text{initial}}[k]$ ) then
   $\mathcal{T}_{\text{new}}[\text{count}] = \mathcal{T}_{\text{initial}}[k + 1]$ .
   $\mathcal{T}_{\text{new}}[\text{count} - 1] = \mathcal{T}_{\text{new}}[\text{count}] - \mathcal{T}_{\text{new}}[\text{count} - 1]$ .
  Associate the corresponding mesh  $\mathcal{H}_{\text{new}}[\text{count} - 1] = \mathcal{H}_{\text{optimal}}[k]$ .
  Increment the counter:  $\text{count} = \text{count} + 1$ .

```

---

---

**Algorithm 10:** Subroutine representing scenario 3 and 4 in the main time meshing algorithm 7.

---

**Scenario 3:** treats the case where the optimal time slab size is larger than the old one.

**if** ( $\mathcal{T}_{\text{optimal}}[k] > \mathcal{T}_{\text{initial}}[k]$ ) **then**

**if** ( $(\mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k]) \geq T$ ) **then**

*The problem has converged to a steady state.*

**Scenario 4:**

        Stop looping and set:

$\mathcal{T}_{\text{new}}[\text{count}] = T$ .

$\mathcal{T}_{\text{new}}[\text{count} - 1] = \mathcal{T}_{\text{new}}[\text{count}] - \mathcal{T}_{\text{new}}[\text{count} - 1]$ .

        Associate the corresponding mesh  $\mathcal{H}_{\text{new}}[\text{count} - 1] = \mathcal{H}_{\text{optimal}}[k]$ .

**else**

        Define a temporary index, *Index*, that accounts for the nodes lying inside the optimal time-slab.

        Set the temporary index to  $k + 1$ :  $\text{Index} \leftarrow k + 1$ .

**while** ( $\mathcal{T}_{\text{initial}}[\text{Index}] < (\mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k])$ ) **do**

            The time slab  $[\mathcal{T}_{\text{initial}}[\text{Index} - 1]; \mathcal{T}_{\text{initial}}[\text{Index}]$ ] contains the node

$\mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k]$ .

            Increment the indexing:  $\text{Index} = \text{Index} + 1$ .

**if** ( $\mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k] > (\mathcal{T}_{\text{initial}}[\text{Index} - 1] + \mathcal{T}_{\text{optimal}}[\text{Index} - 1])$ )

**then**

            This is the case when the interval  $[\mathcal{T}_{\text{new}}[\text{Index} - 1], \mathcal{T}_{\text{new}}[\text{Index}]$ ] will be shrunk and  $\mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k]$  will fall outside the predicted optimal slab.

$\mathcal{T}_{\text{new}}[\text{count}] = \mathcal{T}_{\text{initial}}[\text{Index} - 1]$ .

$\mathcal{T}_{\text{new}}[\text{count} - 1] = \mathcal{T}_{\text{initial}}[\text{Index} - 1] - \mathcal{T}_{\text{new}}[\text{count} - 1]$ .

            Associate the corresponding mesh  $\mathcal{H}_{\text{new}}[\text{count} - 1] = \mathcal{H}_{\text{optimal}}[k]$ .

            Reset  $k$ :  $k = \text{Index} - 2$ .

**else**

$\mathcal{T}_{\text{new}}[\text{count}] = \mathcal{T}_{\text{new}}[\text{count} - 1] + \mathcal{T}_{\text{optimal}}[k]$ .

$\mathcal{T}_{\text{new}}[\text{count} - 1] = \mathcal{T}_{\text{optimal}}[k]$ .

            Associate the corresponding mesh  $\mathcal{H}_{\text{new}}[\text{count} - 1] = \mathcal{H}_{\text{optimal}}[k]$ .

            Reset  $k$ :  $k \leftarrow \text{Index} - 2$ .

**if**  $\mathcal{T}_{\text{new}}[\text{count}] == \mathcal{T}_{\text{initial}}[\text{Index}]$  **then**

            The created node coincides with the endpoint of a slab of time.

            Skip the interval  $[\mathcal{T}_{\text{initial}}[\text{Index} - 1]; \mathcal{T}_{\text{initial}}[\text{Index}]$ ].

            Set  $k = k + 1$ .

**else**

$\mathcal{T}_{\text{optimal}}[k + 1] = \mathcal{T}_{\text{optimal}}[k + 1] - (\mathcal{T}_{\text{new}}[\text{count}] - \mathcal{T}_{\text{initial}}[k + 1])$ .

$\mathcal{T}_{\text{initial}}[k + 1] = \mathcal{T}_{\text{initial}}[k + 2] - \mathcal{T}_{\text{new}}[\text{count}]$ .

        Increment the counter:  $\text{count} = \text{count} + 1$ .

---

### 6.5.1 Application of the Time mesher

Before proceeding into the numerical evaluation of the paradoxical meshing algorithm, let us consider an example to validate the 1D temporal mesh generation. As depicted in figure 6.12, starting with an initial mesh:

$$\mathcal{T}_{\text{initial}} = \{ 0, 0.1, 0.15, 0.17, 0.18, 0.38, 0.68, 1.08, 1.18 \} ,$$

initial time-step sizes:

$$T_{\text{initial}} = \{ 0.1, 0.05, 0.02, 0.01, 0.2, 0.3, 0.4, 0.1 \} ,$$

and predicted optimal time-step sizes:

$$T_{\text{optimal}} = \{ 0.15, 0.2, 0.1, 0.01, 0.1, 0.2, 0.45, 0.1 \}$$

Using the 1D temporal mesher, the temporal nodes will be modified, some of them will be removed and new ones will be inserted. The resulting temporal nodes are:

$$\mathcal{T}_{\text{new}} = \{ 0, 0.15, 0.25, 0.28, 0.31, 0.34, 0.37, 0.38, 0.68, 1.13, 1.18 \} ,$$

As it can be inferred from figure 6.12, the time-slab sizes are different from the predicted optimal sizes and do not necessarily coincide with the initial time-slabs.

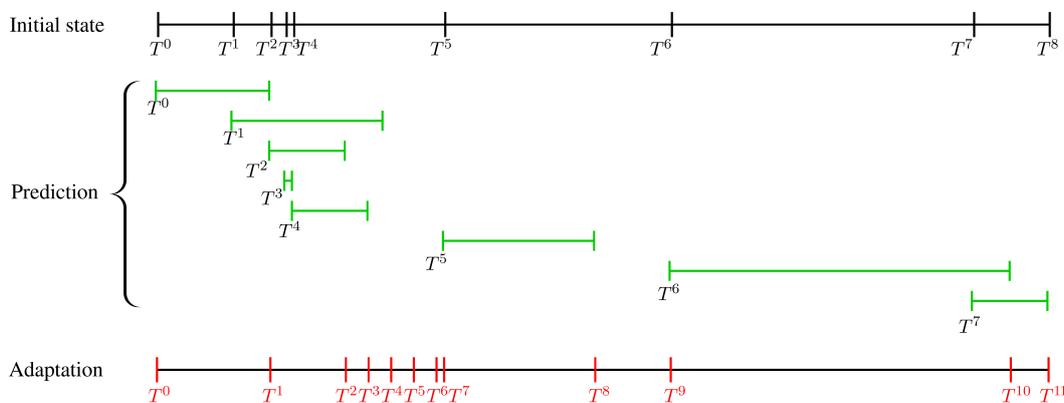


Figure 6.12 – Example of application of the time mesher.

The novel method, called paradoxical meshing, that we presented in this chapter is perceived not only as a fully adaptive technique but also as a space and time accurate way of solving time-dependent problems within reasonable computational costs. Indeed it avoids the computation of a new mesh at every solver iteration and hence the drastic increase of the CPU time. Furthermore, fewer function interpolation are performed lowering the accumulated interpolation errors. The new metric for several time-steps ensures that the error on each edge of the domain remains bounded by the global error. Nevertheless the mesh is not optimal for one solution at a time but optimal for the whole interval of time. A new temporal meshing algorithm is devised.

It allows the redistribution of the temporal nodes with respect to prescribed time-slab sizes. We aim in the following section to validate the proposed approach on several numerical examples where we compare the performance of the paradoxical technique to the classical one and highlight the accuracy and efficiency of the former.

The space-time adaptive algorithm was fully detailed, allowing for an easy, straightforward implementation in any similar context.

## 6.6 NUMERICAL EXAMPLES

In this section, we assess the performance of the newly developed fully-adaptive algorithm on several test cases and compare the result with the single time-step space-time adaptive techniques. The numerical experiments will show that the developed algorithm together with the stabilized finite element method are capable of producing accurate results within a reasonable execution time and a low computational cost. In particular, the boundary layers and the vortices are well captured by the mesh.

### 6.6.1 A two-dimensional rotating Circle

We begin our validation of the fully adaptive technique by evaluating its interface tracking ability. We consider a function representing the rotation of a 2D circular interface of radius 0.1 initially located at (0.5, 0.5) in the computational domain  $[-1, 1] \times [-1, 1]$ :

$$u(x, t) = \mathcal{E} \tanh \left( (x - x(t))^2 + (y - y(t))^2 \right) \quad (6.19)$$

where  $\mathcal{E} = 0.0005$  is a stiffness parameter defining the thickness of the interface. The smaller  $\mathcal{E}$  is, the sharper the function's gradient. The simulation consists in rotating the circle in the counterclockwise direction at the rate  $\theta = 1 \text{ rad/s}$ . The objective is to test the capability of the anisotropic paradoxical meshing technique to accurately capture the dynamically evolving interface. We note that this is a very challenging example, if a single time-step mesh adaptation were to be used, the thickness of the refined layer in the vicinity of the function's high gradient is  $2\mathcal{E}$  and consequently if the time-step size is not small enough to maintain the evolving gradient within the refined zone, a loss of accuracy would result.

Figure 6.13 shows the adapted mesh, made up of 10,000 nodes, for the slab of time  $[t^n, t^{n+1}]$  gathering 10 sub-intervals of time. We can clearly see how refined the mesh is at the location of high function's gradient and how accurate the capture of the interface is as it rotates from time  $t^n$  (left) to time  $t^{n+1}$  (right). The elements all along the interface are isotropic yielding a well respected curvature. The algorithm progressively detects and refines the mesh at the interface when the circle advances in time and moves throughout the domain. We can detect how for a controlled number of nodes, the mesh is naturally, automatically, and anisotropically coarsened in one region (far from the interface) with the goal of reducing the mesh size at the location of interest (at the moving interface level). The time intervals' lengths  $[t^n, t^{n+1}] = [t_0^n, t_1^n, \dots, t_i^n, \dots, t_{10}^{n+1}]$  are

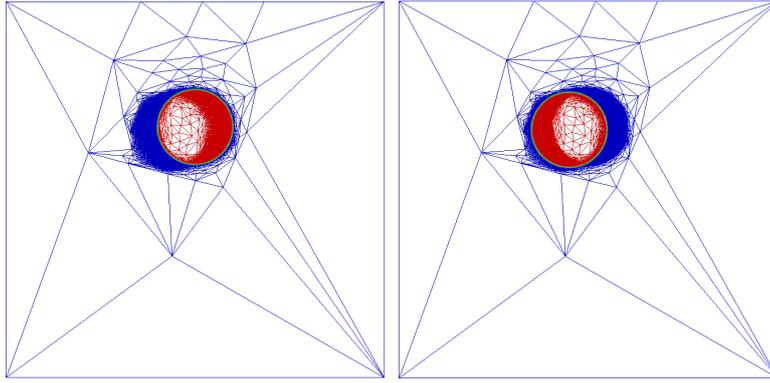


Figure 6.13 – Paradoxical meshing algorithm applied to a rotating circle for the interval  $[t^n, t^{n+1}]$ .

automatically generated by the paradoxical meshing algorithm and the corresponding optimal meshes are valid for the 10 consecutive time-steps. Figure 6.14(right) presents the time-step sizes for the first few iterations of the algorithm. We can clearly see the periodic profile as the time-steps vary periodically and oscillate around a constant value  $7.6 \times 10^{-3}$ . This periodic variation is in good agreement with the nature of the problem, as the circle rotates at a constant rate and maintains the same behavior all over computations. We recall that within the time-slab of size  $\delta t^n$ , the algorithm generates equally distributed sub-intervals of size  $\Delta t_l^n = \frac{\delta t^n}{10}$ .

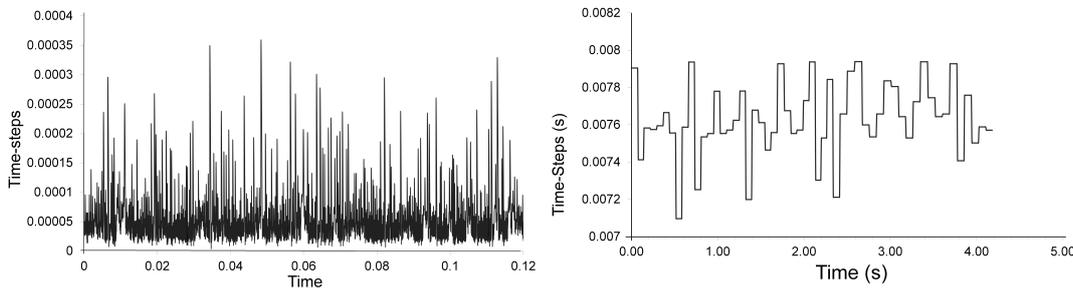


Figure 6.14 – Time-steps generated by the single time-step adaptation (left) and the paradoxical (right) meshing algorithms.

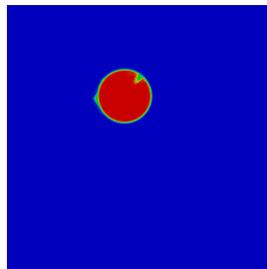


Figure 6.15 – Instabilities appearing when adapting the mesh using the single time-step meshing algorithm applied to a rotating circle with a fixed time-step  $\Delta t = 0.01$ .

Using the single time-step mesh adaptation technique at every solver iteration will keep the mesh in phase with the rotating interface, nevertheless, beside the considerable computational cost, an important accumulation of interpolation error will result, hence affecting the accuracy of the solution. Therefore such an algorithm is not the optimal one when dealing with time-dependent simulations. To reduce the mentioned

drawbacks, the mesh should be adapted less frequently. However, if no time adaptation is applied, the mesh would at some point fail to well capture the solution that is advancing in time and instabilities will appear as shown in figure 6.15. When applying our developed single time-step space-time adaptive technique and adapting the mesh anisotropically every 10 time-increments, the generated time-step sizes will be too small, as seen in figure 6.14(left), in order to reduce the temporal error, preventing the progress of the solution in time. Therefore, in this case, the anisotropic mesh adaptation that provides a sharp capture of the evolving interface and aims at accelerating the computations seems to be blocking the circle's movement. Indeed, the single time-step mesh-time adaptive algorithm produces time-step sizes in such a way to guarantee that no phase lag occurs between the solution and the mesh. By constructing a mesh that is very steep in capturing the gradient of the levelset representing the moving circle, the anisotropic mesh adaptation creates a very thin layer of size  $2\mathcal{E}$  that is well refined around the boundary of the circle. The generated time-step sizes between two spatial remeshing shall be small enough to maintain the interface within the well refined zone. Consequently the execution time of the simulation will significantly increase.

We recall that the paradoxical method aims at dynamically and automatically generating optimal meshes for slabs of time hence yielding a better efficiency than the single time-step adaptation algorithm. This is exactly what we notice when comparing figures 6.14(left) and 6.14(right); the paradoxical meshing technique produces time-step sizes that are in average 20 times larger than those generated by the single time-step adaptation algorithm. We summarize in table 6.1 the CPU times needed to simulate 1s of rotation with a very fine fixed mesh (around 150000 nodes) and a fixed time-step  $\Delta t = 0.005$ , with the single time-step space-time adaptation and with the paradoxical meshing technique. The results reveal the efficiency of the proposed anisotropic adaptation method. The computational time is almost 10 times faster than the one produced with fixed mesh and time-steps, and 7 times faster than the one produced with the single time-step space-time adaptation. It is important to mention that the CPU time of the paradoxical adaptation is the sum of the time needed to generate the sets of meshes/time-step sizes and the time needed for the simulation with these sets. Note that the inner loop of the algorithm was repeated only two times to generate the optimal meshes and time-step size for which the solution remains bounded.

|                                                      | CPU time (hrs)           | Ratio VS fixed | Ratio VS single<br>time-step adaptation |
|------------------------------------------------------|--------------------------|----------------|-----------------------------------------|
| Fixed $\Delta t = 0.005$                             | 19                       | -              | -                                       |
| Single time-step adaptation                          | 14.06                    | 1.35           | -                                       |
| Slab of time adaptation<br>(Generation + simulation) | $1.081+0.78$<br>$=1.861$ | 10.21          | 7.55                                    |

Table 6.1 – History of the CPU times corresponding to the single time-step and slab of time meshing techniques applied to the rotating circle test case.

The developed technique results in a second order accurate solution. Using an isotropic refinement of the mesh, it was stated in [Guégan 10] that a second order ac-

curate capture of a physical phenomena with an error reduction factor of 4 is obtained as the number of nodes in the mesh passes from  $N_1$  to  $N_2 = 16N_1$ . However this condition is not enough for second order convergence, the size of the time sub-intervals should become 8 times smaller too. It was also highlighted in [Guégan 10] that in the anisotropic case, the mesh size should be 4 times smaller in regions with high gradients and only 2 times smaller elsewhere. Hence 2 times smaller sub-intervals of time are necessary for second order convergence when using anisotropic mesh adaptation. Figure 6.16 shows that the time-intervals generated by the paradoxical meshing approach satisfy the necessary condition for second order convergence in the anisotropic case.

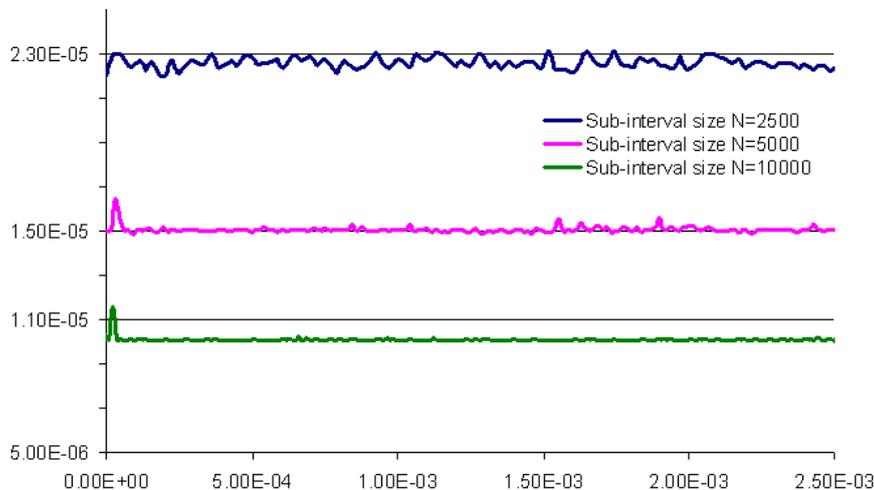


Figure 6.16 – Time sub-intervals satisfying the requirement for second order convergence in the anisotropic case.

We have repeated the same test case in three dimensional spaces and almost the same results were noted in terms of computational acceleration. Figure 6.17 depicts several snapshots on the predicted meshes and evolving sphere over time. Again, one can clearly identify how the mesh is well predicted to contain the moving sphere over slabs of time.

### 6.6.2 A two-dimensional analytical test case

The objective of this numerical example is to illustrate how the metric construction developed in this chapter enables the capture of the displacement of a function's anisotropic features over a slab of time. The analytical function that we consider inhere was presented in [Coupez 11], and was reconsidered in chapter 4 with an extension to account for time evolution. It is defined on the domain  $[0, 1]^d$  and is being evolved over the time interval  $[0, 1]$  as follows:

$$\begin{aligned}
a(\mathbf{x}) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X}\|\right)\right) \\
b(\mathbf{x}, t) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X} - t\mathbf{C}\|\right)\right) \\
\mathbf{C} &= \begin{pmatrix} 1 \\ \cdot \\ 1 \end{pmatrix} \\
u(\mathbf{x}, t) &= a \circ a(\mathbf{x}) + b \circ b(\mathbf{x}, t)
\end{aligned} \tag{6.20}$$

The parameter  $E$  varies from 1 to 32; the larger  $E$  is, the sharper the function's gradient, favoring anisotropic meshing. Despite the simplicity of this function, it is characterized by the displacement of the high gradients over time thus reflecting the ability of the algorithm to predict the behavior of the function and to prepare the adapted mesh for the coming slab of time. The complexity of this example, which is similar to a propagating wave, resides in the expansion and interference of the evolving circular gradients forcing isotropic meshing at the intersection regions.

We have repeated the same test case using the paradoxical meshing algorithm. The mesh is being adapted at a frequency of 10 time-steps. Figure 6.18 shows how the mesh is very well predicted to contain the function over a slab of time. The function is being presented at the beginning (left) and in the middle (right) of the time slab. Note that the paradoxical meshing algorithm is based on the same definition of the error estimators as in the single time-step adaptation algorithm so the same error analysis applies for this new adapting technique. We are presenting here the results obtained when adapting the mesh on the  $L_1$  norm of the error.

We compare in figure 6.19 and table 6.2 the time-step evolution and the CPU time obtained using the single time-step and the paradoxical meshing techniques. Note that the mesh is adapted on the  $L_1$  norm of the edge based error estimation every 5 time-steps so that the single time-step meshing do not lead to very small time-step sizes. We can clearly see that the single time-step method produces smaller time-steps and requires a higher CPU time. Indeed, unlike the paradoxical meshing that predicts a mesh for a slab of time where the solution can evolve smoothly in time, the single time-step meshing produces a mesh that is adapted to the very first time-step following the adaptation. The temporal adaptation algorithm corresponding to the latter approach will generate smaller time-steps, hence higher computational cost, in order to maintain the function's steep gradient within the refined regions. This result validates the efficiency of the newly developed method in terms of computational time.

### 6.6.3 A two-dimensional analytical test case with sinusoidal evolution in time

The objective of this test case is to illustrate the behavior of the time-step algorithm relative to sinusoidal evolution of the function in time. We consider the same analytical

|                             | CPU time (mins) | Ratio VS fixed $\Delta t$ |
|-----------------------------|-----------------|---------------------------|
| Fixed $\Delta t = 0.0015$   | 37              | -                         |
| Single time-step adaptation | 22              | 1.68                      |
| Paradoxical adaptation      | 17              | 2.17                      |

Table 6.2 – History of the CPU times corresponding to the single time-step and the paradoxical meshing techniques applied to (6.20).

function with a sharpness factor  $E = 16$  and we let it vary sinusoidally in time as follows:

$$\begin{aligned}
 a(\mathbf{x}) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X}\|\right)\right) \\
 b(\mathbf{x}, t) &= \tanh\left(E \sin\left(5\frac{\pi}{2}\|\mathbf{X} - t\mathbf{C}\|\right)\right) \\
 \mathbf{C} &= \begin{pmatrix} 1 \\ \cdot \\ 1 \end{pmatrix} \\
 u(\mathbf{x}, t) &= a \circ a(\mathbf{x}) + b \circ b\left(\mathbf{x}, \sin\left(t\frac{\pi}{2}\right)\right)
 \end{aligned} \tag{6.21}$$

The rings emerging inside the domain move periodically back and forth over time. Figure 6.20 shows the time-step history obtained with the single time-step and paradoxical meshing approaches. The mesh made of around 20,000 nodes is being adapted every 5 time increments using the single time-step adaptation approach and every 10 increments using the paradoxical meshing technique. We notice that both profiles have the same tendency of periodic evolution over time. A phase of slow steady time-steps reflects the expansion of the rings inside the domain. A deceleration of the function follows and is accompanied by a quick increase of the time-steps. An abrupt decrease of the latter is observed once the function changes its direction of evolution. This succession of time-step behavior is periodically repeated in accordance with the sinusoidal displacement of the function in time.

The higher time-steps produced by the paradoxical meshing algorithm are in accordance with expectations. The predicted mesh, being well refined to contain the function for a slab of time, enables a faster and smoother propagation of the function over the domain. This explains also the lower CPU time cost of the method illustrated in table 6.3.

|                             | CPU time (mins) | Ratio VS fixed $\Delta t$ |
|-----------------------------|-----------------|---------------------------|
| Fixed $\Delta t = 0.00015$  | 351             | -                         |
| Single time-step adaptation | 213             | 1.647                     |
| Paradoxical adaptation      | 132             | 2.65                      |

Table 6.3 – History of the CPU times corresponding to the single time-step and the paradoxical meshing techniques applied to (6.21).

#### 6.6.4 Fluid flow for different Reynolds numbers

We continue our validation of the paradoxical adaptive technique by considering the classical 2-D lid-driven flow problem with a zero source term. We intend to prove the applicability and efficiency of the newly developed paradoxical meshing technique to unsteady CFD problems. The viscosity is adjusted to yield Reynolds number of 5,000, 10,000 and 20,000. In order to generate and compare the solutions we fix the number of nodes  $N$  to 30,000. The user-prescribed number of nodes is an advantage of this method as it avoids the drastic increase in the number of degrees of freedom, and hence controls the computational time. We start the simulation with a time-slab size of 2s on an unstructured isotropic mesh and adapt every 20 increments. We adapt the mesh on the velocity norm as well as the normalized velocity vector field.

For illustration purposes, we consider the case of  $Re = 5,000$  and present the resulting mesh on the evolving solution over the time slabs  $[1.93, 2.75]$  and  $[6.8027, 7.11646]$  in figures 6.21 and 6.22, respectively. Figure 6.23 depicts the final mesh generated once the flow has been settled down inside the cavity. It shows the correct orientation and deformation of the mesh elements (longest edges parallel to the boundary). This yields a great reduction of the number of elements. These results give confidence that the extension of the approach to take into account all the velocity components ( $V_x$ ,  $V_y$ , and  $||V||$ ) holds very well and plays an important role for transient flows. Note the high resolution in the domain region where the solution will be traveling, on the boundary layers and at the detachment regions reflecting the multi-component structure of the metric field. The paradoxical meshing algorithm proves to be capable of capturing the anisotropy of the solution caused by the discontinuity of the boundary conditions and the nature of the flow.

The same simulation is conducted for  $Re = 100,000$ , the flow is characterized by turbulent features with the emergence of vortices of different sizes over the simulation. Since benchmark values on the velocity profile cannot be found in the literature, we limit this case to showing that the mesh is well adapted to capture the different features of the flow. We can detect from figure 6.24 how refined is the mesh in the vicinity of the velocity gradient over the slab of time  $[9.524, 10.073]$ . The emerging small eddies are very accurately captured and followed over time. As the mesh is well predicted to contain the velocity field, the computations went very smoothly with a rapid convergence of the Navier-Stokes VMS solver. This is highlighted by the reduced CPU time (33192s) as compared with the one needed when using the single time-step space-time adaptive meshing algorithm (41347s). We move on next to compare the accuracy of the newly developed paradoxical meshing technique, to the single time-step adaptation approach and to very accurate reference solutions. For that purpose, we first plot the velocity profiles for  $U_x$  along  $x = 0.5$  and for  $U_y$  along  $y = 0.5$  at Reynolds 1,000, 5,000, and Reynolds 10,000. It can be seen that all the results are in very good agreement with the reference solution computed on a  $600 \times 600 = 360,000$  points fixed mesh [Hachem 10b]. Almost the same accuracy of the solution is obtained with both

anisotropic meshing techniques for Reynolds 1,000 and 5,000. A better performance of the paradoxical algorithm can be spotted for Reynolds 10,000 resulting from the high concentration of refined meshes over the regions of sharp gradients allowing the good capture of the solution all over the simulation time and avoiding the accumulation of numerical errors due to the mesh-solution time lag.

Regarding the efficiency of the proposed algorithm, we summarize in table 6.4, the CPU time needed to simulate 1,000s of the flow using a fixed mesh and a fixed time-step size, an anisotropically adapted mesh and time-steps generated by the single time-step adaptation algorithm and an anisotropically adapted mesh and time-steps generated by the paradoxical meshing algorithm. It is important to mention that the reported CPU times using paradoxical meshing technique account for both the generation and the simulation phases. It can be inferred from the gathered statistics that the single time-step adaptation is faster than the paradoxical meshing for  $Re \leq 5,000$ , that is for laminar flows. As the Reynolds number increases, the paradoxical meshing technique becomes more efficient. This is in accordance with the time-step sizes generated by both methods and reported in figure 6.26. The observed slower computations can be related to the predictor corrector nature of the algorithm therefore the simulation is repeated several times whereas in the single time-step adaptation technique a single run is performed. A fast convergence of the Navier-Stokes VMS solver is noted in both cases. In addition, being adapted all over the simulation, the mesh provides a good track of the small scale vortices that develop with time. Consequently, the time-step size decreases to allow the resolution of these new physics. When the flow becomes turbulent and develops eddies of different sizes, the solver requires more time to converge the solution on the available mesh and with the available time-steps generated by the single time-step adaptation algorithm, hence the increase in the CPU time. We recall that the mesh is adapted every 5 time increments while it is not actually valid for this slab of time. On the other hand, using the paradoxical meshing technique proposed in this chapter, the mesh holds for slabs of time with lengths equal to 20 time increments. A faster convergence of the solvers is detected yielding a reduction of the total CPU time.

| Reynolds Number  | 1,000 | 5,000 | 10,000 | 20,000 | 100,000 |
|------------------|-------|-------|--------|--------|---------|
| Single time-step | 372   | 618   | 1487   | 2622   | 41347   |
| Slab of time     | 426   | 791   | 1530   | 1970   | 33192   |

Table 6.4 – CPU time needed for computing the solution with the single time-step space-time adaptive and the paradoxical methods.

### 6.6.5 Flow past a wind turbine

In this application we intend to assess the validity and potential of the paradoxical meshing tool together with the IVM and the stabilized solver in simulation complex three dimensional flow behind a moving structure.

Over the last decade, considerable attention has been devoted to the numerical studies of renewable energy devices such as wind turbines in the view of optimizing the power production, energy conservation and the shapes of the blades. The main focus shall be on the wake velocity deficits that reduce the power production and on the aerodynamics loads on the blades. The numerical investigation that we conduct inhere does not intend to perform an aerodynamic analysis or optimize the wind farm layout. Instead, we aim at providing a useful tool that facilitates and improves the accuracy of such an analysis. The objective is to test the applicability and potential of the paradoxical meshing algorithm together with the Immersed Volume Method and the stabilized Variational multiscale approach in simulating such a challenging problem. Indeed the full scale modelling of wind turbine applications requires several numerical tools. The first is the generation of a well refined mesh that is capable of following the blades' movement. The second involves the precise capture of the boundary layers on the blades and the third accounts for the compressibility effects at the blade tips.

Inhere we are interested in addressing the first component of such an application. The computational domain is set to  $20m$  in the stream direction and  $10m$  in the lateral direction. We consider a 3 twisting blades wind turbine positioned at  $3m$  from the tunnel's inlet. The wind inflow velocity  $\mathbf{v} = 5.75 \left(\frac{y}{195}\right)^{0.35}$  such that  $\|\mathbf{v}_{\max} \approx 2m/s\|$  is chosen to yield a low Mach number and a Reynolds number of around  $35,000$  whereas the tip speed ratio is set to  $\lambda = \frac{R\omega}{v} = 7$  with  $R$  being the rotor radius and  $\omega$  the angular velocity of the blades in the  $z$ -axis direction. This setup of the problem implies a low Mach number and consequently can be performed using the available incompressible solver.

As discussed in example 5.4.4, the use of the single time-step meshing and time adaptive algorithms slows down the computations as the mesh should be maintained very dense and sharp around the interfaces of the blades and rotor. This will induce significantly small time-step sizes ( $\Delta t = 0.0005$ ) and consequently imply a considerable execution time. On the contrary, the paradoxical meshing algorithm predicts the solution's evolution over the frame of time and constructs the optimal mesh that is adequately designed to minimize the  $L^p$  norm of the interpolation error over the slab of time. This example reflects the multi-components feature of the devised anisotropic adaptation algorithm by adapting the mesh on the velocity norm, the velocity field directions, the filtered levelset function of the blades and the filtered levelset of the rotor. The number of nodes is fixed to  $N = 500,000$  and the initial time-step size to  $10^{-4}$ . The simulation was run on 64 cores from a GNU/LINUX cluster of 2.4 Ghz Opteron cores connected by an InfiniBand network.

Figures 6.27,6.28 and 6.29 depict respectively the Q-criterion value, the pressure field and velocity streamlines and the velocity magnitude on the blades with the corresponding anisotropic meshes generated over the simulation time. We recall that the Q-criterion represents the second invariant of the velocity field's gradient. When this value is positive, the rotation dominates the strain and vice versa. The contours of

this field highlight the vortices developed over time by the trailing interfaces and tips of the blades. This is also reflected in figure 6.28 by the swirling and intensity of the velocity streamlines at the level of the rotating blades. The negative pressure profile on the zero iso-value of the levelset function pinpoints the zones of intensive lift forces exerted on the moving blades. One can clearly detect, in figure 6.29, how well the mesh is refined to capture the gradient of the velocity field, the emerging eddies and the rotating geometry. We can identify how the anisotropic mesh adaptation renders a good capture of the flow's details such as the fully developed vortical structure around the turbine and the boundary layers at the blades' level. The elements are highly concentrated and well oriented in the wake region and at the blades' boundary level where a strong boundary layer is being developed. This reflects how, for a fixed number of nodes, the algorithm automatically and dynamically optimizes the nodes' distribution and orientation. The evolution of the velocity streamlines enables us to understand the flow characteristics over time. With time, the flow starts exhibiting a turbulent behavior, fluctuations become more important and small and large eddies appear at different positions behind the geometry. In parallel, one can identify how the dynamic mesh adaptation automatically refines the critical zones of sharp velocity gradients behind the wind turbine where the wake is developing. The mesh is also highly refined in a large layer around the blades and the cylindrical support, following the development of a strong boundary layer on their surface.

Using these tools, a full scale wind turbine with the exact dimensions and airflow conditions will be the subject of an upcoming thesis work [Billon 16] where the boundary layers and compressibility effects at the blades' tips will be accounted for.

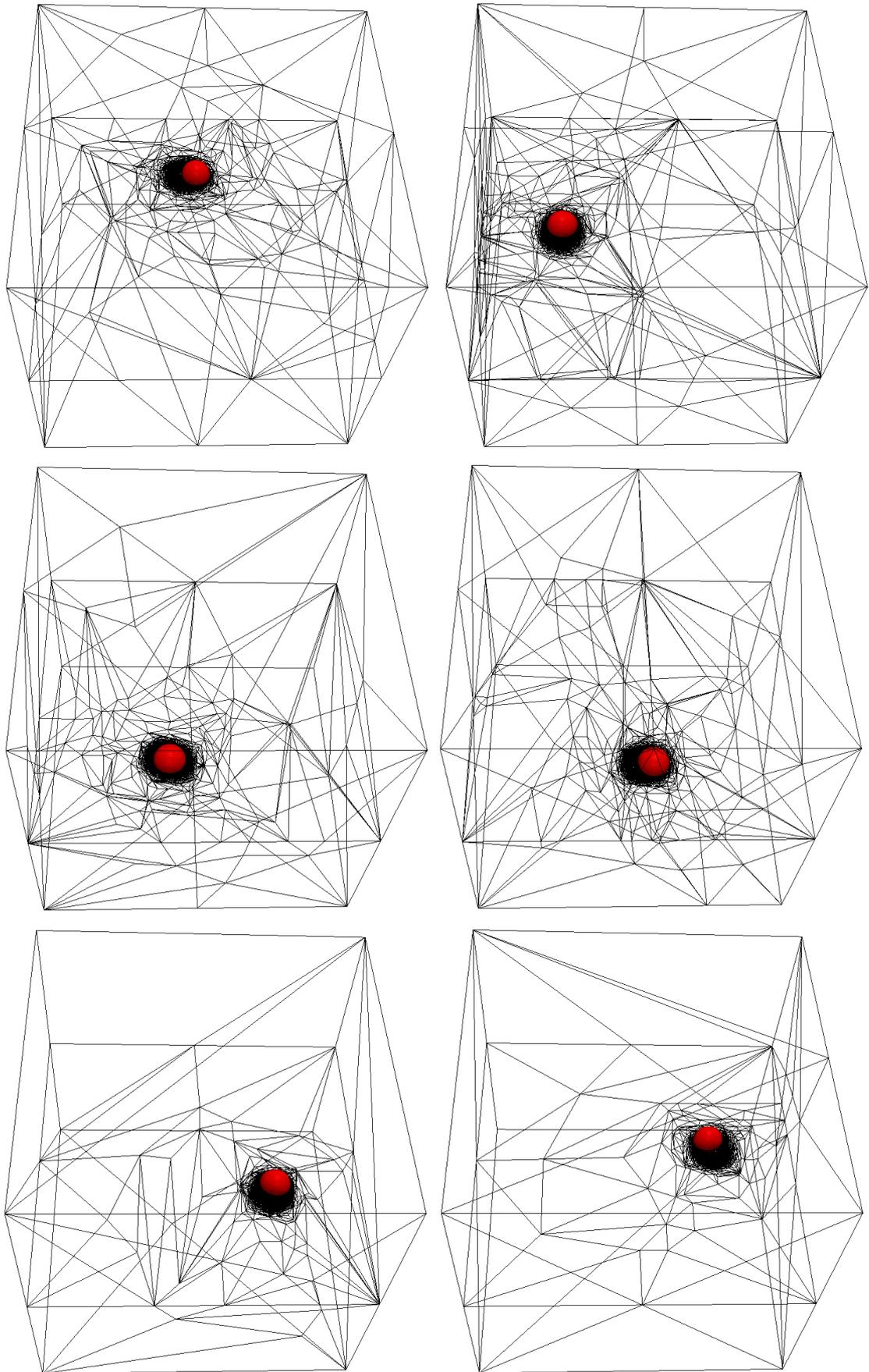


Figure 6.17 – Different snapshots reflecting the rotating sphere at the beginning, the middle and the end of the predicted slabs of time.

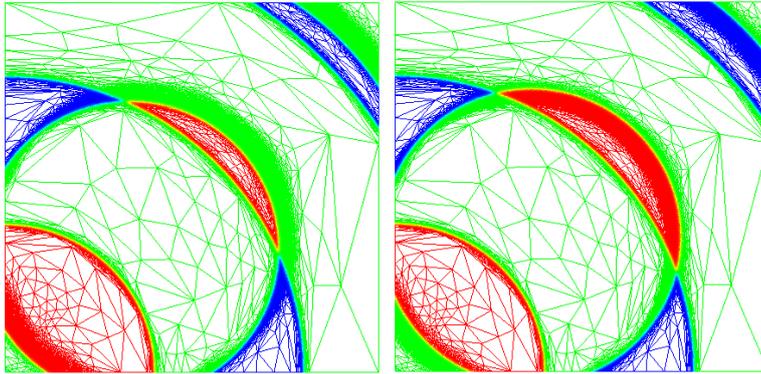


Figure 6.18 – Anisotropic mesh adapted over a slab of time to contain the analytical function defined by (6.20).

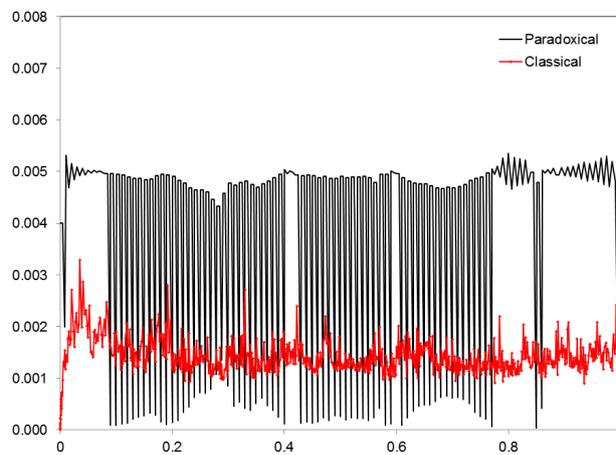


Figure 6.19 – History of the time-steps corresponding to the single time-step and the paradoxical meshing techniques applied to (6.20).

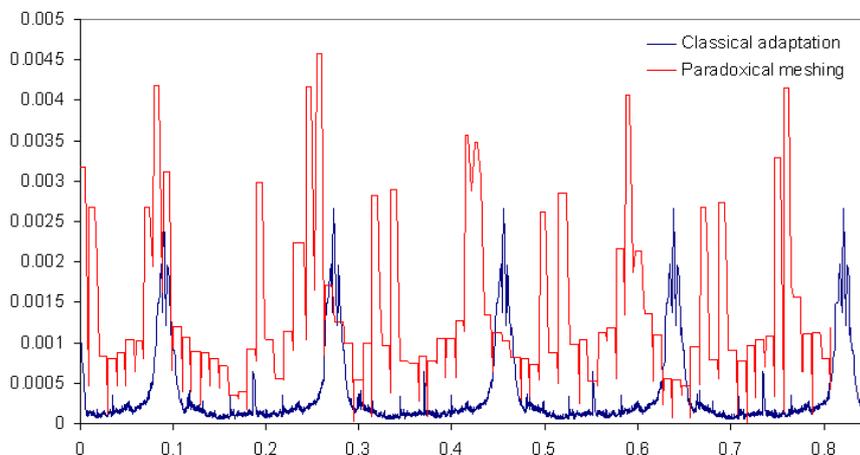


Figure 6.20 – History of the time-steps corresponding to the single time-step and the paradoxical meshing techniques applied to (6.21).

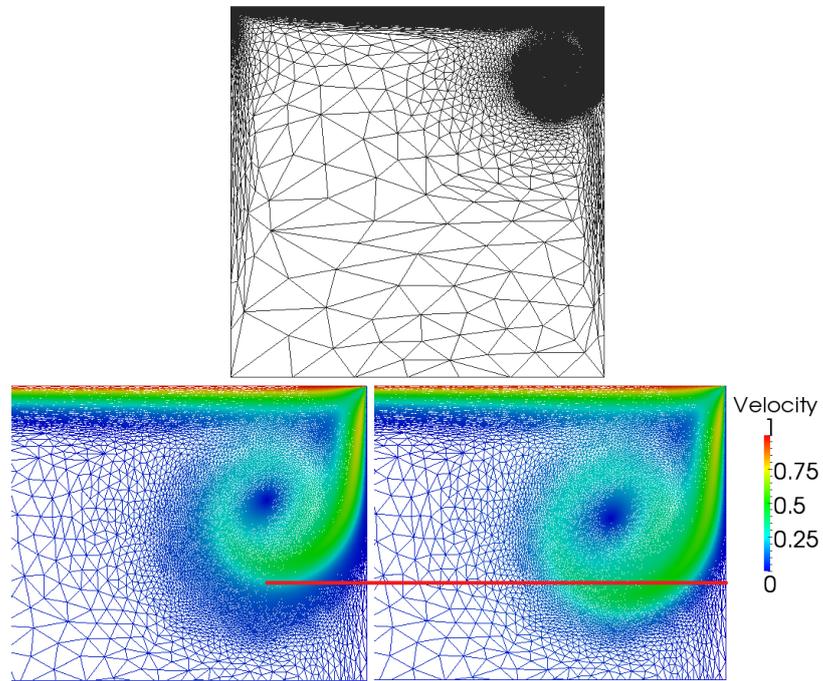


Figure 6.21 – Plot of the mesh and a zoom on the main velocity vortex over the time slab  $[1.93, 2.75]$  corresponding to 20 time-increments.

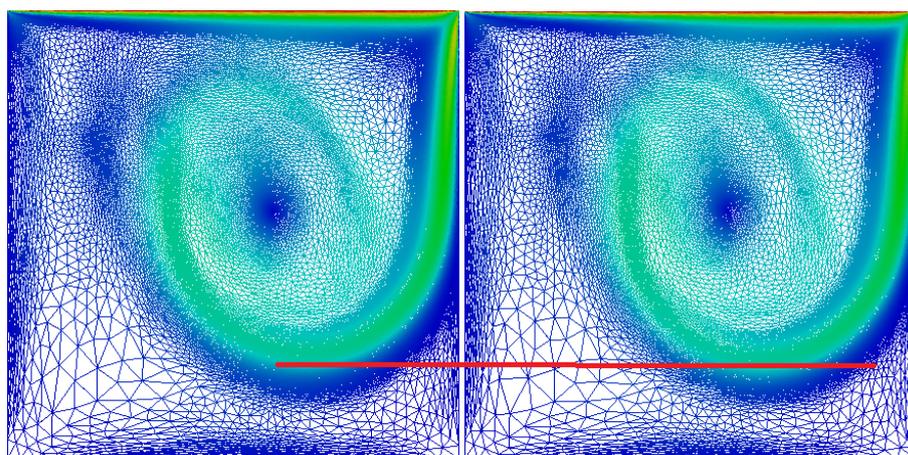


Figure 6.22 – Plot of the mesh and the main velocity vortex over the time slab  $[6.8027, 7.11646]$  corresponding to 20 time-increments.

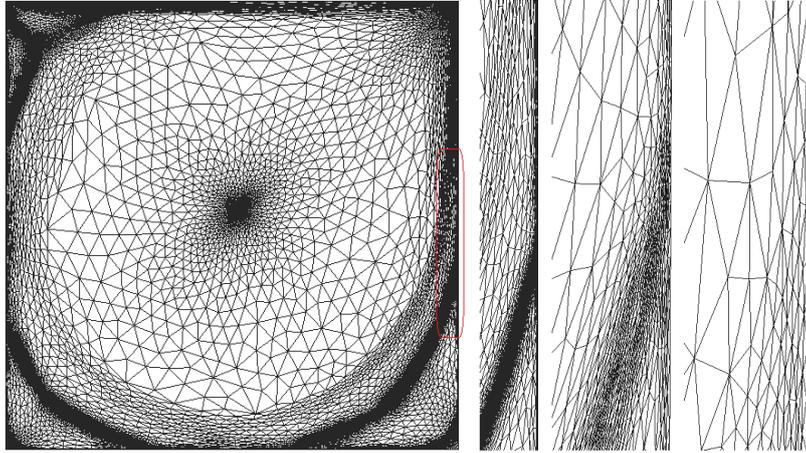


Figure 6.23 – Final mesh corresponding to the stable flow inside the cavity (left). Zooms on the mesh close to the right wall (right).

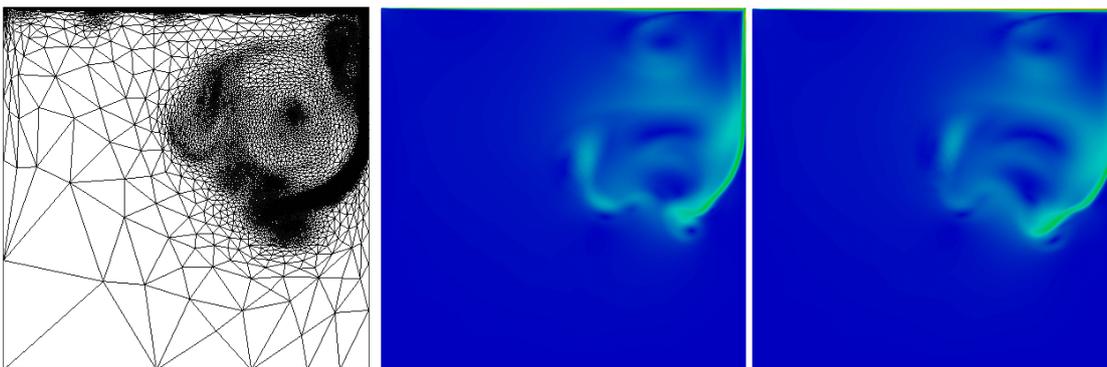


Figure 6.24 – Velocity magnitude over a slab of time with the optimal anisotropic mesh obtained at  $Re=100,000$ .

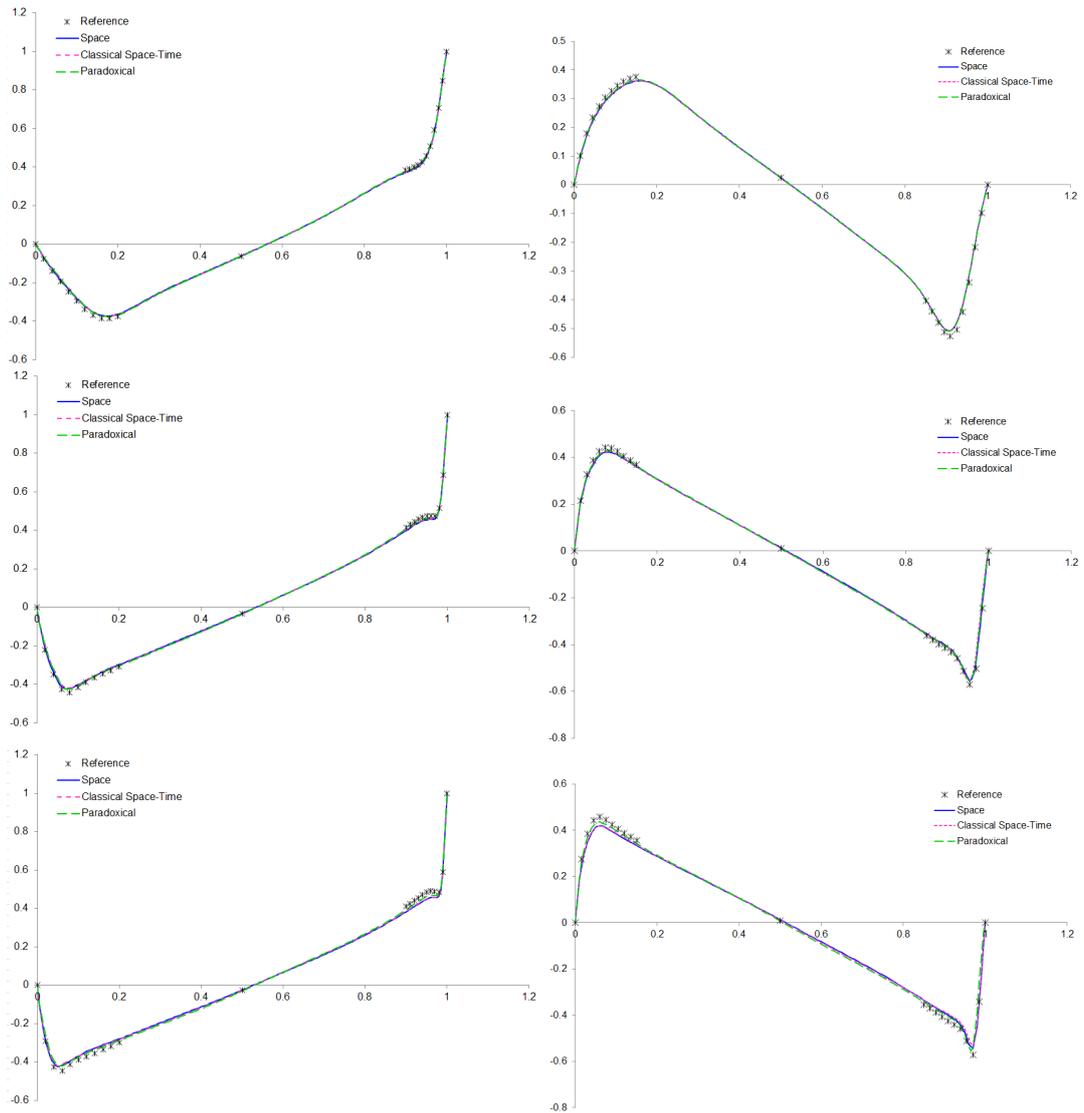


Figure 6.25 – Comparison of velocity profiles in the mid-planes for  $Re = 1,000$  (top),  $Re = 5,000$  (middle) and for  $Re = 10,000$  (bottom). Left: Velocity profiles for  $U_x$  along  $x = 0.5$ . Right: Velocity profiles for  $U_y$  along  $y = 0.5$ .

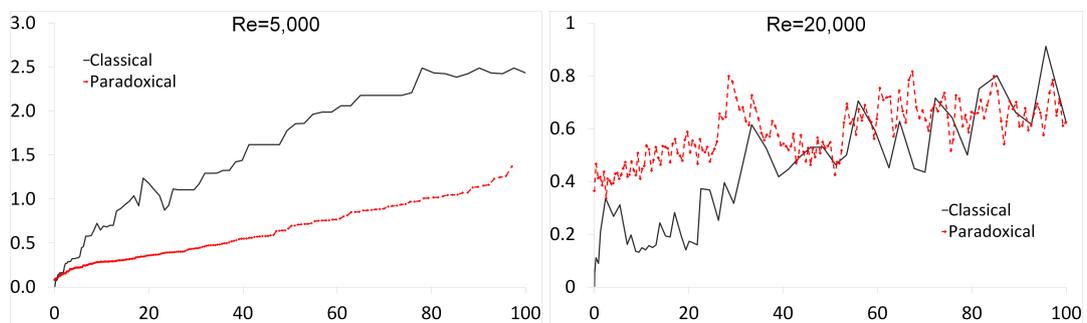


Figure 6.26 – Variation of the time-slab sizes generated by the single time-step (solid line) and the paradoxical (dashed line) meshing techniques for the driven cavity problem at Reynolds numbers 5,000 and 20,000.

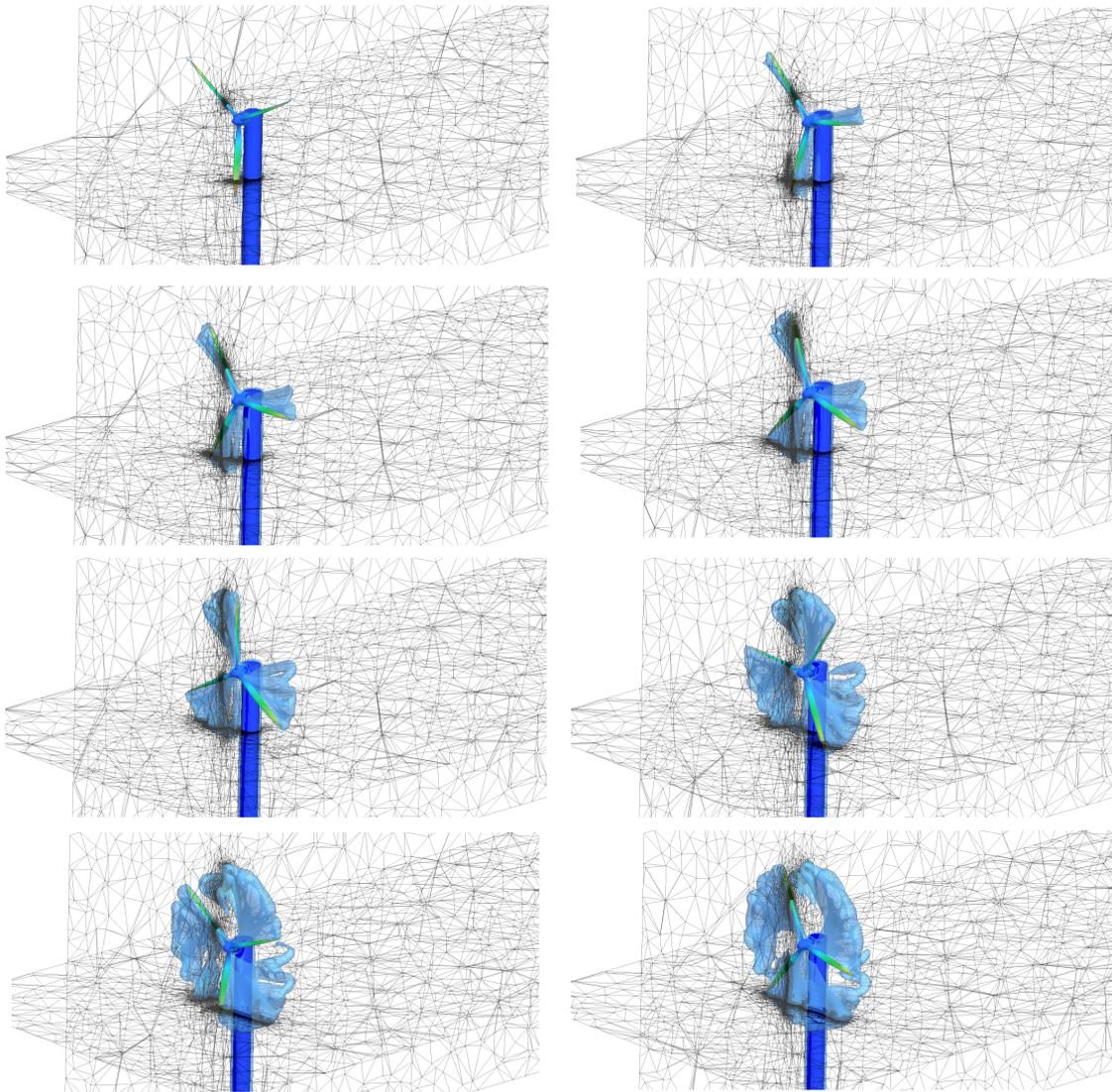


Figure 6.27 –  $Q$ -criterion contours for the value 0.5 at different time instances.

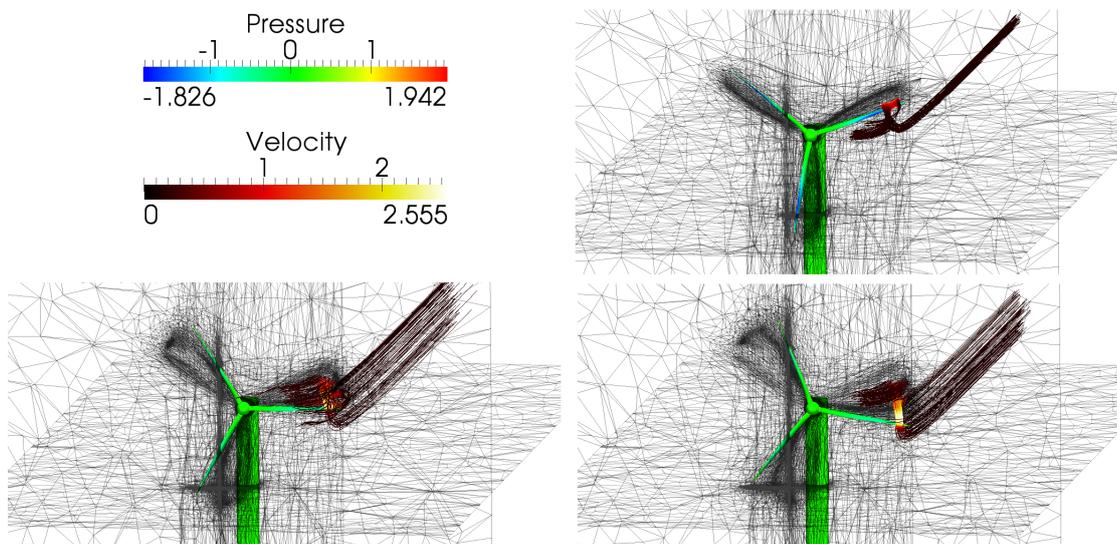


Figure 6.28 – Pressure field and velocity streamlines at three successive times.

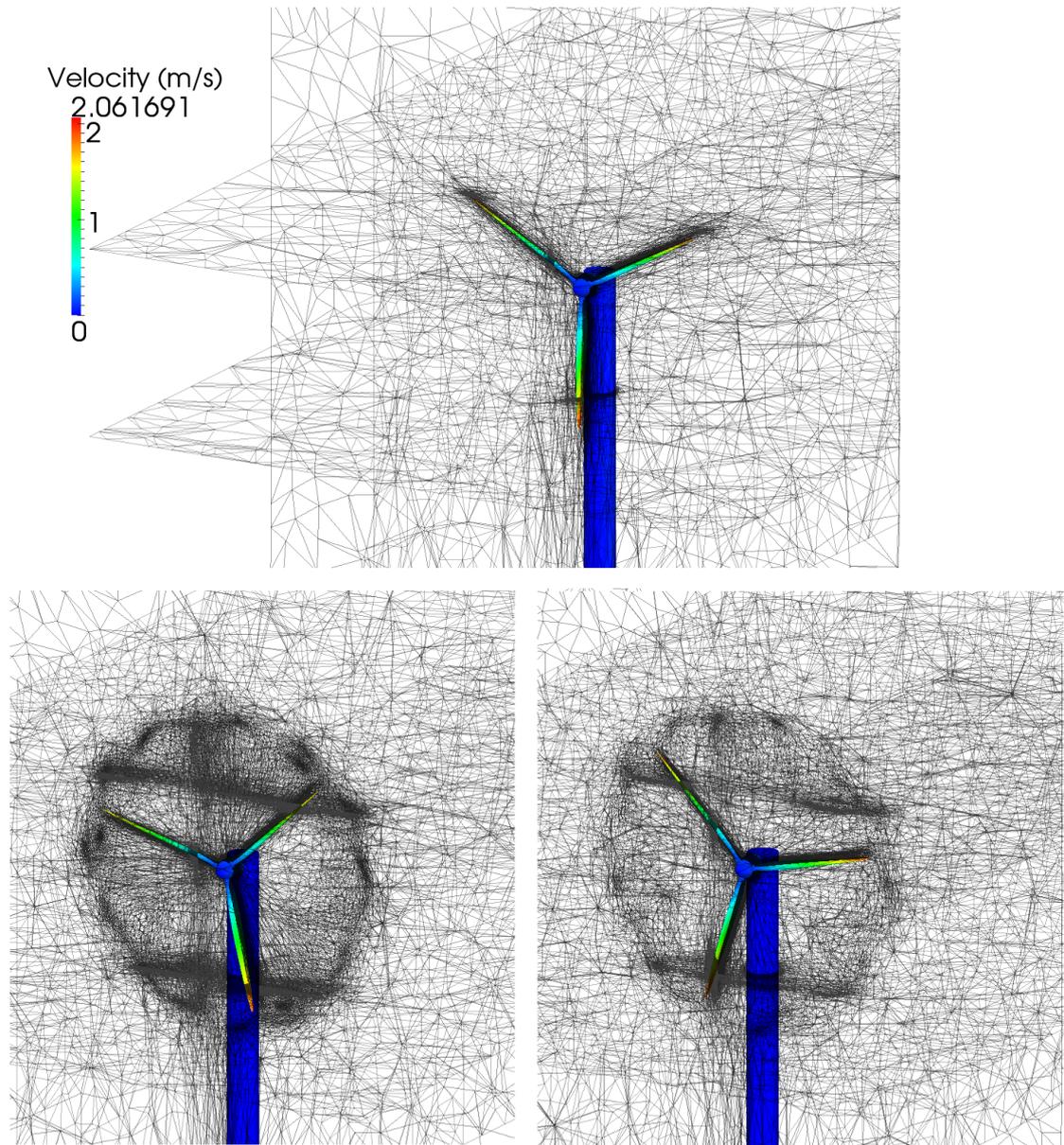


Figure 6.29 – Velocity magnitude on the blades and the generated anisotropically adapted meshes.

### 6.6.6 Conjugate heat transfer inside an industrial furnace

We are interested in this numerical example in validating the developed full adaptation approach on a 3D simulation of conjugate heat transfer and fluid flows inside an industrial furnace. The performance of the stabilized finite element solvers has been investigated on this physical problem in chapter 1. We intend inhere to reconsider the same problem setup and apply the adaptation algorithms in the view of improve the accuracy and efficiency of computations. For that purpose, we fix the number of nodes to 50,000 and adapt the mesh for 10 time-increments based on the levelset functions associated with the ingots, the outlets and the burner, on the temperature field, and the velocity's norm and direction. We present in figure 6.30 the initial mesh generated by the adaptation algorithm. We highlight the sharp capture of the ingots allowing a precise material distribution on each side of their interfaces. We recall that hot gas is injected inside the furnace at a speed of 38m/s and a temperature of 1350°C. The enclosure is initially taken at rest and having a temperature of 700°C and the Steel 40CDVL3 workpieces are immersed with a temperature of 400°C.

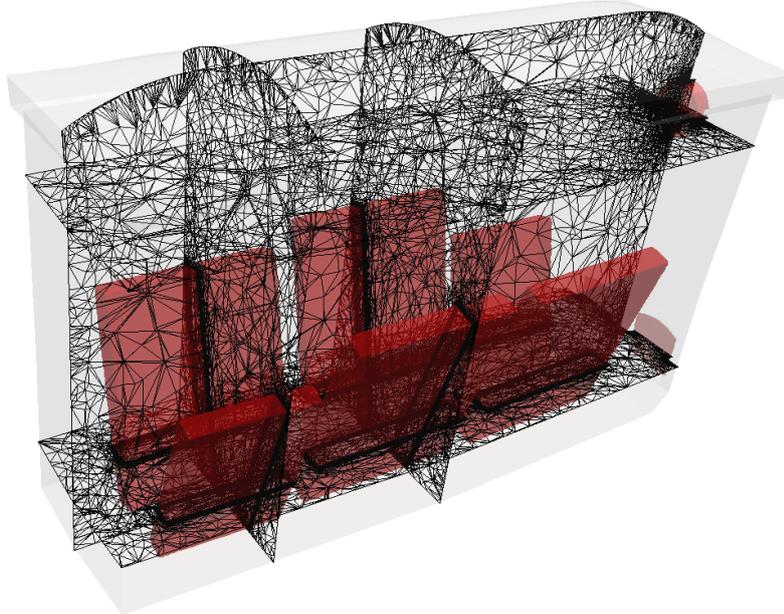


Figure 6.30 – Initial configuration and mesh for the heat treatment inside an industrial furnace.

The thermal distribution inside the 3D enclosure along 5 different horizontal and vertical plane cuts is shown in figures 6.31 and 6.32 at different time instances throughout the simulation. Once the burner is turned on, the hot gas spreads inside the domain and gets in contact with the air initially at rest creating sharp temperature and velocity gradients. The temperature profile follows the expected spread over time. While it took only few seconds to achieve the desired temperature inside the furnace, a considerable time was needed to raise the solids' temperatures by few degrees. This is clearly in accordance with the material properties of the ingots. We identify the gradual change in the temperature of the ingots with time. As the surface gets in contact with the hot fluid, it gets heated faster than the heart of the workpieces then the heat

is transferred by conduction to the interior. We can also spot that the workpieces that are the farthest from the burner are heated first. This is in accordance with the features of the flow. Indeed, the injected flame does not get in direct contact with the solids, on the contrary, it hits the opposite wall generating a counter-rotating flow and the main vortex dissociates into two smaller vortices: the first one climbs the wall and the second spreads into the lower part of the cavity ensuring an equidistributed heating of the furnace. The latter vortex transports the hot gas and induces the heating of the cold ingots. This interaction between the hot fluid and the cold solid results in a heat exchange between them, cooling the fluid and heating the solid. Thus the moving flow will lose few degrees of its temperature and hits the following workpiece at a lower value. This explains the slower raise in the temperature of the ingots that are closer to the burner.

The velocity field and streamlines distribution within the duct are provided in figures 6.33 and 6.34. They reveal the complex structure of the flow pattern. The latter is characterized by a fluctuating and chaotic behavior with the appearance of small and large scale vortices. Indeed, as the hot fluid is pumped, it induces a turbulent and swirling motion inside the volume. In order to account for the turbulent pattern, the  $k - \varepsilon$  model is employed. We notice that primary vortex is decelerated by the workpieces and is dissociated into smaller secondary vortices in a counter-rotating behavior near the fluid/solid interfaces.

Figure 6.35 presents the anisotropic mesh generated for the slab of time [26.05, 29.65]. We can clearly see how sharp is the capture of the workpieces and how well the mesh is being predicted to contain the flame jet over this interval.

A quantitative comparison between the temperatures evolution profiles obtained with and without space-time adaptations at different positions inside the furnace are reported in figure 6.36. We detect a very good agreement between the thermal distribution on a fixed mesh made up of 157,347 nodes and on the anisotropically adapted ones having around 50,000 gridpoints in the furnace volume whereas a difference can be spotted at the level of ingots. By optimizing the distribution of the nodes inside the computational domain a more precise representation of the interfaces and a better material distribution are provided by the newly developed anisotropic mesh adaptation tools. This higher accuracy results in a less diffusive thermal expansion inside the ingots. Consequently, a faster heating is numerically reflected as compared with the fixed mesh and time-stepping. Furthermore, while the former simulation required 5 days of computations on 32 cores, the latter took only 17 hours of execution time. This computational improvement reveals the great utility of the developed adaptive approaches.

The generated time-slab sizes over the simulation time are summarized in figure 6.37. We notice that during the first 2000s of heating, small time-steps are created allowing an accurate capture of the thermal evolution and flow pattern inside the volume. Once the furnace reaches the desired temperature and the flow settles down, the main heat transfer process is dominated by the conduction at the level of the workpieces,

the generated time-slabs increase in size resulting in faster computations. We recall that the  $k - \varepsilon$  turbulence model has been employed to generate a mean flow profile, a more accurate solution can be obtained by deactivating this model however a higher mesh density is acquired to achieve a good level of accuracy, hence inducing smaller time-step sizes and thus a higher CPU time.

Therefore, the developed paradoxical meshing method permits, through the dynamic and automatic update of the anisotropic mesh and the time-slab sizes, the improvement of the accuracy and efficiency of the computations.

## 6.7 CONCLUSION

In this chapter, we have developed a new approach for space and time adaptation in the context of unsteady CFD problems, and more importantly, problems involving moving geometries. In the latter case, the adaptation benefits from the advantages of the Immersed Volume Method whereby, a unique mesh is employed for the different sub-domains and the evolving geometries' interfaces are tracked using a levelset function. The gradient of this function will guide the spatial adaptation to follow the movement of the objects over time.

The developed anisotropic mesh adaptation and time adaptive algorithm relies on a predictive corrective approach whereby the solution's evolution over a slab of time is predicted and the corresponding spatial mesh and time-step sizes are automatically generated allowing a good level of accuracy. Furthermore, fewer function interpolation are performed lowering the accumulated interpolation errors. The new metric constructed for several time-steps ensures that the error on each edge of the spatial domain remains bounded by the global error over the slab of time. Moreover, the efficiency of computations is improved with respect to the classical approach for adaptation as the novel technique avoids the computation of a new mesh at every solver iteration.

The paradoxical meshing algorithm enabled the generation of a set of optimal time-step sizes and anisotropic meshes that hold for a slab of time. The method was formulated as an iterative process of prediction/remeshing/correction whereby the aim was to control the  $L_\infty(t; L_p(\Omega))$  norm of the spatial interpolation errors and the  $L_\infty$  norm of the temporal error over the different slabs of time. An extension of the edge-based error estimation developed in chapter 2 was provided. It allowed the estimation of the maximum  $L_p(\Omega)$  norm of the interpolation errors along the edges of the mesh over the slab of time. These collected error estimates were then used to estimate a homogenized equidistributed error and yield the construction of metric tensors. Each of these tensors were defined in a way to prescribe on a node of the initial mesh a set of stretching factors along and in the direction of the edges connected to it.

Moreover, an extension to take into account several normalized fields in a single metric was proposed. The constructed spatial metrics were then provided to the MTC mesher that in turn generated appropriate optimal meshes. A coupling between the space and time adaptations was enhanced through the use of a homogenized equidis-

tributed error to control the temporal errors induced on the slabs of time. Consequently, stretching factors were determined on the different slabs of time predicting for each temporal edge its optimal size.

A new 1D temporal mesher was developed and allowed the redistribution of the temporal nodes according to the estimated optimal time-slab sizes.

The paradoxical algorithm has been validated on analytical examples, CFD problems and problems involving moving geometries in two and three dimensional spaces. In all the cases, the method demonstrated a good performance and a high level of precision. In terms of CPU cost, an important reduction was highlighted due to the adaptive time-stepping and the smoother convergence of the solvers. This faster convergence is in accordance with the consistency between the predicted meshes and the evolving solution.

We note that the novel adaptation technique can be applied in conjunction with the classical adaptation approach. Indeed, the computations can be performed using the classical approach with a fixed frequency of adaptation. When the induced errors along the spatial edges of the mesh become large on an interval of time between two-remeshings, the paradoxical technique can be activated to construct the appropriate optimal meshes and time-slab sizes.

We intend in the following chapter, to validate the developed space and time adapted tool on complex industrial applications. The numerical results will be confronted to experimental ones and with the ones obtained without adaptation reflecting the potential and efficiency of the novel approach.

## 6.8 RÉSUMÉ FRANÇAIS

Dans ce chapitre, nous avons développé une nouvelle approche dite paradoxale (ou adaptation espace-temps pour des slabs de temps) dans le contexte de simulation de problèmes CFD instables, ainsi que des problèmes impliquant des géométries mobiles. La nouvelle méthode repose sur une approche prédictive correctrice.

Une prévision de la solution sur un slab de temps est réalisée, le maillage anisotrope optimal et les pas de temps correspondants à ce maillage sont automatiquement générés permettant un bon niveau de précision vis-à-vis de la solution prédite. Une nouvelle métrique est construite pour plusieurs pas de temps garantissant que l'erreur induite sur chaque arrête du maillage spatial reste limitée par l'erreur globale associée au slab de temps.

La méthode a été formulée comme un processus itératif de prédiction / remaillage / correction permettant de contrôler la norme  $L_\infty(t; L_p(\Omega))$  des erreurs d'interpolation spatiale et la norme  $L_\infty$  de l'erreur temporelle sur les différents slabs de temps. Une extension de l'estimateur d'erreur développé dans le chapitre 2 a été développée. Ainsi, la méthode réalise un contrôle du maximum des normes  $L_p(\Omega)$  des erreurs d'interpolation induites le long des arrêtes du maillage au cours du slab de temps.

Ces estimations d'erreur ont ensuite été utilisées pour estimer une erreur équadistribuée homogénéisée sur tout le maillage au cours du slab de temps. Ensuite, une métrique était construite en chaque nœud du maillage initial imposant une taille de maille le long des différentes arêtes connectées à ce nœud. De plus, une extension vers une construction multi-composante a été proposée. Les slabs de temps optimaux ont été déduits en appliquant le principe d'équi-distribution de l'erreur en espace et en temps. Cependant pour répartir les pas de temps d'une manière optimale sur l'intervalle de temps de la simulation, un maillage 1D temporel a été développé.

L'algorithme de la méthode paradoxale a été validé sur des exemples analytiques, des problèmes CFD et les problèmes impliquant des géométries mobiles en deux et trois dimensions. Dans tous ces cas, la méthode a démontré une bonne performance et un niveau élevé de précision. En termes de coût de calcul, une réduction importante a été soulignée en raison de l'adaptation dynamique des pas de temps et de la convergence rapide des solveurs. Cette convergence plus rapide est une conséquence directe de la bonne compatibilité entre le maillage prédit et l'évolution des gradients de la solution.

Nous notons que la technique d'adaptation paradoxale peut être appliquée en combinaison avec l'approche d'adaptation classique (adaptation pour un seul pas de temps). En effet, les calculs peuvent être effectués en utilisant la méthode classique avec une fréquence fixe d'adaptation. Lorsque les erreurs induites le long des arêtes spatiales du maillage deviennent grandes sur un intervalle de temps entre deux remaillages, la technique paradoxale peut être activée pour prédire des maillages et des pas de temps optimaux le long de slabs de temps.

Nous avons l'intention dans le chapitre suivant, de valider la méthode d'adaptation espace et temps développée sur des applications industrielles complexes. Les résultats numériques seront confrontés à des données expérimentales reflétant le potentiel et l'efficacité de la nouvelle approche.

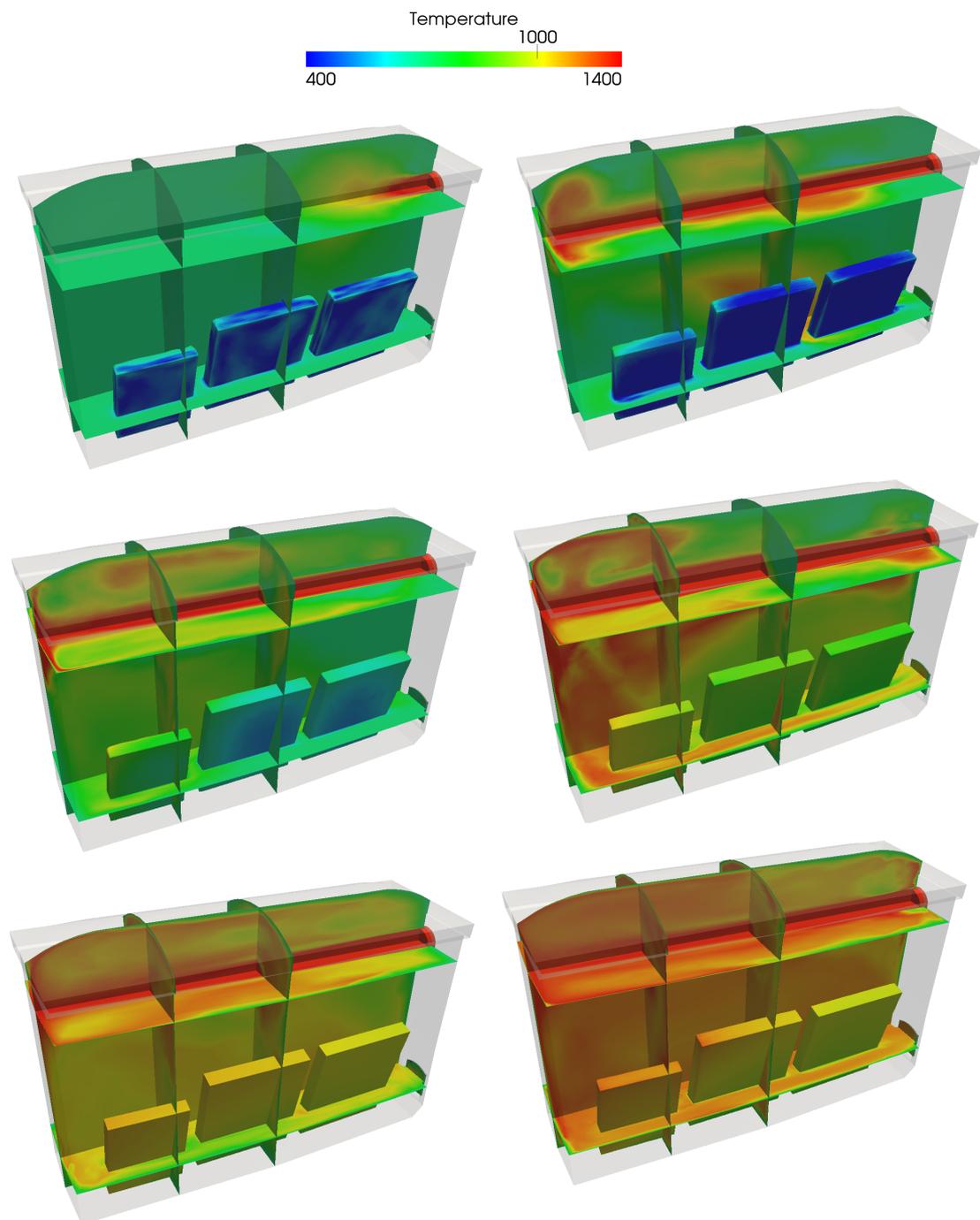


Figure 6.31 – Thermal distribution inside the volume and on the surface of the immersed solids.

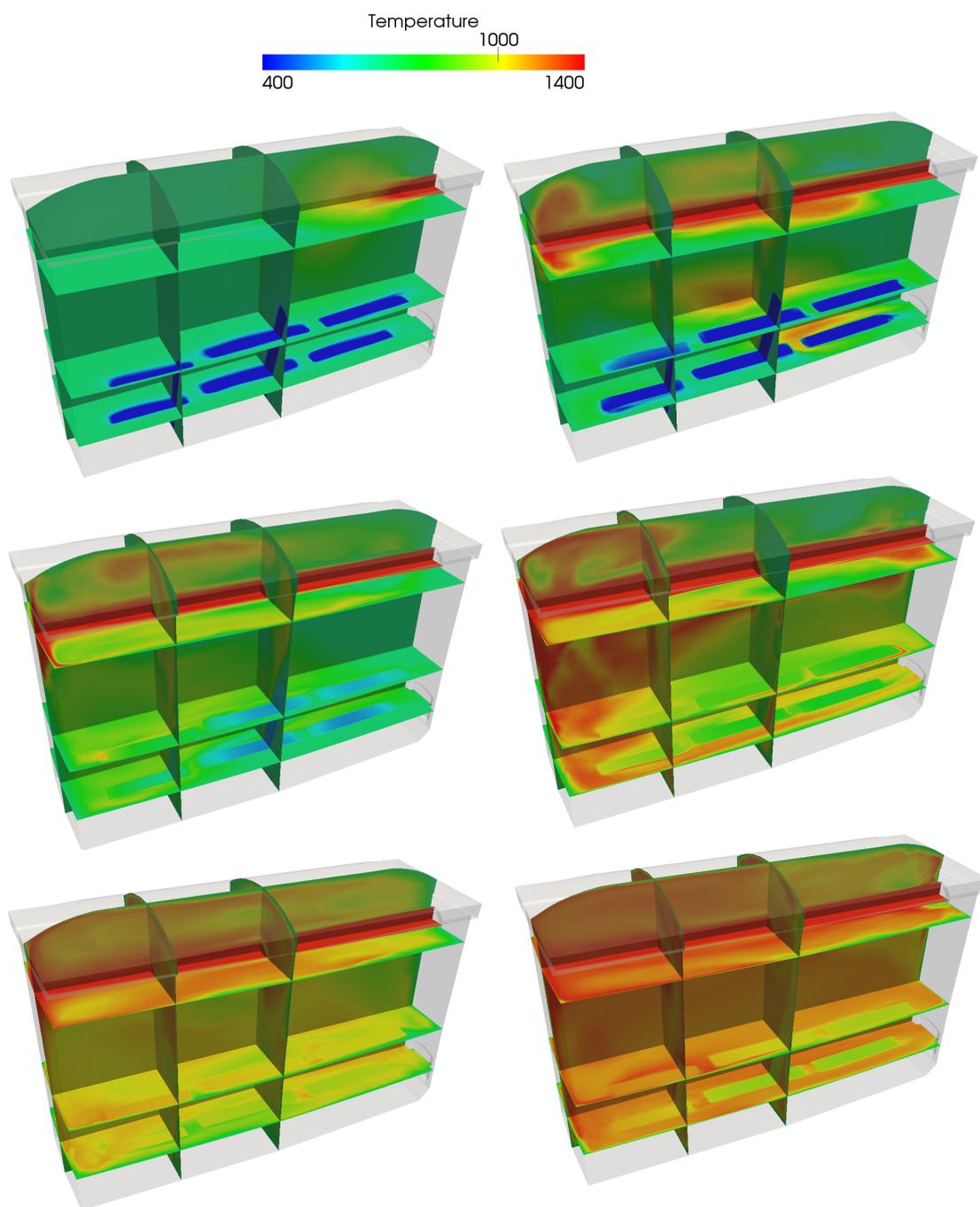


Figure 6.32 – Thermal distribution inside the volume and in the core of the immersed solids.

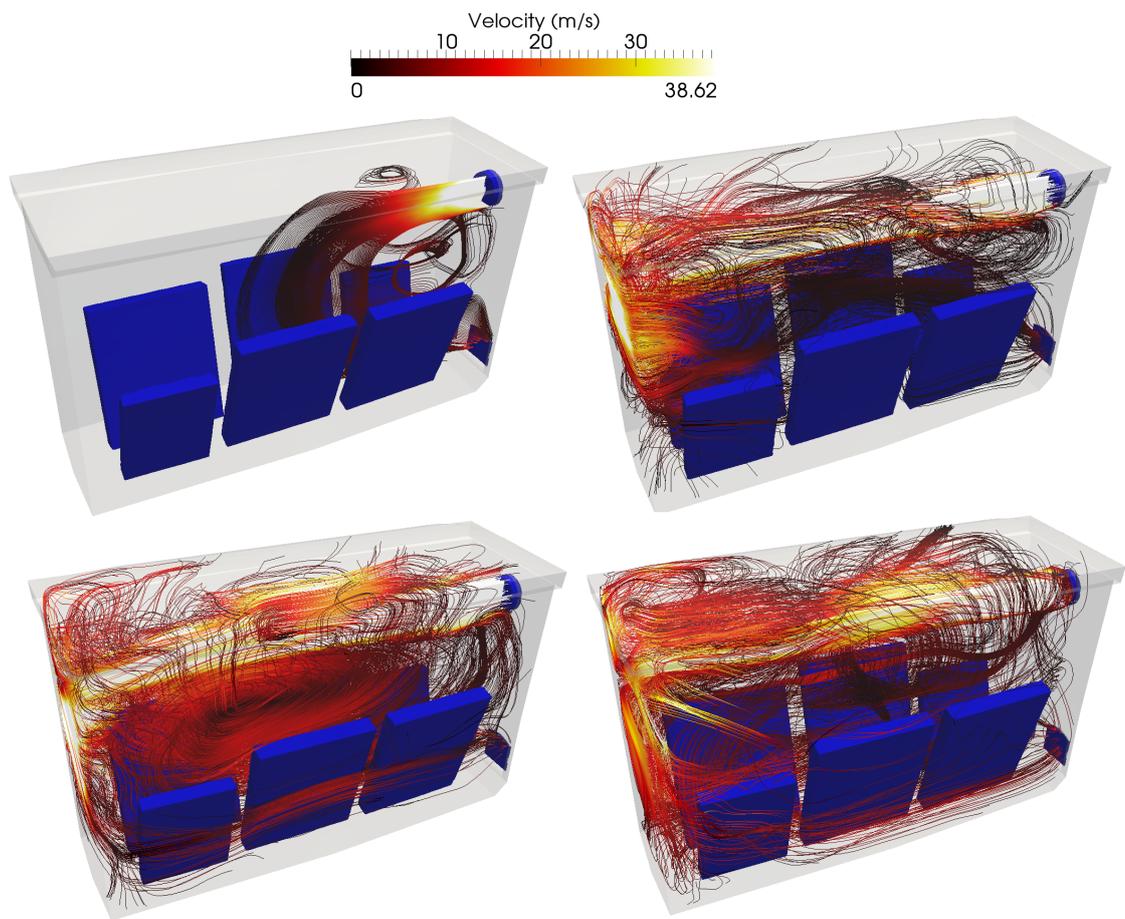


Figure 6.33 – Velocity streamlines at different time instances.

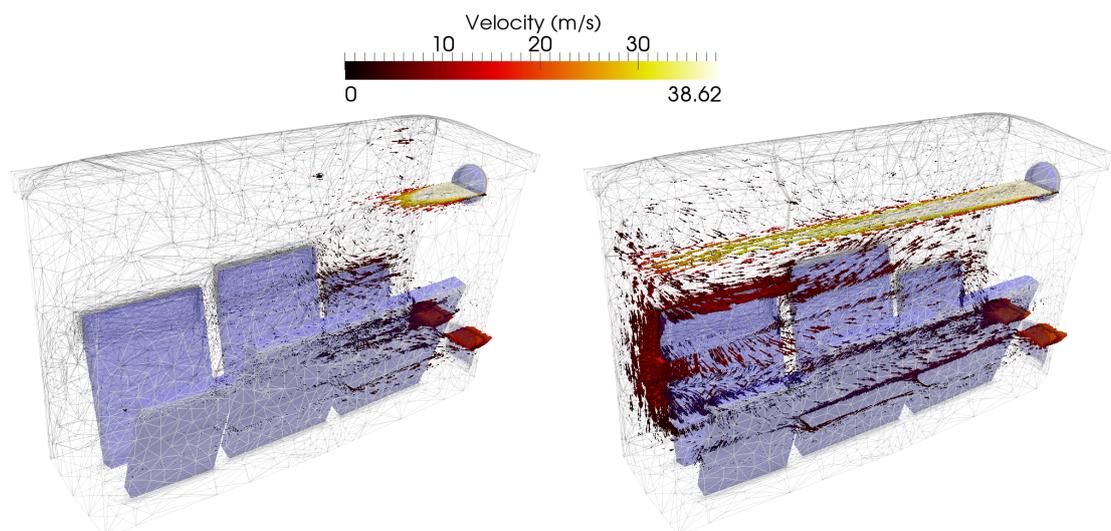


Figure 6.34 – Velocity field at the beginning and during the course of the simulation.

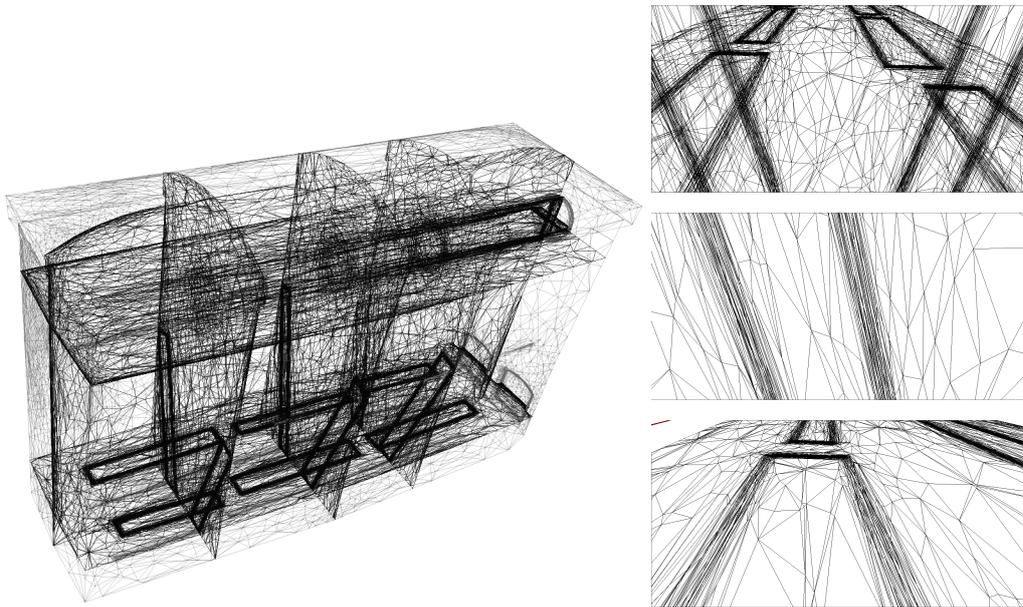


Figure 6.35 – Anisotropic mesh generated for the time-slab [26.05, 29.65].

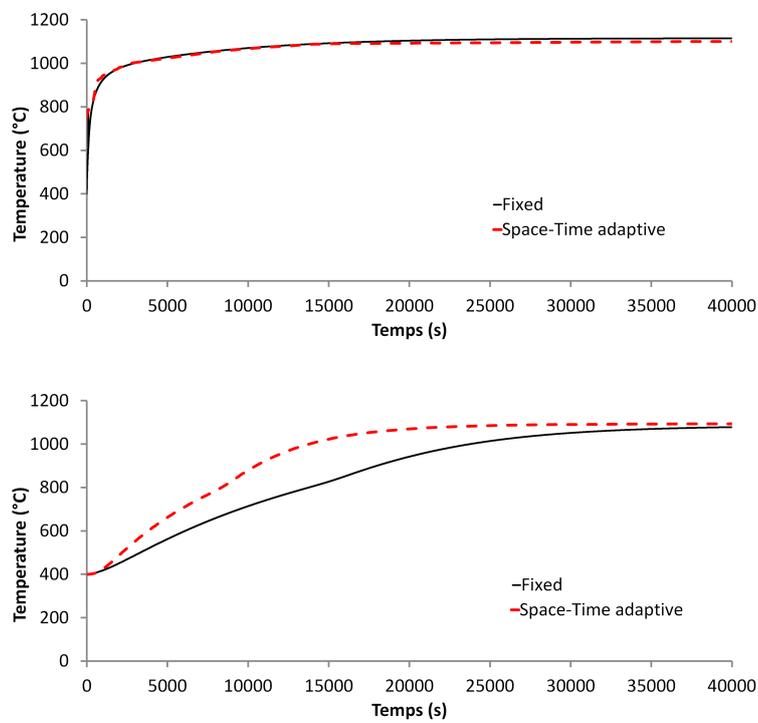


Figure 6.36 – Temperature distribution at selected nodes inside the volume (top) and at the heart of the central thick workpiece (bottom).

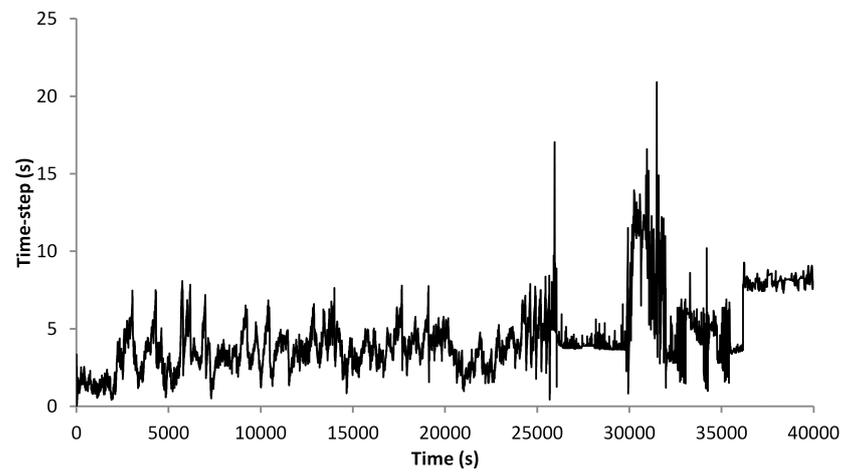


Figure 6.37 – History of the time-step sizes generated by the paradoxical meshing approach for a conjugate heat transfer application.



# INDUSTRIAL APPLICATIONS

“La sagesse est la vertu oubliée de notre temps. Un savant qui n’est pas aussi un sage est, soit dangereux, soit, dans le meilleur des cas, inutile.”

Amin Maalouf

## CONTENTS

---

|     |                                                                   |     |
|-----|-------------------------------------------------------------------|-----|
| 7.1 | INTRODUCTION . . . . .                                            | 321 |
| 7.2 | 2D QUENCHING OF A HEATED INGOT BY FORCED CONVECTION . . . . .     | 323 |
| 7.3 | AIR COOLING OF A HAT SHAPED INGOT BY NATURAL CONVECTION . . . . . | 329 |
| 7.4 | 3D HEATING OF AN INDUSTRIAL FURNACE . . . . .                     | 335 |
| 7.5 | 3D COOLING INSIDE A QUENCHING CHAMBER . . . . .                   | 346 |
| 7.6 | RÉSUMÉ FRANÇAIS . . . . .                                         | 355 |

---

**I**N this chapter, we are interested in evaluating the developed numerical tools in resolving large scale and long time computations. More precisely, we aim at validating the potential of the immersed volume method with an anisotropic mesh adaptation and time adaptive control and the stabilized finite element solvers to deal with heat treatment phenomena within complex 3D enclosures. The study will include thermal heating inside industrial furnaces and thermal cooling inside quenching chambers by either natural or forced convection. Heat transfer in its three forms conduction, convection and radiation will be accounted for. We recall that this work was developed in the context of the REALisTIC project with the main objective being the improvement of the accuracy and efficiency of the numerical solutions computed using the available solvers through the development of space and time adaptive algorithms. The numerical investigations will compare the results obtained using the anisotropic mesh adaptation and time adaptive algorithms to experimental data and to the numerical solutions computed on fixed meshes using fixed time-step sizes.



## 7.1 INTRODUCTION

Back to the iron age, when iron started replacing bronze in tools and weapons for its ductility, strength and resistance properties when alloyed with carbon, the mechanical and physical properties were considered as the essential characteristics of metals. It was in the late 1900s that iron was replaced by steel which showed to be more interesting in terms of flexibility and hardness for its higher content in carbon. Though not as brittle as cast iron, steel has a higher resistance and durability which makes it more reliable.

Nowadays, steel is a major component of a wide range of manufacturing industries including automotives, machinery, household equipments, and construction. The final steel product used in these appliances goes through a series of steel-making processes: iron ore extraction and processing, mixing with carbon materials, and heat treatment.

The heat treatment cycle consists in a sequence of controlled heating, quenching and tempering operations whereby a raw material steel undertakes metallurgical and mechanical changes to either convert it into molten steel or to change its shape (metal forging/rolling). During this treatment, a steel metal is loaded into a pre-heated furnace and placed on a skid to ensure its heating from the top and the bottom. Heated gas is then injected from the burners yielding a combustion process and releasing chemical and thermodynamic energy in the heating chamber. This energy will then be absorbed by the workpieces changing their metallurgical and physical properties. The heating process will be conducted for several hours. When the temperature of the steel metal reaches its critical value (austenitic temperature), the furnace door is opened, the heated ingot is removed and a new one is loaded. Then the heated workpiece is transported to a quenching bath or it is left in the large to cool down via natural convection. Figure 7.1 presents the extraction of a heated large hollow steel shell from the furnace and its transportation to the quenching chamber. We note that these images were supplied by our industrial partners at *Areva, Creusot Forge*.



Figure 7.1 – Heat treatment of a large hollow steel shell.

An accurate control on this process and a good calibration of heating sequences and energy injection results in a good quality production. However, heat treatment requires high energy consumption, indeed a typical industrial furnace consumes between 700 – 4000kWh per ton of steel [Von Schéele 10]. This includes the chemical energy from the burners as well as the chemical reactions within the load. Driven by the need to respect the new regulations for reducing energy consumption and pollutant carbon emissions

while maintaining a high quality production, industrials are interested in limiting the energy losses during the heating process. Different forms of losses can be identified as stated in [Was 14] and shown in figure 7.2. One form of losses is issued from the pre-heating of the furnace's insulation structures which store heat all over the ingots' treatment process then release it once the workpieces are removed and the furnace is turned off. The second form is caused by the heat conduction at the walls' level, roof and workpieces support positioned on the floor. Other forms of losses are related to the heat release when opening the door of the furnace to insert/remove workpieces, and to the radiations effect with the external surrounding. The most important form is the waste-gas (also known as flue losses) which manifests in the heat that cannot be removed from the combustion gases. It was also emphasized in [Was 14] that this waste is closely related to the production volume and in order to optimize the energy consumption it is important to maintain the furnace efficiency around 100%. Two main factors affect the furnace efficiency: thermal scheduling and furnace loading. The thermal *scheduling* refers to the planning of the heat treatment sequences, calibration of the burners' temperatures over time and defining the charges' insertion times whereas *loading* stands for the amount of ingots inserted in the furnace over a given time. Hence in order to keep the furnace efficiency around 100%, optimization analysis shall be carried on the scheduling and loading aspects of the heat treatment cycle.

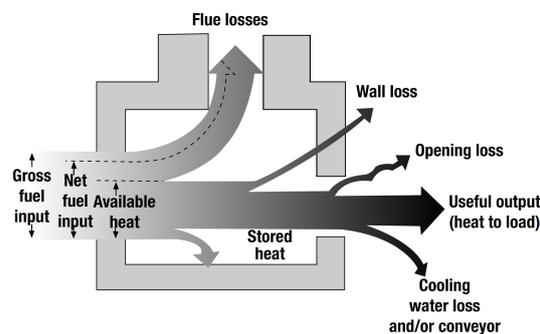


Figure 7.2 – Heat losses during a furnace heating process taken from [Was 14].

To do so a better understanding of the heat treatment cycle needs to be acquired in the view of devising a thermally and energy efficient heating schedule and optimizing the duct throughput and microstructure of the workpieces. This can be achieved either by experimental or by numerical analysis. Based on the thermal history of the workpieces and temperature distribution inside the furnace, the quality of the final product can be deduced in terms of flexibility, ductility and resistance. However, in order to conduct experimental studies, raw materials and machinery are needed. Moreover, experimental testings on the design of the furnace, the location of the burners, the thermal schedule and the positioning of the ingots necessitate material resources and require a long time. Therefore, these studies can rapidly become unfeasible. On the other hand, with the continuous growth in computer resources and development of numerical tools, the modelling of heat transfers and fluid flows inside industrial compartments (furnaces, quench chambers, pools, ...) is now possible. This modelling

involves computationally solving conjugate heat transfer problems with a good level of accuracy and a reasonable computational cost. The simulation is very challenging as it involves solving simultaneously heat transfers between the solid bodies and the surrounding fluid, radiation due to the interaction between the solid bodies and also their interactions with the walls, conduction and turbulent fluid flows.

Prior to this work, finite element stabilized solvers for the modelling of the conjugate heat transfer process were developed within CEMEF. With these tools a real full scale simulation was unaffordable as it required very long computational time and a high fidelity mesh [Hachem 09]. The REALisTIC project, in the scope of which the work of this thesis stands, intends to design and develop adaptation techniques capable of reducing the execution time by accelerating the numerical prediction of heat treatment while achieving quality solutions with good precision.

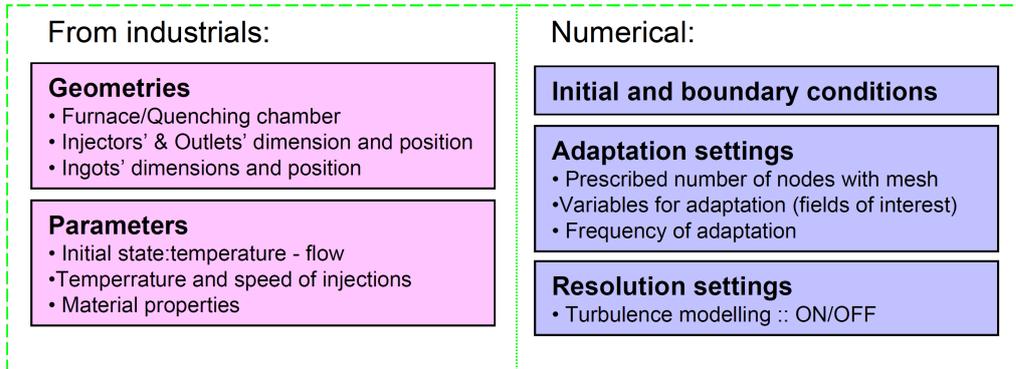
In that regard, we aim in this chapter to validate the developed anisotropic adaptation tools and the modified stabilized formulations in their capacity to reduce the computational time of simulations. Consequently, they can be used for optimization and thermal control strategies. For our industrial partners, this means easier analysis of 'what-if' scenarios and sensitivity of parameters, material properties and geometric setups. We note that in these simulations some simplifications were adopted to cope with the currently available solvers. In particular, the combustion processes at the burners' levels were ignored and a simplified radiation model, the  $P_1$  model, was considered for the computation of the radiation transfers although this model has shown to be diffusive and might yield an over prediction of the radiation effect on the global solution when the temperature's gradient is important as it employs a uniform radiation distribution in all the directions. We recall that a more accurate model, the  $M_1$  model, that takes into account directional dependence of the radiation energy is currently under development in the thesis work of Schmidt [Schmidt 16]. Bearing the above limitations on the numerical tools, we have taken into account the real setup in terms of full geometric design of the problem, the material properties of the workpieces, and the true initial and boundary conditions.

We depict in figure 7.3 the general diagram for the simulation of a heat treatment inside an industrial heating or quenching chamber.

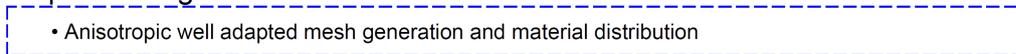
## 7.2 2D QUENCHING OF A HEATED INGOT BY FORCED CONVECTION

As a first validation of the accuracy and efficiency of the developed tools on industrial applications, we consider an air cooling problem of a crown shaped Inconel-718 solid by forced convection. The solid having a thickness of around 10cm is initially heated up to 1030°C and placed inside the quenching chamber. Air at 20°C is pumped into the enclosure at a constant stream of 11.55m/s from a ventilation device positioned at the bottom wall. Adiabatic temperature, a Neumann boundary condition  $\nabla \mathbf{v} \cdot \mathbf{n} = 0$  and a pressure  $p = 0$  are prescribed at the top wall. Slip conditions and adiabatic temperature are assigned on the other walls. The material properties of the air and the

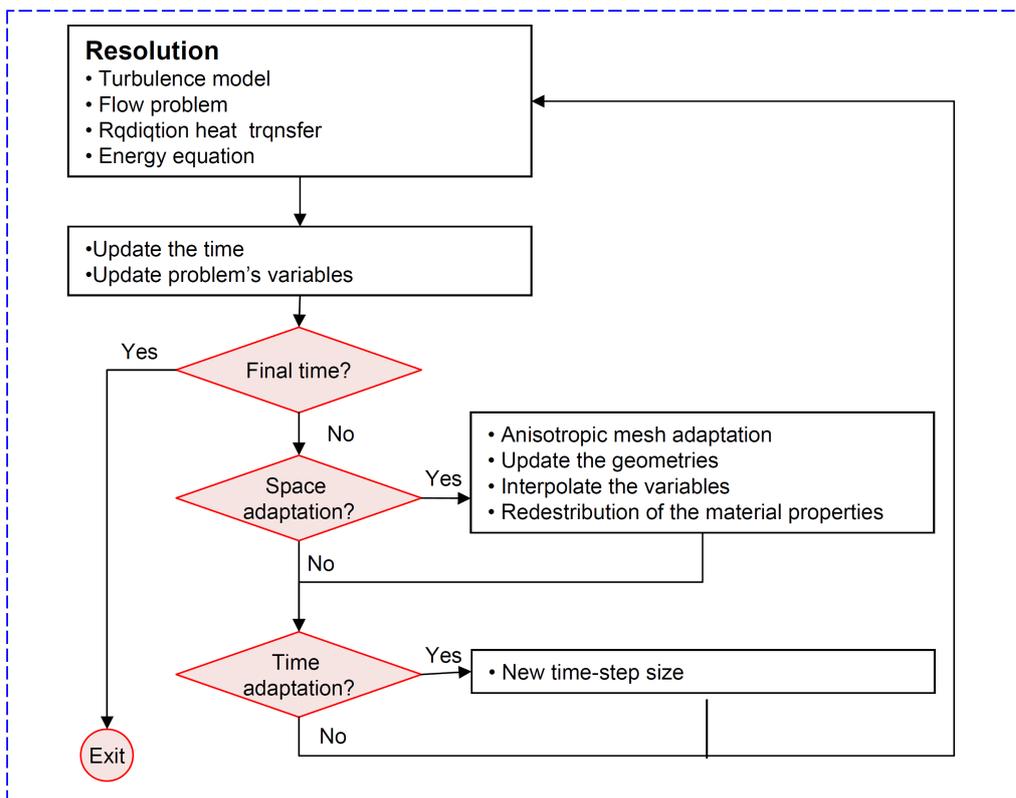
Input



Pre-processing



Processing



Post processing

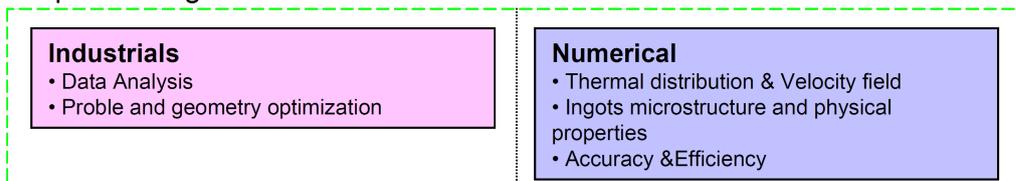


Figure 7.3 – General diagram for heat treatment simulation.

ingot are updated dynamically in accordance with their temperatures. The computational domain, the initial mesh and temperature and the thermocouples' positions are depicted in figure 7.4.

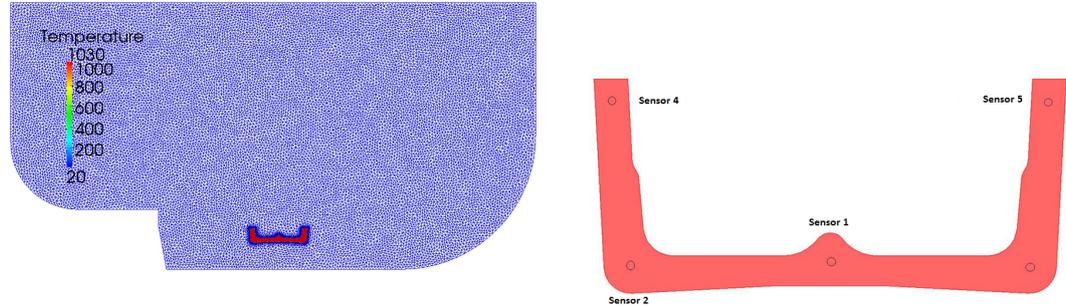


Figure 7.4 – Computational domain, initial mesh and temperature (left) and thermocouples' positions (right).

For comparison purposes, we restrict the analysis to the 2D case, for which numerical validation with the commercial CFD software package, FLUENT, were provided to us by our industrial partner Aubert& Duval. The objective is to simulate the first 600s of forced convection where the radiation effects can be neglected and compare our numerical tools. The Immersed volume method is used to immerse the solid object into the domain and distribute the material properties. A single set of equations is solved simultaneously for both fluid and solid domains. The variational multiscale approach is employed to stabilize the fluid flow solution whereas the SCPG method is used to preclude numerical oscillations at the locations of convection domination and sharp gradients especially in the vicinity of the interface. Turbulent effect in the flow are accounted for using the  $k - \varepsilon$  model with standard logarithmic wall functions defined in [Lauder 74] and [Han 97]. Therefore, as described in chapter 1, we solve the RANS problem with two additional transport equations accounting for the turbulent kinetic energy  $k$  and the turbulent dissipation  $\varepsilon$ . This modelling introduces two additional terms, the effective viscosity and the effective conductivity that will be used in the RANS equations:

$$\mu_e = \mu + \mu_t \quad \text{and} \quad \lambda_e = \lambda + \frac{C_p \mu_t}{Pr_t}, \quad (7.1)$$

with  $Pr_t = 0.89$  being the turbulent Prandtl number and  $\mu_t$  the turbulent viscosity computed by:

$$\mu_t = \rho C_\mu \frac{k^2}{\varepsilon} \quad (7.2)$$

with  $C_\mu$  an empirical constant set to 0.09.

At the inflow boundary, we impose:

$$k_{\text{inlet}} = c_{bc} \cdot |\mathbf{v}|^2, \quad (7.3)$$

where  $\mathbf{v}$  refers to the velocity and  $c_{bc} = 0.02$  is an empirical constant. We then determine

the value of  $\varepsilon$  at the inlet by:

$$\varepsilon_{\text{inlet}} = \frac{C_\mu \cdot k^{3/2}}{L}, \quad (7.4)$$

where  $L$  stands for the characteristic length of the model [Lauder 74]. We note that these values of  $k$  and  $\varepsilon$  are also imposed as initial conditions on the domain's interior. Homogeneous Neumann boundary conditions are assigned on the outflow boundary:

$$\mathbf{n} \cdot \nabla k = 0 \quad \text{and} \quad \mathbf{n} \cdot \nabla \varepsilon = 0. \quad (7.5)$$

We use the classical wall function [Lauder 74] on the rest of the computational boundary to describes the asymptotic behavior of the different variables near the wall. Accordingly, when the boundary mesh nodes are located in the logarithmic region, wall shear stress which serves as non-homogeneous Neumann boundary condition for the momentum equation in the tangential direction is computed:

$$\tau_w = \rho V^{*2}. \quad (7.6)$$

On the other hand, the normal component of the velocity is set to zero. In equation 7.6,  $V^*$  refers to the friction velocity, which is the solution to the following nonlinear problem:

$$\frac{V_\tau}{V^*} = \frac{1}{\kappa} \ln \left( \frac{\rho E \delta}{\mu} V^* \right), \quad (7.7)$$

inhere,  $V_\tau$  is the tangential velocity,  $\delta$  the distance to the wall,  $\kappa = 0.41$  the Von Karman constant and  $E = 9.0$  is a roughness parameter.

A function of the friction velocity [Lauder 74] is used to define the boundary values of the turbulent kinetic energy and its dissipation as follows:

$$k_w = \frac{V^{*2}}{\sqrt{C_\mu}} \quad \text{and} \quad \varepsilon_w = \frac{U^{*3}}{k_w \delta}. \quad (7.8)$$

Furthermore, a temperature wall function enables to define the boundary condition for the energy equation where the effective heat flux in the wall function is being computed by:

$$q_w = \mathbf{n} \cdot \mathbf{q}_w = \frac{\rho C_p C_\mu^{1/4} k_w (T_w - T)}{T^+}, \quad (7.9)$$

where  $T_w$  denotes the wall temperature and

$$T^+ = 2.1 \ln(y^+) + 2.5,$$

refers to the normalized temperature as suggested in [Han 97].

It is important to mention that using the FLUENT software conjugate heat transfer problem is decoupled into two phases whereby in the first phase, the flow problem is solved until the velocity field stabilizes then using the obtained steady state velocity profile the energy equation is solved yielding the desired temperature field.

In our numerical simulation, the fully coupled problem is being solved, and the classical anisotropic mesh adaptation and time adaptive algorithm were applied. The mesh is adapted on the temperature field, the velocity norm and direction every 5 time-increments with a fixed number of nodes equal to 20,000. We highlight the challenge in remeshing the domain as it includes curvature and sharp angles on its boundary. Figure 7.5 presents the temperature's evolution over time together with the corresponding anisotropically adapted meshes. We can detect how the mesh refinement is localized and responds to the characteristics of the problem. It mainly reflects the trajectory of the small scale plumes of temperature and the direction of the velocity field that is guiding them. We report in figure 7.6 the history of temperature at different locations in

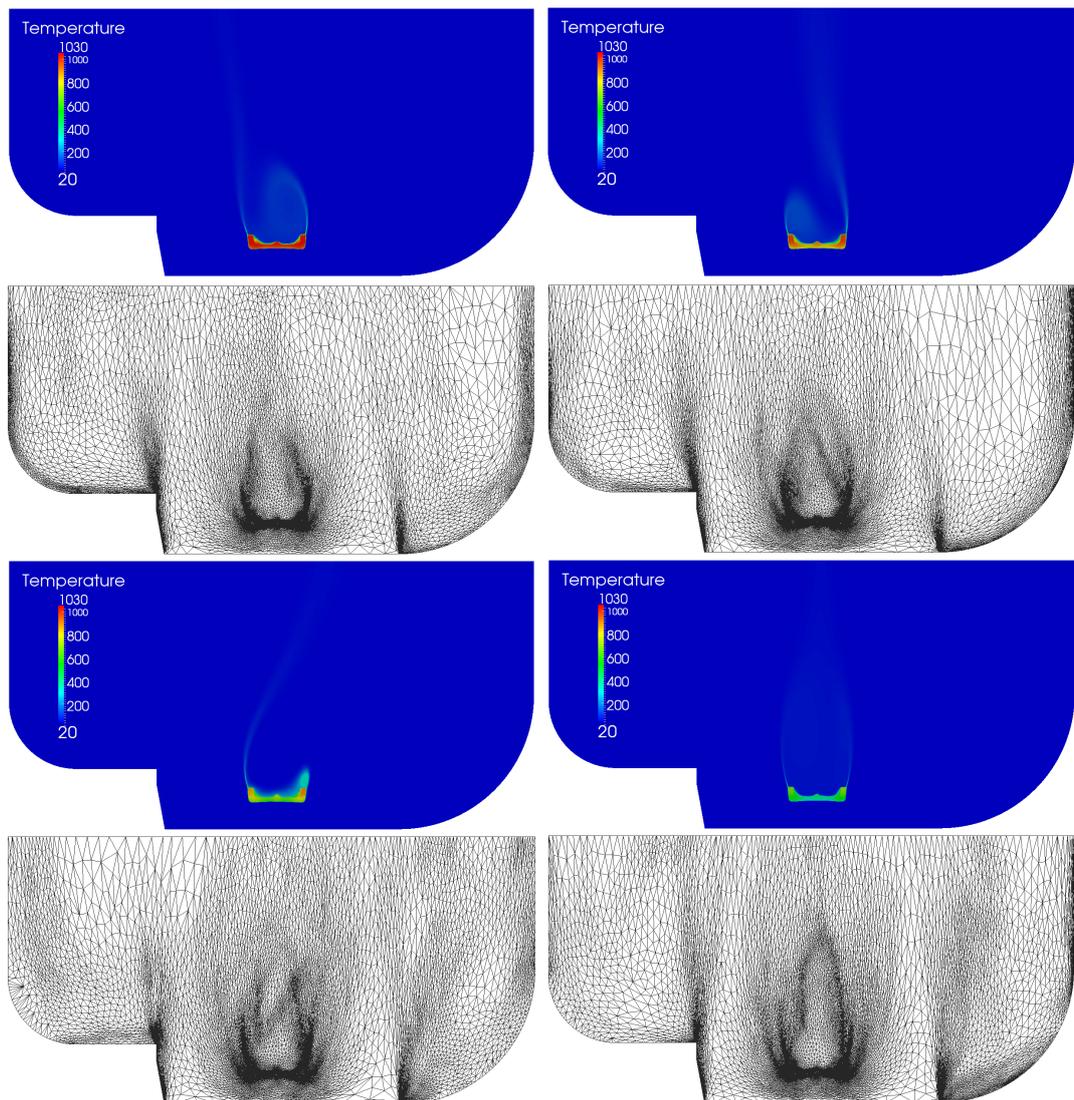


Figure 7.5 – Temperature profiles and corresponding meshes at times  $t = 53s, 175s, 323s, 600s$  (from top to bottom and left to right).

the workpiece. A good agreement can be noted between the results computed with the FLUENT software and the present work. The slight difference can be associated with the procedural difference in solving the problem since in here we solve a fully coupled system and use the immersed volume method whereas in the reference, a decoupled

system is solved. To validate the anisotropic adaptation and modified stabilized flow solver accounting for anisotropic meshes, we compare the velocity distribution at different horizontal cuts at time  $t = 600s$  with the ones obtained with FLUENT. The velocity field at the final time and the positions of the horizontal cuts are provided in figure 7.7. We plot the  $v_y$  component of the velocity field along the different cuts and report

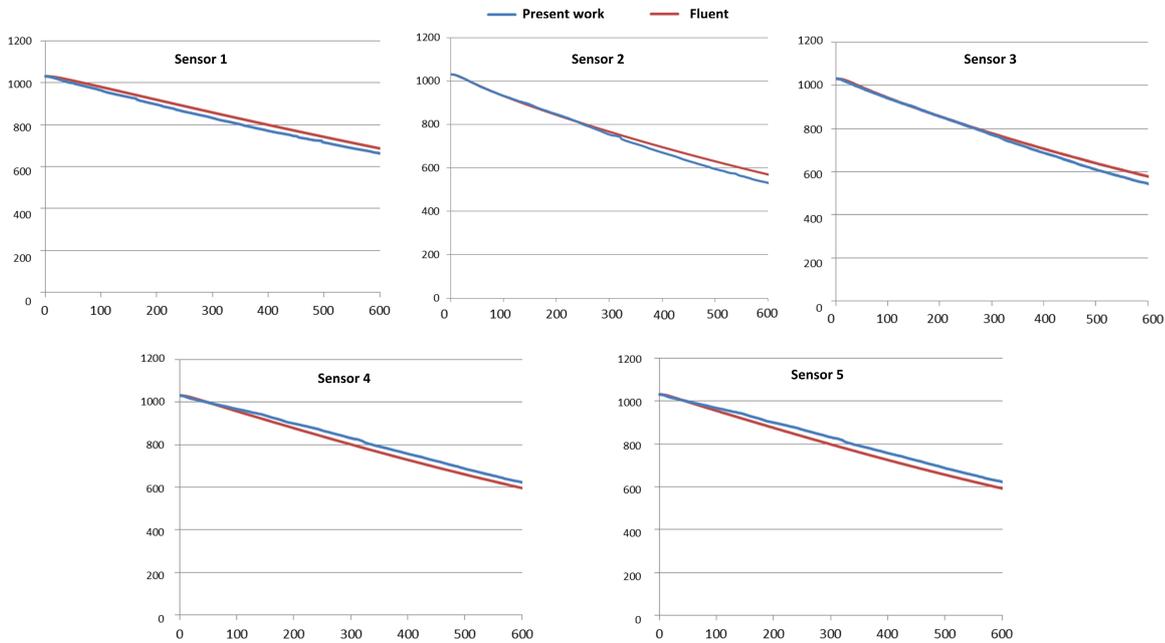


Figure 7.6 – History of temperature at different sensors inside the workpiece.

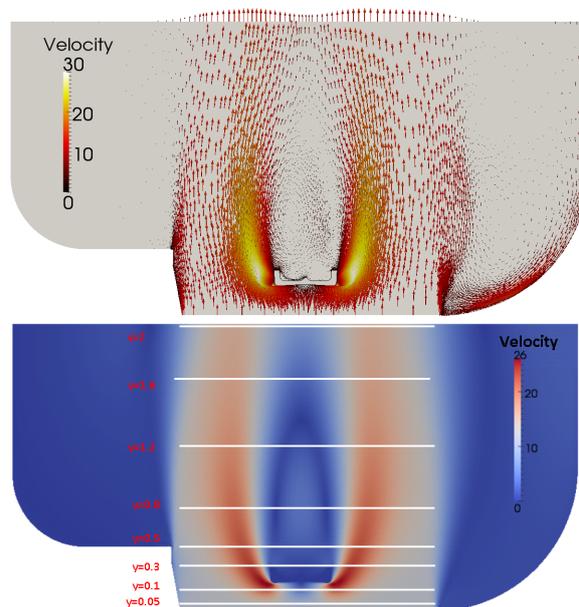


Figure 7.7 – Velocity field (top) and position of the horizontal cuts (bottom).

the comparisons in figure 7.9. We point out the very good agreement in the velocity field between the two approaches at the bulk region. Nevertheless, in the near-wall region, some differences can be spotted. One possible reason for this disagreement can be related to the difference in the used wall functions. Another possible reason is

the difference in the approaches: an immersed volume method from one hand and a decoupled resolution from the other hand. The time-step sizes' evolution over time is

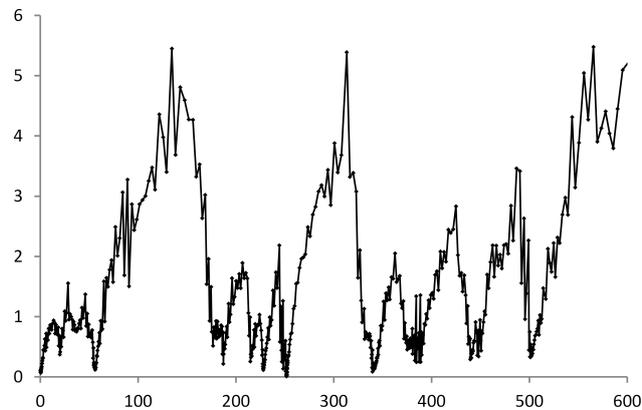


Figure 7.8 – History of time-step evolution for the forced convection quenching of a heated workpiece.

provided in figure 7.8. It shows an oscillatory profile with a tendency to generate large time-step sizes. We recall that the use of a turbulence model to account for the effect of the emerging small eddies on the global solution reduced the scale heterogeneity of the velocity field captured by the adaptation algorithm and consequently helped in accelerating the computations. The simulation necessitated 56 minutes using 8 2.4 Ghz Opteron cores.

Therefore, in this example the numerical tools developed in this thesis were confronted with a well known software package and reflected good performance in capturing the quenching process in a reasonable execution time.

### 7.3 AIR COOLING OF A HAT SHAPED INGOT BY NATURAL CONVECTION

The objective of this test case is to validate the performance of the developed space-time adaptive methods and the modification of the solvers to cope with highly anisotropic elements on a three dimensional industrial application. The problem studies the heat transfers by conjugate natural convection and thermal radiation involved in an hour of cooling process of an Inconel-718 hat shaped ingot initially heated to 1160°C and embedded inside an air filled cubical cavity.

The interest from an industrial perspective is to understand the characteristics of the flow and ingot's thermal history in the view of benefiting to the maximum from this naturally occurring quenching process without resorting to cooling devices. The numerical analysis can serve them for optimizing the heat transfer effect in terms of the enclosure's volume and to determine the needed spacing between ingots when several workpieces are to be cooled in the same duct.

The computational domain, the initial mesh and the immersed solid geometry are given in figure 7.10. The air inside the enclosure is initially at rest and has a temperature of 20°C. The bounding walls are maintained adiabatic throughout the computations. A Neumann boundary condition  $\nabla \mathbf{v} \cdot \mathbf{n} = 0$  and a pressure  $p = 0$  are prescribed at the

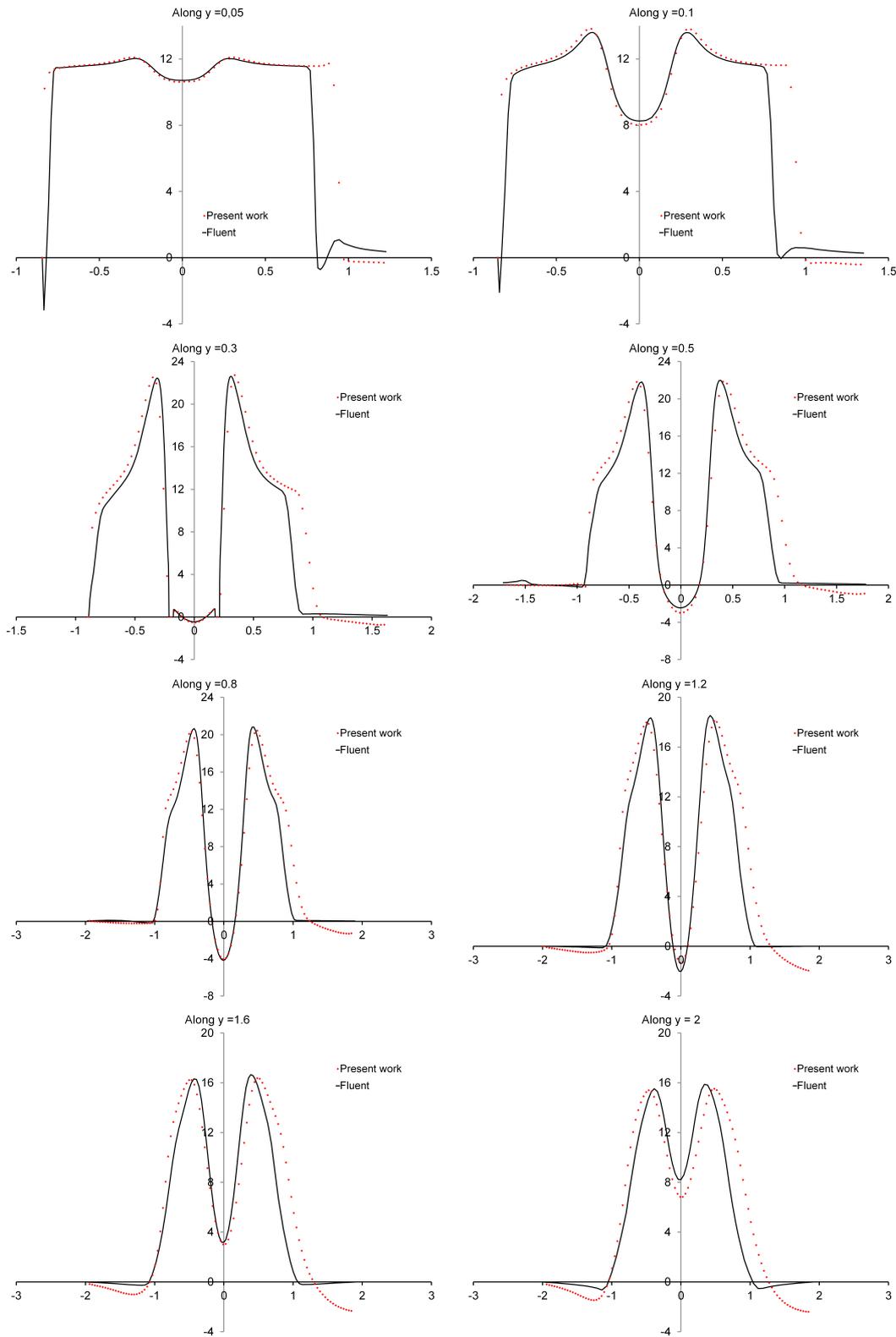


Figure 7.9 – Comparisons of the velocity field's component  $v_y$  along the different horizontal cuts at the final time 600s.

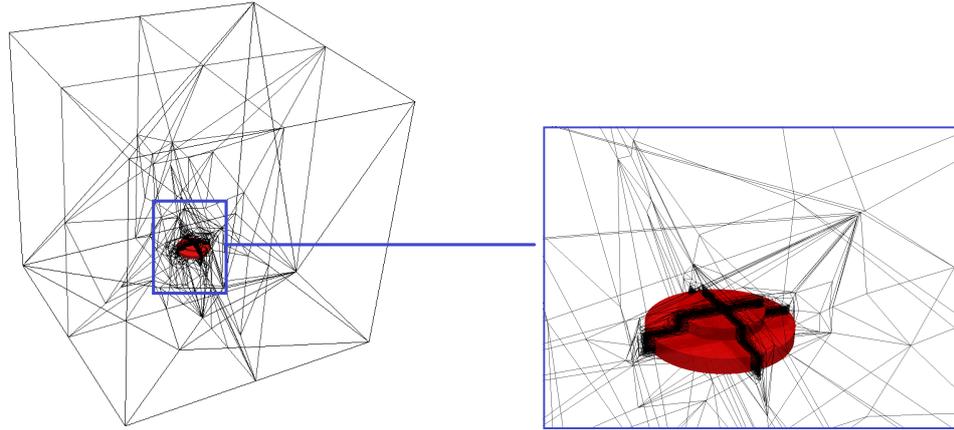


Figure 7.10 – Computational domain, initial mesh and the iso-zero value of the immersed solid.

top wall whereas a slip condition is assigned on the other walls. The immersed volume method is adopted to embed the solid and to distribute the material properties over the domain. We note that the thermophysical properties of the air and the ingot are updated dynamically in accordance with their temperatures. The initial mesh, shown in figure 7.10, is adapted using the newly developed anisotropic mesh adaptation tools based on the filtered levelset function. We can clearly detect the sharp representation of the solid's interface and the highly stretched elements away from the ingot. This reflects how for a controlled fixed number of nodes (around 50,000), the anisotropic mesh adaptation algorithm generates an optimal representation of the desired field. Using the obtained mesh, the levelset function and the mixing laws, the material properties are distributed over the domain. We recall that harmonic mean formulation enables to well render the sharp discontinuity in the thermal conductivity field. We point out that a high relative kinematics viscosity is prescribed inside the solid region yielding a negligible velocity at that level. Consequently, the heat transfers are limited to the conduction inside the workpiece. The natural convection problem under the Boussinesq assumption (explained in section 1.6.2) is solved coupled with the thermal radiation and heat conduction. The medium is supposed to be gray, emitting and absorbing radiation, and isotropically scattering. Under this assumption, the mean absorption coefficient  $\kappa_a$  is determined from the ingot's emissivity respecting the Bouguer's law given by:

$$\kappa_a = -\frac{1}{L_m} \ln(1 - \epsilon), \quad (7.10)$$

where  $L_m$  represents the mean beam length and is computed as follows:

$$L_m = 3.6 \frac{\Delta V}{\Delta S}, \quad (7.11)$$

with  $\Delta V = \Delta x \Delta y \Delta z$  and  $\Delta S = 2(\Delta x \Delta y + \Delta y \Delta z + \Delta z \Delta x)$  being respectively the volume and surface of each element in the mesh [Siegel 02].

In this problem, the IVM permits a natural treatment of solid/air heat transfers without a priori definition of a heat transfer coefficient which is usually determined through

experimental studies. Subsequently, if the geometry were to be changed the same simulation can be leveraged without modification which makes of these tools an important asset for industrial investigations.

In order to reflect on the thermo-dynamics inside the cavity induced by the natural convection phenomena, we perform a first simulation while ignoring the radiative effects. Figure 7.11 presents snapshots on the temperature's distribution over time (*left*) with the corresponding anisotropic meshes (*right*). The obtained fluid motion inside the cavity respects the physics of the problem. Buoyancy forces induced by the gravitational acceleration and the thermal gradient in the duct are at the origin of this motion. Indeed, the air in contact with the hot solid gets warmer and thus lighter. Consequently, in response to the buoyancy forces, this air rises whereas heavier one falls down creating a fluid motion inside the enclosure. This is exactly what we observe in figure 7.11(left) where ascending thermal plumes of heated air can be detected. Moreover, a Marangoni effect can be spotted whereby the fluid that is getting in contact with the outer interfaces of the hat shaped solid moves along that interface from hot to cold regions then it is dragged by the fluid at that level. We report in figure 7.12 the temperature distribution inside the steel workpiece at different time instances, we can observe how the solid cools down and how the heat is transferred between the two subdomains.

In this application, we adapted the mesh every 10 time-increments with a fixed number of nodes  $N = 50,000$  on the  $L^2$  norm of the interpolation error based on the temperature field and the velocity's direction and norm. The anisotropic mesh adaptation algorithm responds very well to the thermal variations and the induced flow motion by strongly refining the regions with sharp gradients and highly stretching the elements in the rest of the domain thus optimizing to the best the use of the prescribed nodes. A dynamic and automatic follow of the developing flow can be identified ensuring its ascendance. In addition, a precise capture of the interface is maintained all over the simulation resulting in an accurate heat transfer between both sides of the interface.

From a quantitative perspective, experimental analysis was also conducted on the hat shaped solid which was equipped with several sensors at different positions inside the workpiece as shown in figure 7.13. The thermal history has been registered through a data acquisition device and provided to us by our industrial partner. We compare in figure 7.14 the temperature variations at sensors 4, 6, 12 and 13. A very good agreement is noted between the experimental data and the numerical distribution.

We note that in order to validate the importance of the radiative effects on the simulation, we have carried out another experiment accounting only for the heat transfers due to natural convection. We report in figure 7.15 the temperature variations on sensors 1, 5, 8 and 14 collected from the experimental studies and the numerical tests. We point out that the selected sensors are the most critical ones since two of them are very close to the interface and reflect the accuracy of the heat transfers at that level and the third is at the bulk of the solid. It can be inferred from the obtained results that the radiative exchanges play an essential role on the temperature distribution and

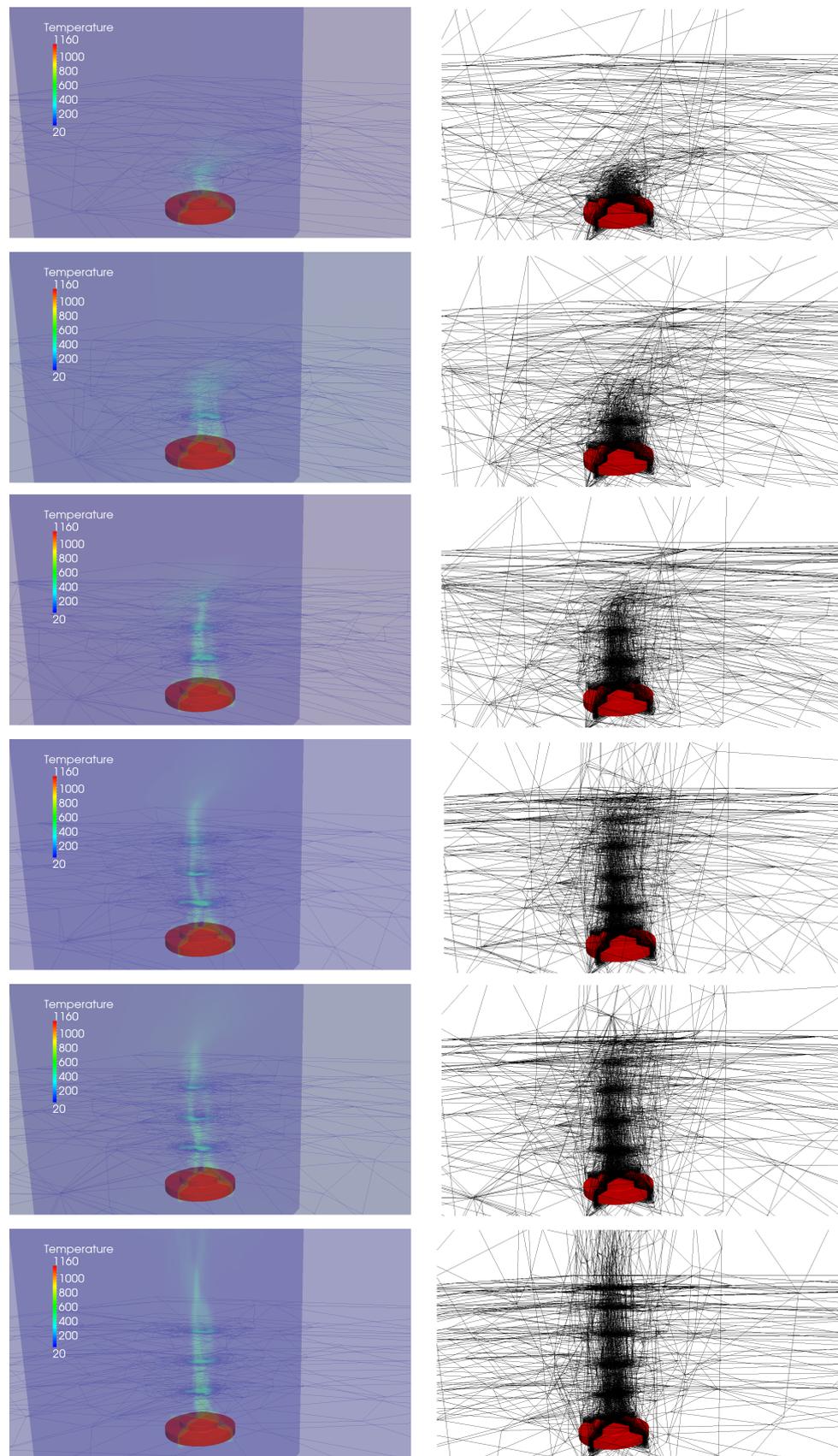


Figure 7.11 – Thermal evolution inside the enclosure at different time instances with the corresponding anisotropic meshes.

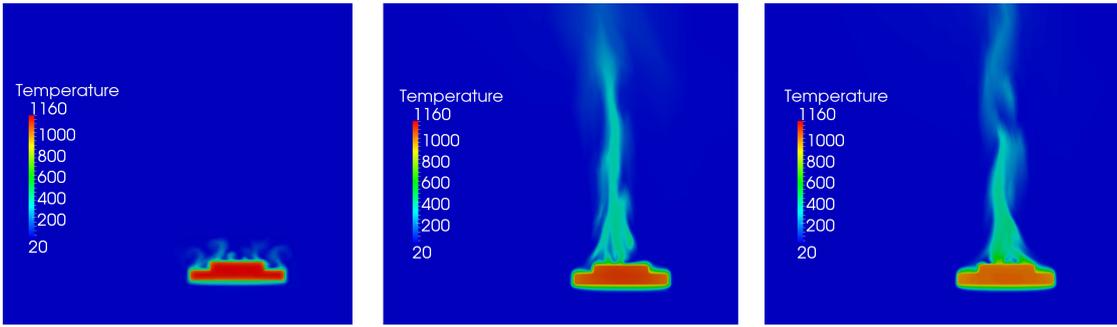


Figure 7.12 – Temperature distribution inside the hat shaped ingot at times  $t = 40s$ ,  $t = 600s$  and  $t = 1600s$ .

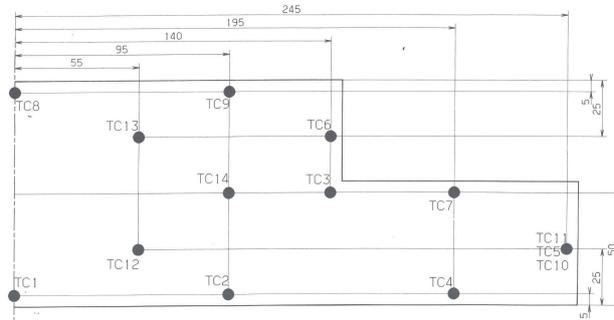


Figure 7.13 – Thermocouples' positioning inside the hat shaped ingot.

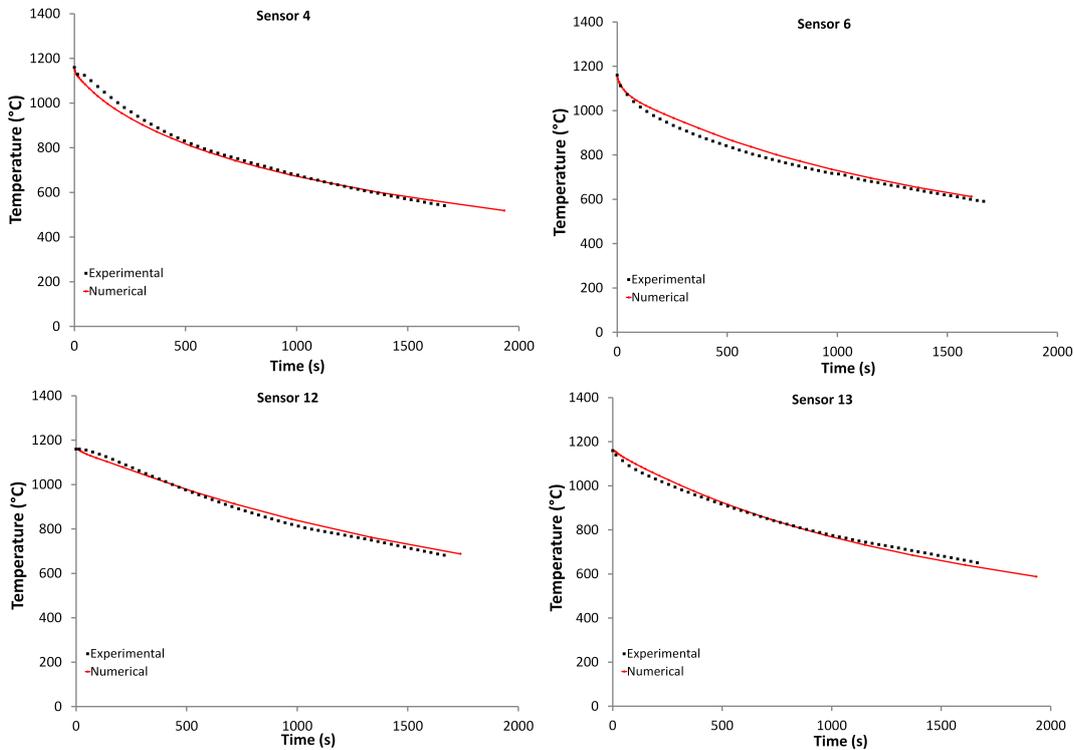


Figure 7.14 – Temperature evolution at different locations inside the hat shaped ingot.

the physical characteristics of the flow. On the other hand, the slight discrepancies (less than 5%) depicted at the surface of the ingot can be associated with the diffusive character of the  $P_1$  radiation model as it does not account for directional emissions.

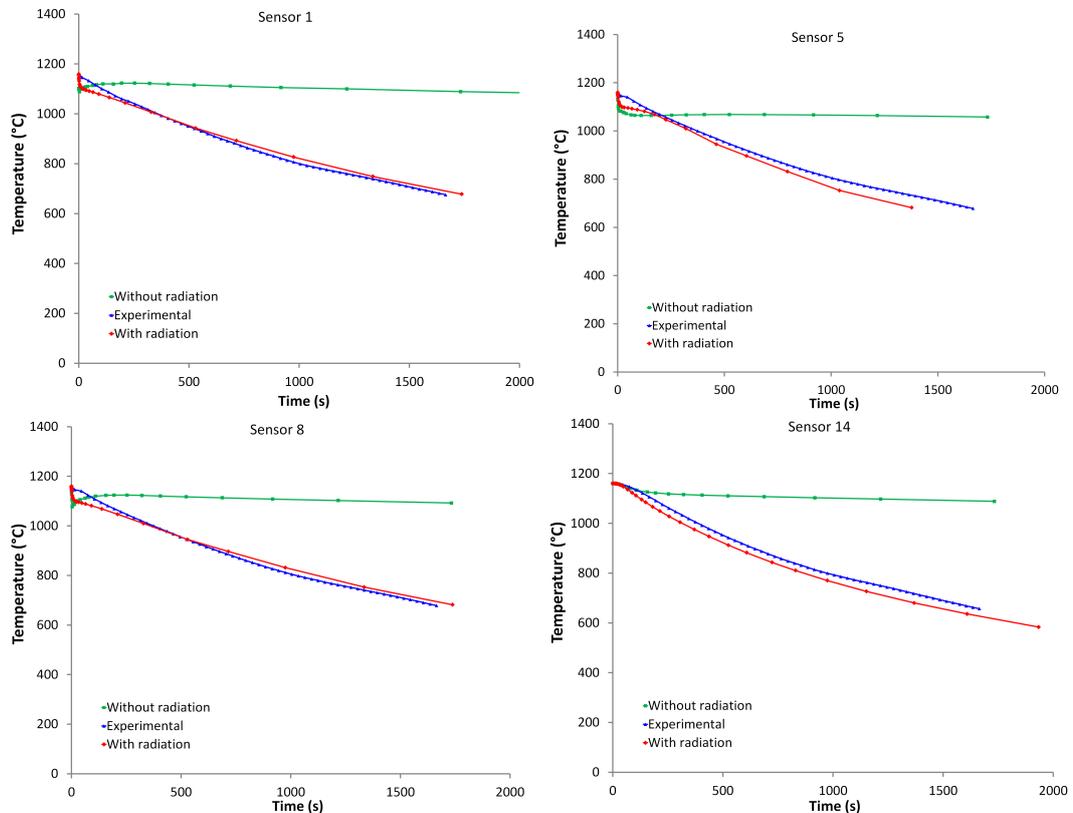


Figure 7.15 – Temperature evolution at different locations inside the hat shaped workpiece, with and without thermal radiation effects.

Regarding the efficiency of computations, we present in figure 7.16 the time-steps generated by the time adaptive algorithm to advance the solution. We notice that the method starts by generating small values in order to follow up with the developing flow. Once the motion is well developed inside the enclosure, large time-steps are produced thus reducing the global execution cost. It is important to mention that the simulation took only 32 minutes to get to 1600s with the developed space-time adaptive algorithm.

## 7.4 3D HEATING OF AN INDUSTRIAL FURNACE

In this numerical example, we aim at investigating the conjugate heat transfer and turbulent flows inside an industrial furnace. The validation is twofold: on the one hand, we validate the ability of the numerical solvers to produce an oscillation free solution, and on the other hand, we investigate the performance of the immersed volume method (IVM) with the developed anisotropic mesh and time adaptations on a three dimensional complex geometry and highly coupled physical phenomena.

The geometry of the problem (fig. 7.17) consists in a three dimensional cylindrical

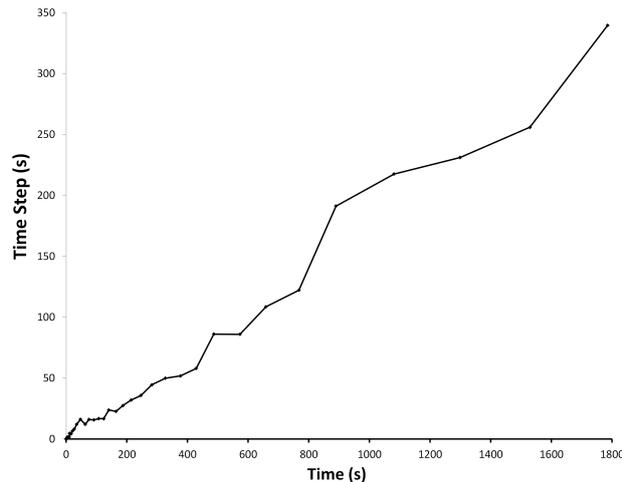


Figure 7.16 – Generated time-steps for the cooling of a hat shaped solid.

duct instrumented with 4 inclined burners and a rectangular prismatic outlet centered at the top wall of the enclosure. For confidentiality purposes, we omit the details on the geometry and the parameters' scales. A cylindrical Inconel-718 ingot with diameter 200mm and height 280mm is immersed at the center of the bottom wall.

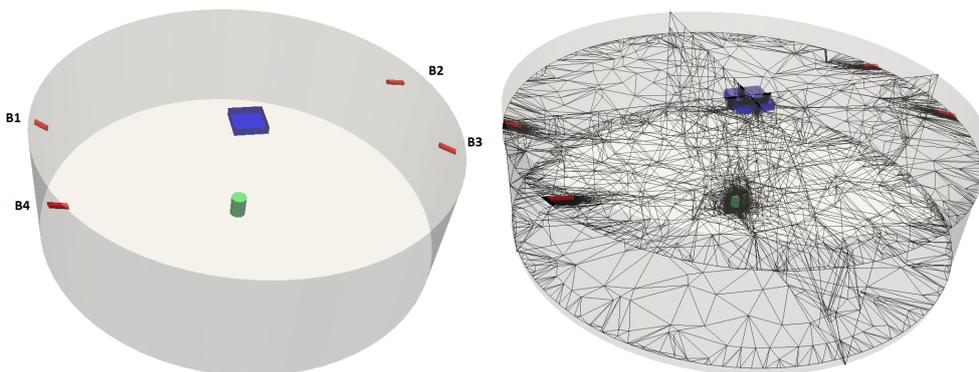


Figure 7.17 – Geometry and initial mesh of the 3D furnace.

The temperature in the domain is initially set to  $T_{\text{domain}}$ , heated gas is injected into the furnace at a temperature  $T_{\text{max}}$  at speeds  $v_{\text{inlet}}^1$  from burners B1 and B2 and  $v_{\text{inlet}}^2$  from burners B3 and B4, with  $v_{\text{inlet}}^1 > v_{\text{inlet}}^2$ . We note that  $T_{\text{domain}}$  is taken to be  $0.8T_{\text{max}}$ . At time  $t = 840\text{s}$ , the last two burners are turned off. The air is vented out through the outlet at the top wall. On the other walls, slip boundary conditions and adiabatic temperature are maintained all over the simulation. At the level of the immersed solid geometry, the no-slip boundary condition is prescribed by imposing a high kinematic viscosity. Before starting the computations, an anisotropic mesh, of around 250,000 nodes, well adapted to the initial temperature field and the iso-zero value of levelset functions corresponding to the immersed solid, the burners and the outlet was generated using the newly developed tools. Based on this high quality mesh, the material properties are distributed over the domain. We note that the physical properties of the air inside the furnace and the solid object are changed dynamically

with respect to their temperatures. The radiative effect are accounted for under the gray gas assumption.

The workpiece is initially taken at 23°C and the objective is to study three hours of the heating process. The anisotropic mesh adaptation is applied every 10 time-increments using a fixed number of nodes 250,000 based on the temperature field, the velocity norm and direction, and the ingot's levelset function. The 3D computations were run in parallel on 64, 2.4Ghz Opetron cores linked by an Infiniband network. Figures 7.18 and 7.19 present respectively the evolution of the temperature field and the velocity field and streamlines inside the enclosure at different times. We can see how the temperature spreads into the domain in accordance with the fluctuating motion of the flow. Moreover, we can clearly see how the temperature's profiles do not suffer from numerical instabilities or spurious oscillations that appear in the presence of sharp gradients near the interface. This validates the adopted stabilized finite element approaches and the appropriate smoothing of the material properties. The streamlines, presenting a rotational behavior, ensure the good homogenous distribution of the temperature field all over the enclosure which is an essential asset for the final microstructure and material properties of the workpiece. We point out as well that the hot air pumped from the burners do not directly tap the solid which causes a thermal shock and results in a defected final product. Instead, it spins within the furnace increasing its global temperature and hence the ingot's temperature will be increased progressively avoiding the appearance of thermal shocks. The flow is decelerated by the ingot and forms a counter-rotating swirling around it. This contact between the cold solid and the warm fluid favors the heat transfers and enables the heating of the workpiece. The flow's movement is changed at that level and it ascends the cavity to the top wall where it exists the enclosure dragging with it the cooled air.

We provide in figure 7.20 the anisotropic meshes corresponding to the temperature field and the velocity streamlines shown in figures 7.18 and 7.19. We can clearly identify the directional feature of the mesh which reflects very well the characteristics of the problem. The elements are well refined along the gradients of the adaptation fields and highly elongated in their orthogonal direction. We observe how the mesh is dynamically and optimally updated to ensure an accurate capture of the desired fields. The boundary layers and the developing recirculations are sharply represented. Indeed, when the four burners are turned on at the beginning of the simulation, a sharp gradient is induced due to the interaction between the air at rest inside the volume and the injected fluid. The anisotropic mesh adaptation algorithm responds to this sharp change in the flow motion by refining the zones of steep gradients. Moreover, we can identify how well refined the mesh is anticipating the propagation of the flame inside the enclosure and hence ensuring its accurate capture. When the burners 2 and 3 are turned off and their corresponding flames fade out, the mesh dynamically and automatically de-refines their locations optimizing the distribution of the nodes and hence permitting a better control of the interpolation errors. Once the incident flow hits the opposite wall, a boundary layer is formed and the mesh responds systematically by re-

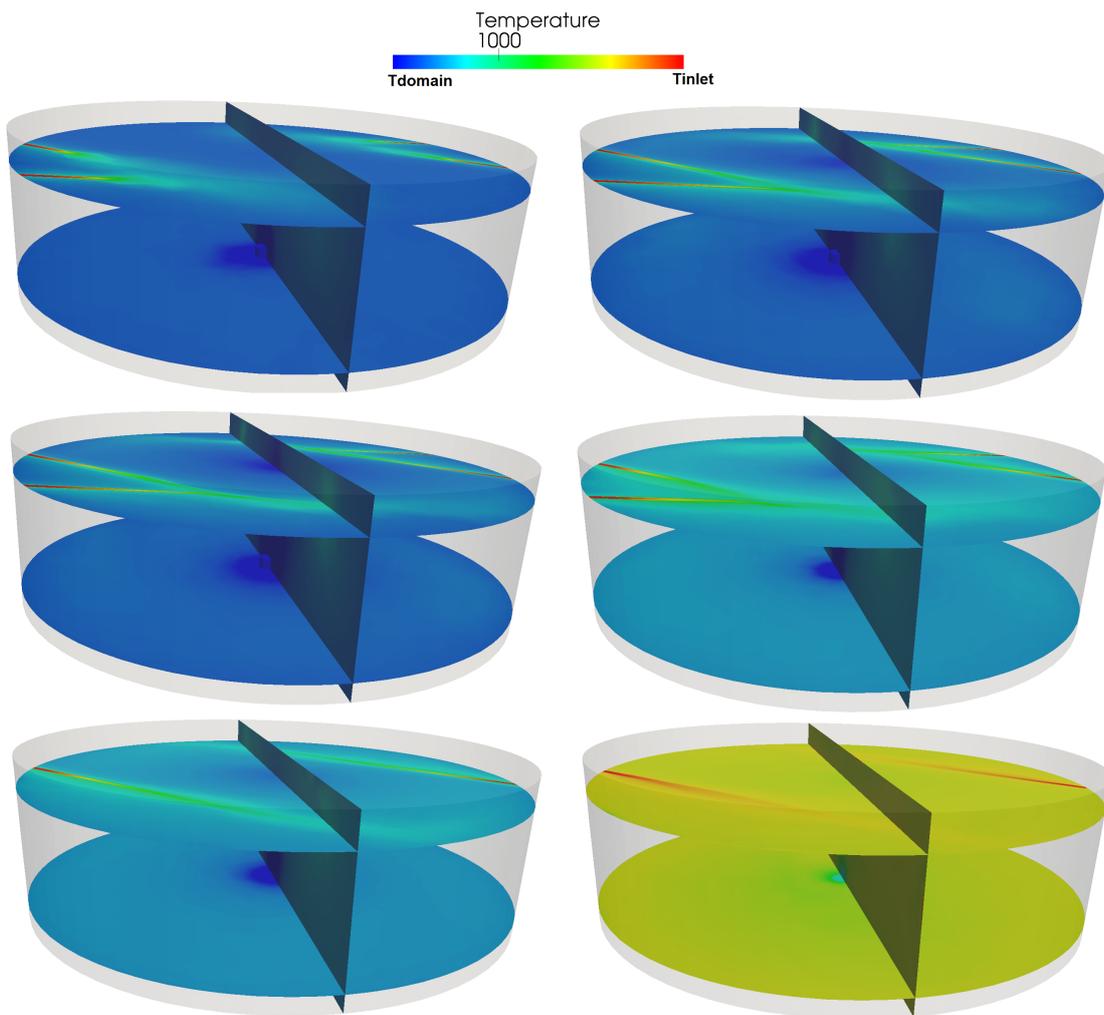


Figure 7.18 – Temperature distribution inside the furnace volume at different time instances.

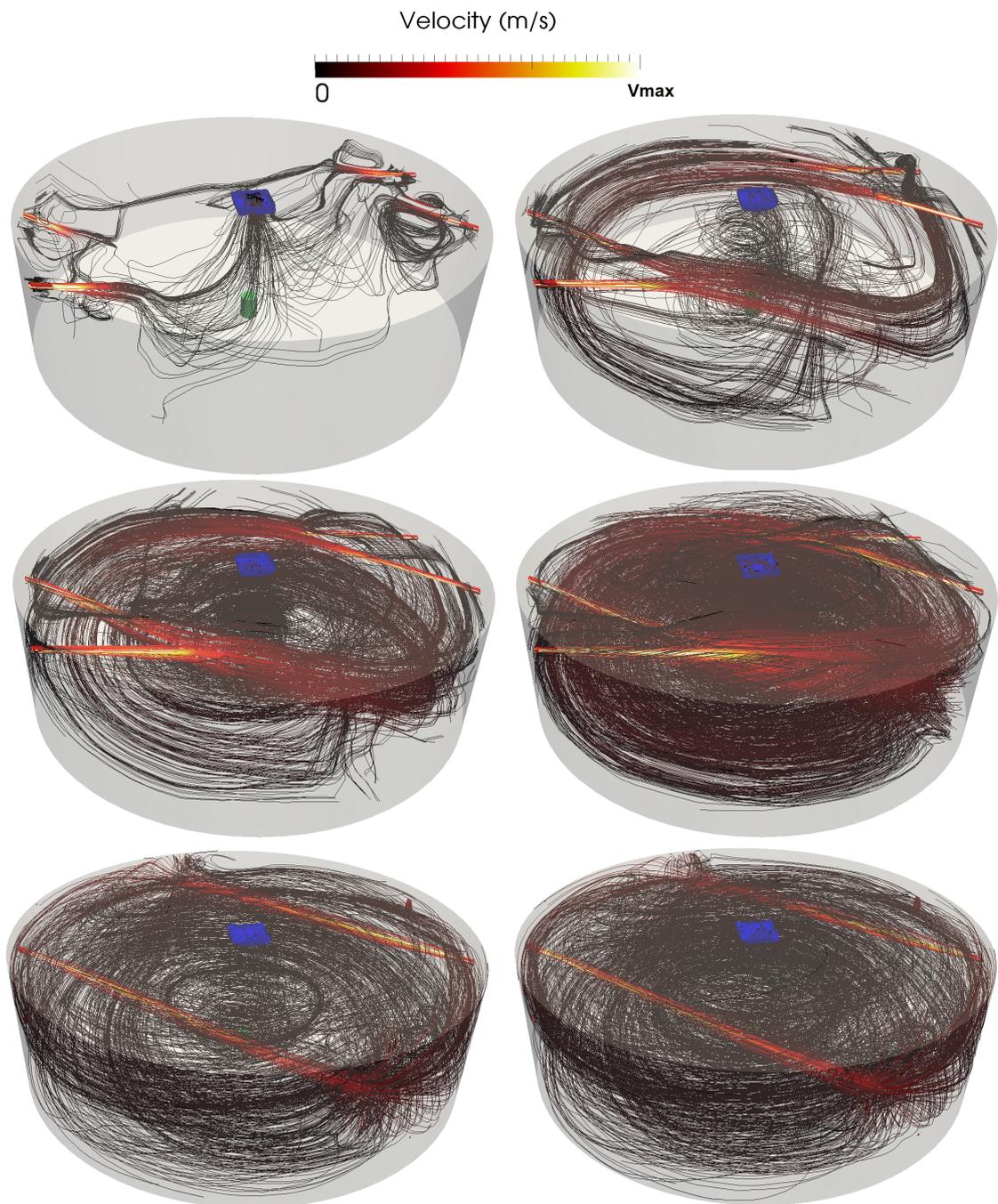


Figure 7.19 – Velocity streamlines inside the furnace volume at different time instances.

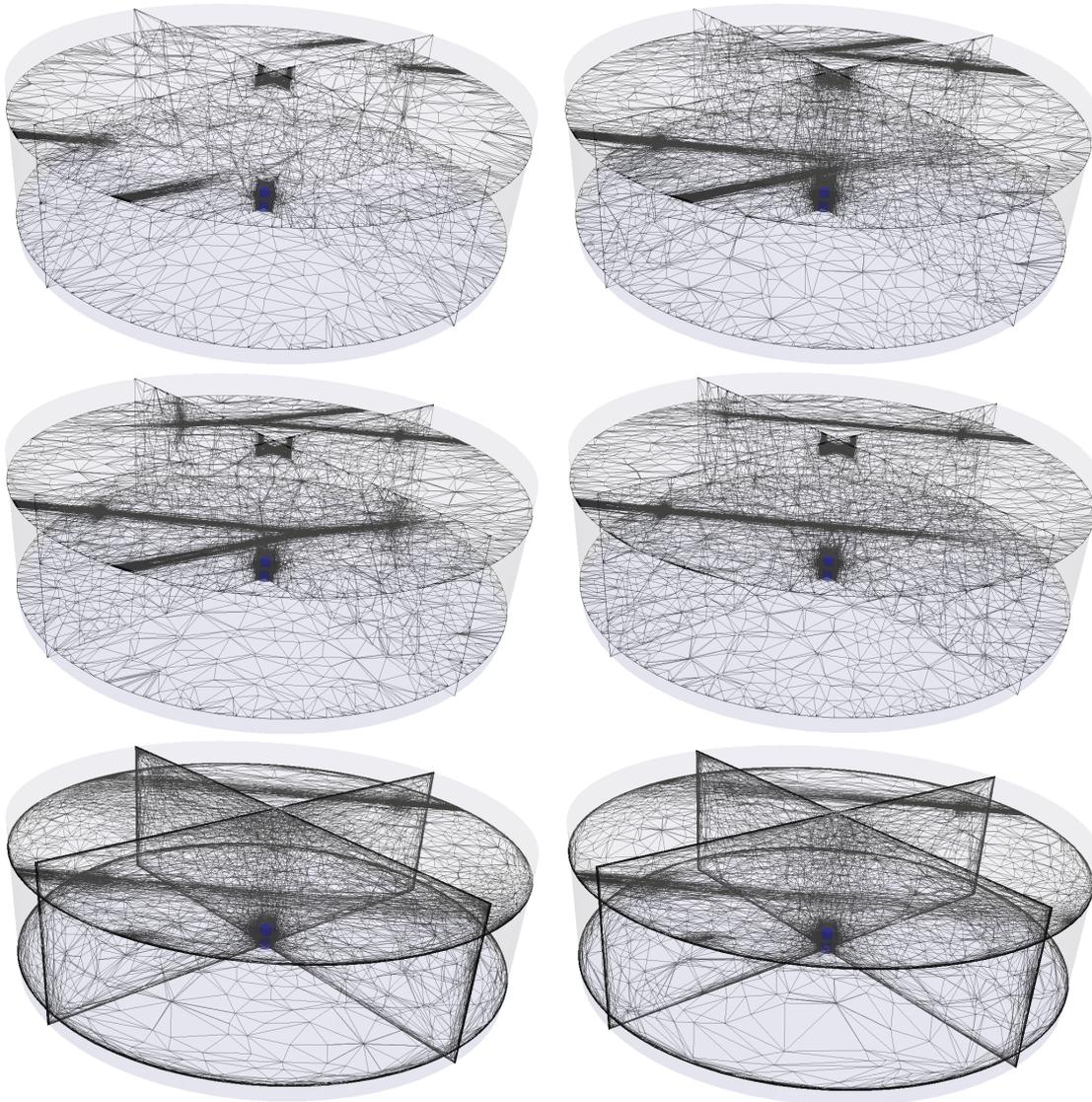


Figure 7.20 – Anisotropic meshes generated at different time instances.

localizing the available nodes in an optimal way to allow a good capture of this layer. At this level, we highlight the importance of adapting the mesh on the velocity field in order to detect and represent the emerging boundary layer. Then, as the temperature gets homogenized inside the cavity and the flow achieves a steady rotating motion, the conduction dominates the heat transfers at the ingot's level. Consequently, the mesh concentrates most of the elements in the workpiece ensuring the accurate capture of the thermal distribution inside it. While the elements inside the volume are highly elongated and very well oriented, a well refined boundary layer is identified and a high density of elements is spotted near and inside the ingot. Finally, since the inside of the furnace does not reach the temperature of the injected fluid, an important gradient is present and well captured at the flame's level. We note the complexity of generating a well adapted anisotropic meshing of the furnace's geometry as it presents at the same time curvatures and sharp edges. We point out that as the velocity field stabilizes inside the enclosure, the generated meshes converge to an optimal configuration and do not undertake significant improvements.

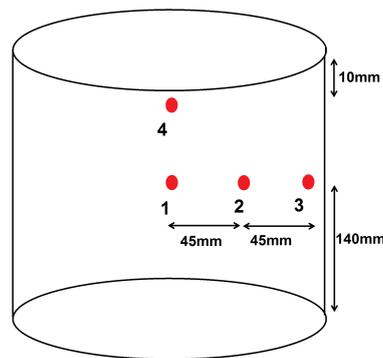


Figure 7.21 – Sensors' positions inside the cylindrical workpiece.

This application has been studied qualitatively from both experimental and numerical aspects. The industrial partner provided us with temperature data at different positions, inside the enclosure and the Inconel solid, over the simulation time to validate the accuracy of the developed numerical tools. The sensors locations inside the workpiece are sketched in figure 7.21 and the thermocouple inside the furnace volume is placed at the center of the planar cut having the same horizontal level as the centers of the burners. During the numerical investigation, we have run three simulations: the first on a fixed mesh with the same number of nodes (250,000) and a fixed time-step size 0.1s, the second using the anisotropically adapted mesh and a fixed time-step size 0.1s, and the third using the anisotropically adaptive space-time algorithm. Figures 7.22, 7.23 and 7.24 present the temperature evolution over time at the different sensors positioned inside the ingot and in the furnace chamber.

Comparing the numerical results to the experimental data, we observe that the temperature plots obtained using a fixed mesh and time-step size, present the same response to heating: a quick rise in temperature followed by a slowdown. However, the reached final temperature at the different sensors presents an error of around 48%.

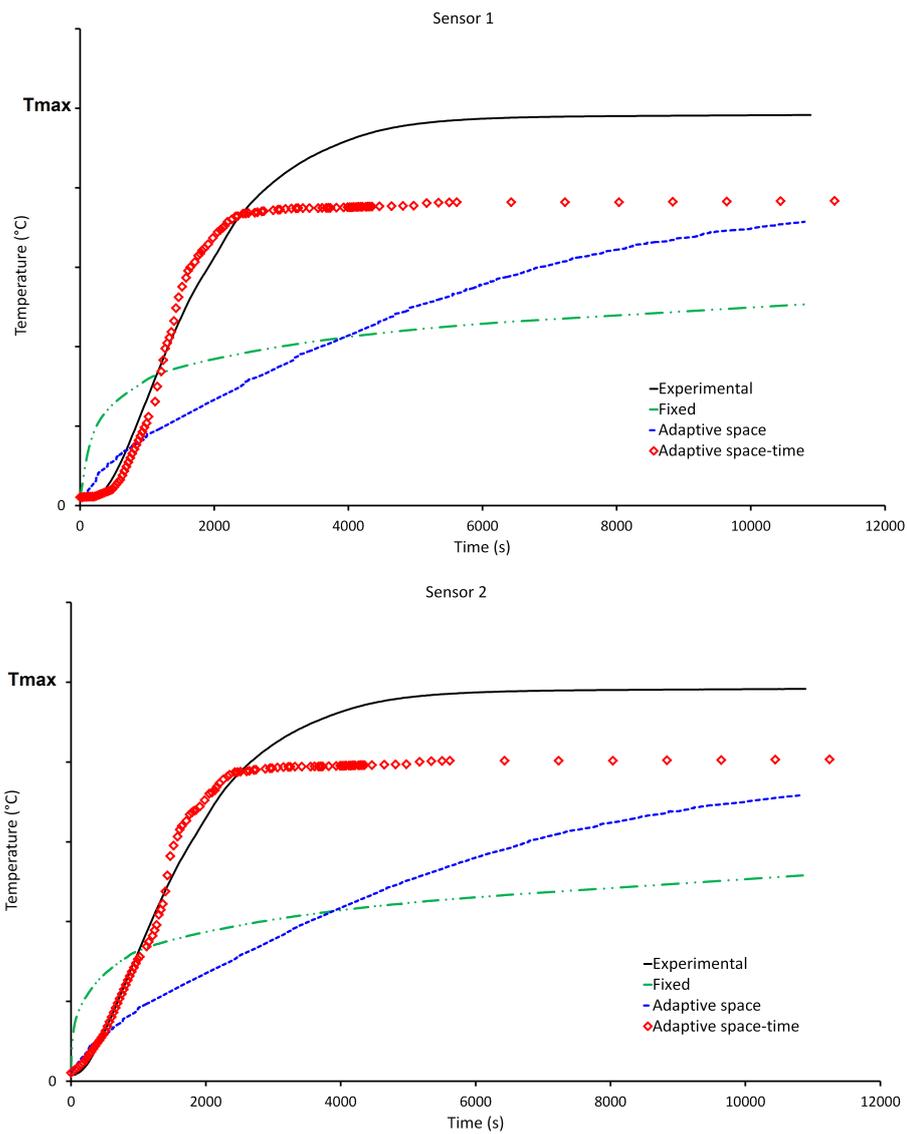


Figure 7.22 – Temperature distribution in the core of the workpiece.

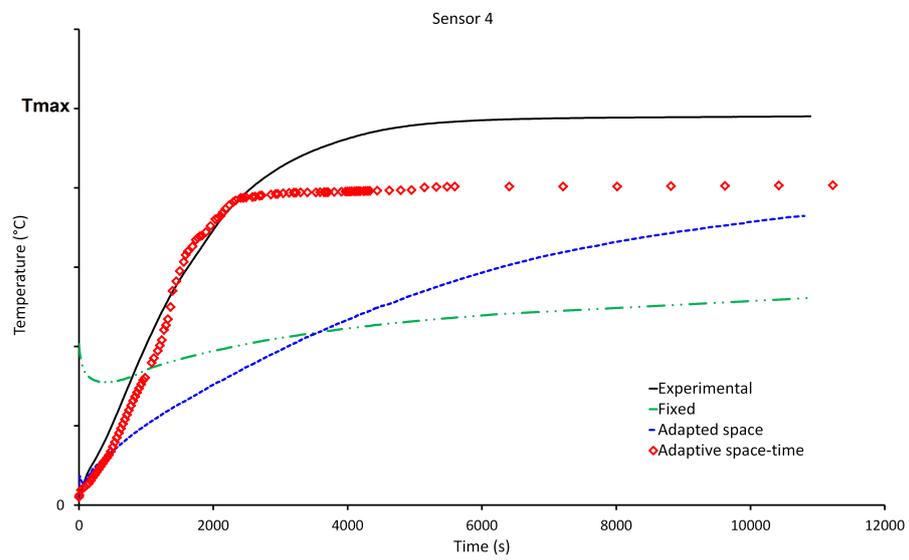
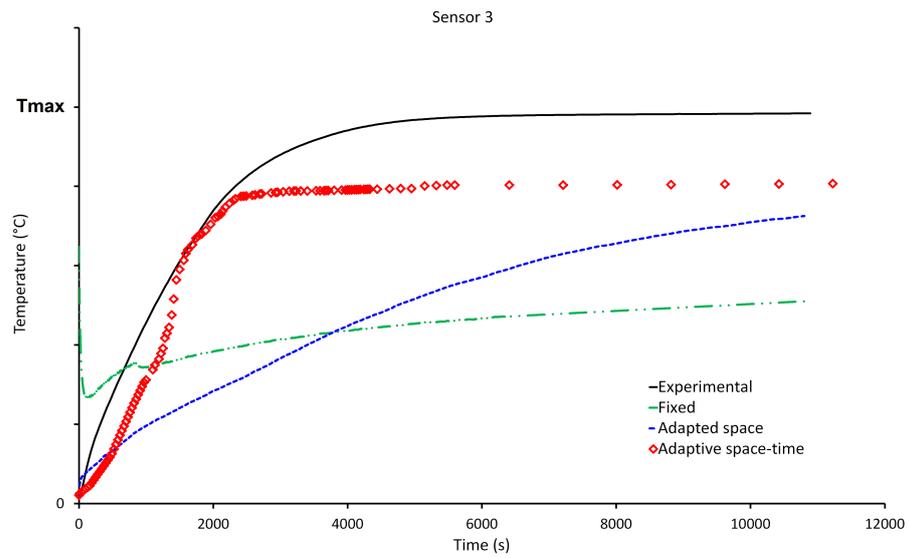


Figure 7.23 – Temperature distribution at the surface of the workpiece.

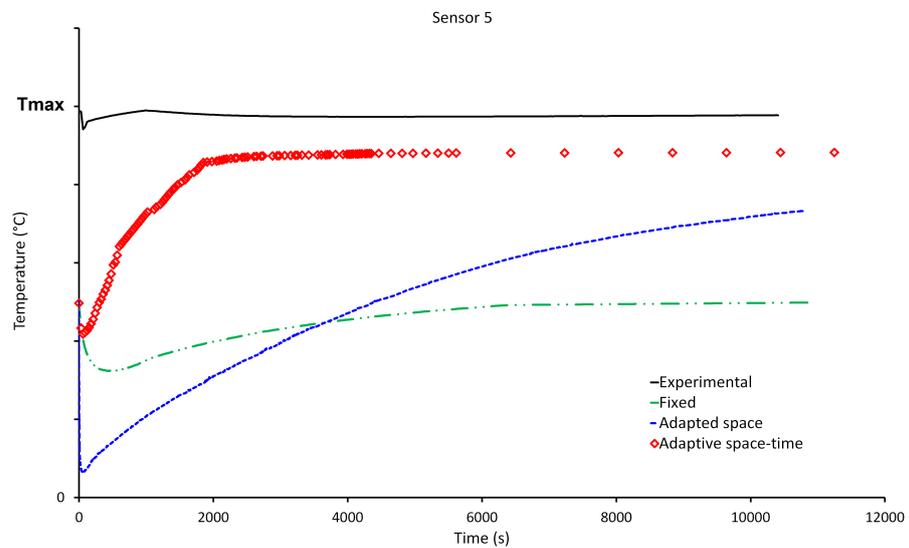


Figure 7.24 – Temperature distribution in the furnace volume.

The challenge posed on the new adaptation tools was to reduce this error while maintaining the same mesh resolution. As a first test, we have considered the same fixed time-step size and adapted anisotropically the mesh. The generated numerical temperature evolution at the different sensors reflected a diffusive nature. Indeed, inside the ingot instead of showing a rapid rise in temperature during the first 3,000s then a slowdown followed by a stagnation at a certain critical value, the temperature exhibits a gradual and smooth increase over time. Nevertheless, the achieved final temperature is closer to the experimental one and the error is around 27.3%. Thus, the anisotropically adapted test case provides a closer final temperature for the same mesh resolution on the different thermal sensors. Combining the anisotropic mesh adaptation with the time-adaptive algorithm, we observe a great improvement in the numerical solution, reducing the final error to 19%. Although the final experimental temperature at the heart of the solid geometry is still not reached, the numerical solution portrays well the profile of variation: a fast rise in temperature, followed by a gradual increase and a stagnation. Moreover, we can clearly detect a phase lag between the numerical and the experimental graphs. Several possible interpretations for this discrepancy can be attributed. First, the prescribed mesh resolution might not be enough to detect and follow the small scale details of the solution. Second, the thickness of the smoothed Heaviside function might not be small enough to render the sharp discontinuity in the material properties. A third reason is related to the inaccurate modelling of the initial setup of the problem in terms of boundary conditions, the approximated thermal heat flux at the walls, the approximation of the material and gas properties, *etc.* Moreover, the discrepancies can also be related to the diffusive aspect of the radiative model. Finally and most importantly, the failure to achieve the experimental temperature profile can be associated with the lack of a combustion modelling which is considered as a backbone component of heat exchanges and energy production inside industrial furnaces. Indeed, these geometries are called combustion chambers reflecting the role of combustion in stimulating and heating the induced thermal flow. We provide in figure 7.25 the thermal evolution at sensors 1 and 4. The temperature within the ingot evolves as expected with a faster heating on the surface which is in direct contact with the heated air.

As for the efficiency of computations, we summarize in table 7.1 the execution time required for each of the numerical tests. On the one hand, compared with a fixed mesh, an anisotropic mesh adaptation with a fixed time-step size reduces the computational cost of the simulation. Although the cumulative time for remeshing constitutes around 20% of the total CPU time, the solvers reflect faster convergence when the mesh is well prepared for the evolving solution, yielding faster computations. On the other hand, a more reduced execution time is noted when applying the time adaptation technique. This is in accordance with the tendency of the time adaptation algorithm to generate larger time-step sizes as depicted in figure 7.26.

Therefore, in this complex 3D conjugate heat transfer and fluid flow simulation

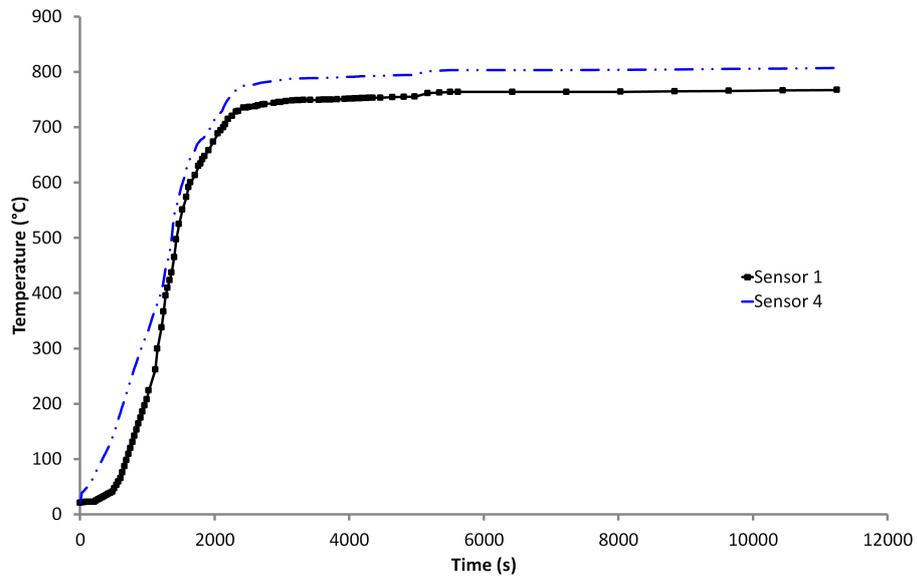


Figure 7.25 – Thermal evolution on the surface and at the heart of the workpiece.

|                                   | CPU Time (h) |
|-----------------------------------|--------------|
| Fixed mesh & $\Delta t = 0.1$     | 201          |
| Adaptive space & $\Delta t = 0.1$ | 185          |
| Adaptive space & time             | 167          |

Table 7.1 – CPU time needed to simulate the conjugate heat transfers and fluid flows inside an industrial furnace.

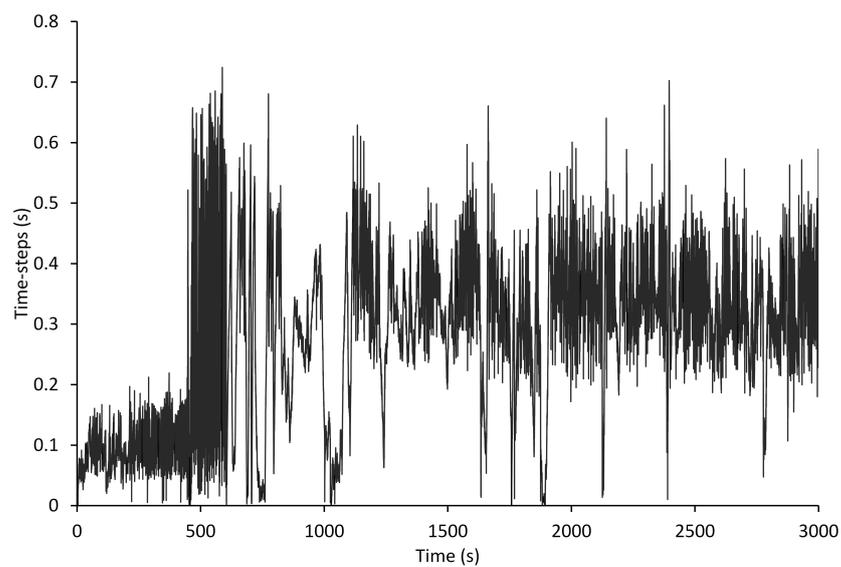


Figure 7.26 – Evolution of the time-step sizes.

inside a furnace volume, we have validated the role of the developed space and time adaptation tools in improving the accuracy and efficiency of the numerical solutions.

It is worth mentioning that the actual  $T_{\text{domain}}$  used in the experimental analysis was not the same as in the numerical setup for confidentiality reasons. The aim was mostly to improve, using the developed tools, the results obtained without adaptation.

## 7.5 3D COOLING INSIDE A QUENCHING CHAMBER

We are interested in this numerical example in studying the cooling of a steel work-piece inside a water-filled tank. This quenching process is known to induce a change in the metal structure and thermo-physical properties of the ingot. Therefore, an accurate modelling of the process is highly desirable in order to well predict the resulting final product. The simulation is divided into two phases: the pre-cooling whereby a flow is initiated inside the enclosure and the cooling phase where a hot solid is injected and is being cooled down by forced convection. The pre-cooling chamber's configuration is shown in figure 7.27 together with the generated initial mesh anisotropically adapted to the different components of the geometry. A cylindrical water injector, with surface area  $0.3m^2$ , is positioned at the center of the bottom wall and pumps water at temperature of  $21^\circ C$  and a speed of  $0.1m/s$ . The flow inside the enclosure is vented out through an cylindrical outlet, with radius  $0.2m$  and length  $0.3m$  located at the top of the octahedral walls. In order to provide a uniform cooling and prevent the accumulation of heat around the ingot during the cooling process, the periphery of the tank was instrumented with 8 cylindrical turbine agitators. While agitators 1 through 6 have a length of  $0.64m$ , a radius  $0.1845m$  and mixing speeds of  $0.294m/s$ , turbines 7 and 8, of length  $0.85m$  and radius  $0.29m$ , provide faster agitation of order  $0.618m/s$ . Before immersing the hollow cylinder steel object inside the quenching medium, the water is agitated for  $1100s$  to achieve its maximum capability as a liquid quenchant and provide an adequate cooling process. We present in figures 7.28 and 7.29 the velocity stream-

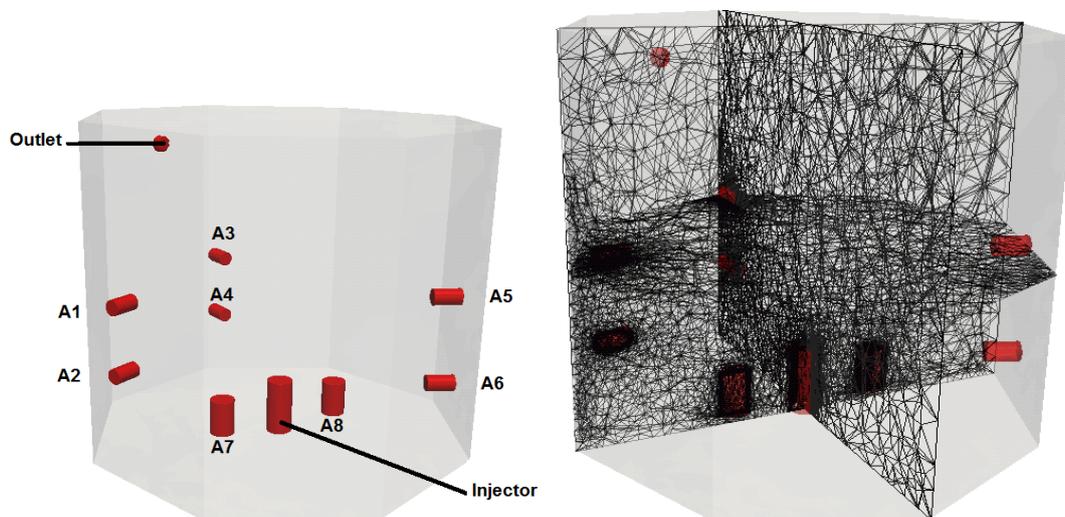


Figure 7.27 – Initial configuration and mesh of the quenching chamber for the pre-cooling phase.

lines' evolution over the pre-cooling phase. It shows the spread of the flow inside the chamber and the developing chaotic behavior. We can detect how the flow is injected at high speed and hits the opposite wall forming a boundary layer and decomposing into smaller vortices that are at the origin of the dynamic recirculating motion inside the cavity. A good capture is rendered by the use of the anisotropic mesh adaptation tools, where the adaptation is performed based on the velocity field and norm and the levelset functions associated with the turbines, the injector and the outlet. We can detect in figure 7.30 the sharp representation of the emerging small and large eddies and boundary layers. By the time 1100s, the hollow cylinder of height 4.065m, outer

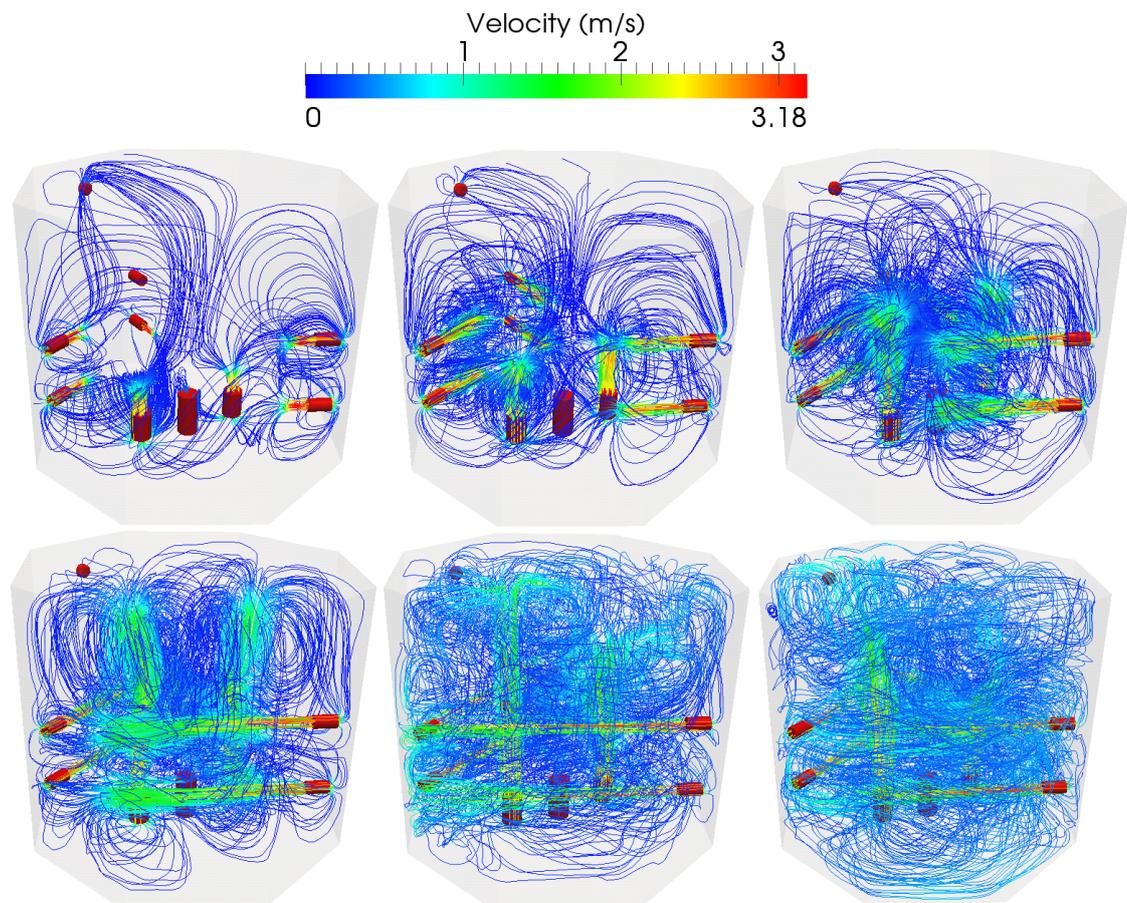


Figure 7.28 – Velocity streamlines during the pre-cooling phase inside the quenching chamber at different time instances.

radius 2.605m, and an inner radius 2.2425m, taken initially at 980°C, is inserted into the water tank and positioned on the ground, creating a strong temperature gradient along the fluid/solid interface. This gradient can numerically be at the origin of a thermal shock and numerical instabilities if the numerical method is not well prepared to treat these cases. We recall that, in the numerical simulation, we use the shock capturing Petrov-Galerkin stabilized finite element method to deal with this kind of problems. No slip boundary conditions are imposed on the walls. The thermal exchange with the external medium is ensured by enforcing a Fourier type heat flux at the top wall and a constant flux at the other walls. The developed space and time adaptive tools are

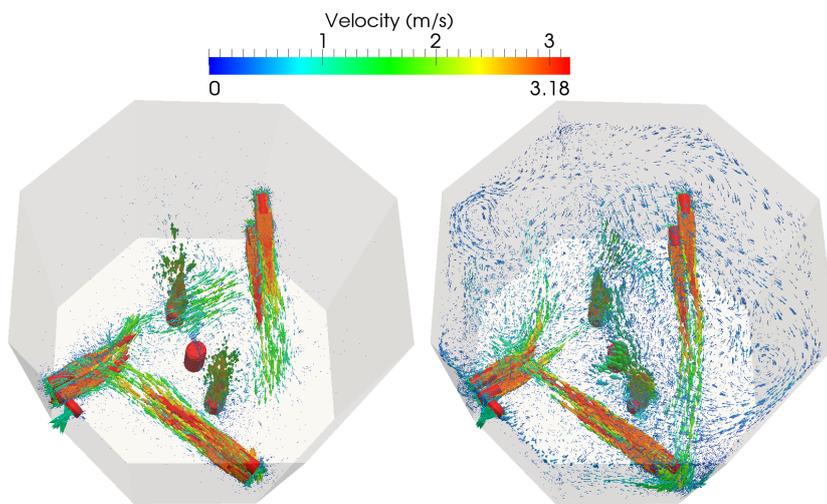


Figure 7.29 – A top view on the velocity field inside the quenching chamber during the pre-cooling phase at two different time instances.

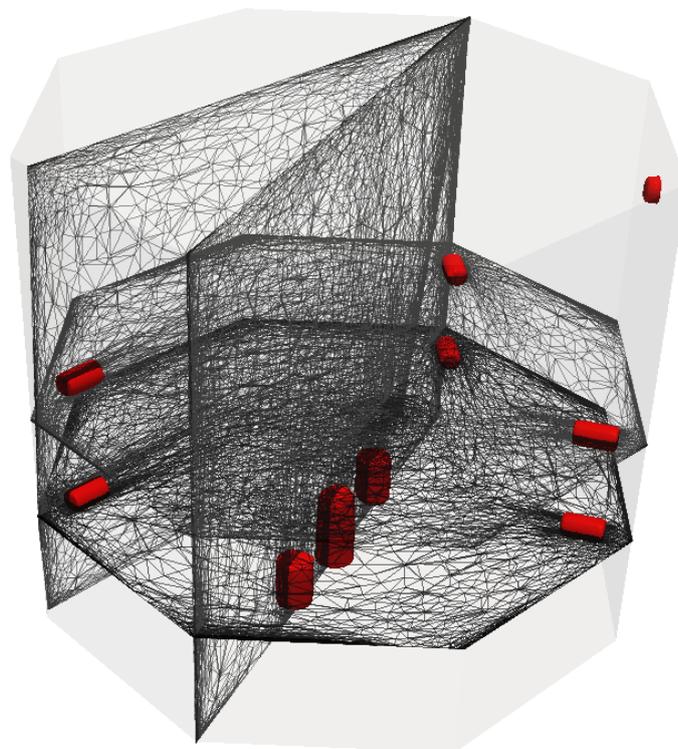


Figure 7.30 – Snapshot on the anisotropic mesh associated with the quenching chamber at time 1100s.

applied at a frequency of 10 time increments using 100,000 nodes while adapting on the levelset functions of the agitators, the injector, the outlet and the hollow metal as well as the temperature field and the velocity field and norm.

We present in figures 7.31 and 7.32 the velocity field and streamlines' distribution after immersing the heated workpiece at increasing time instances. We can identify the toroidal aspect of the flow. It is characterized by uniform internal and external circular rotating behaviors around the hollow cylinder. This velocity distribution allows homogenous heat transfer between the cold water pumped into the enclosure and the steel ingot. The thermal distribution over the simulation time on different planar cuts

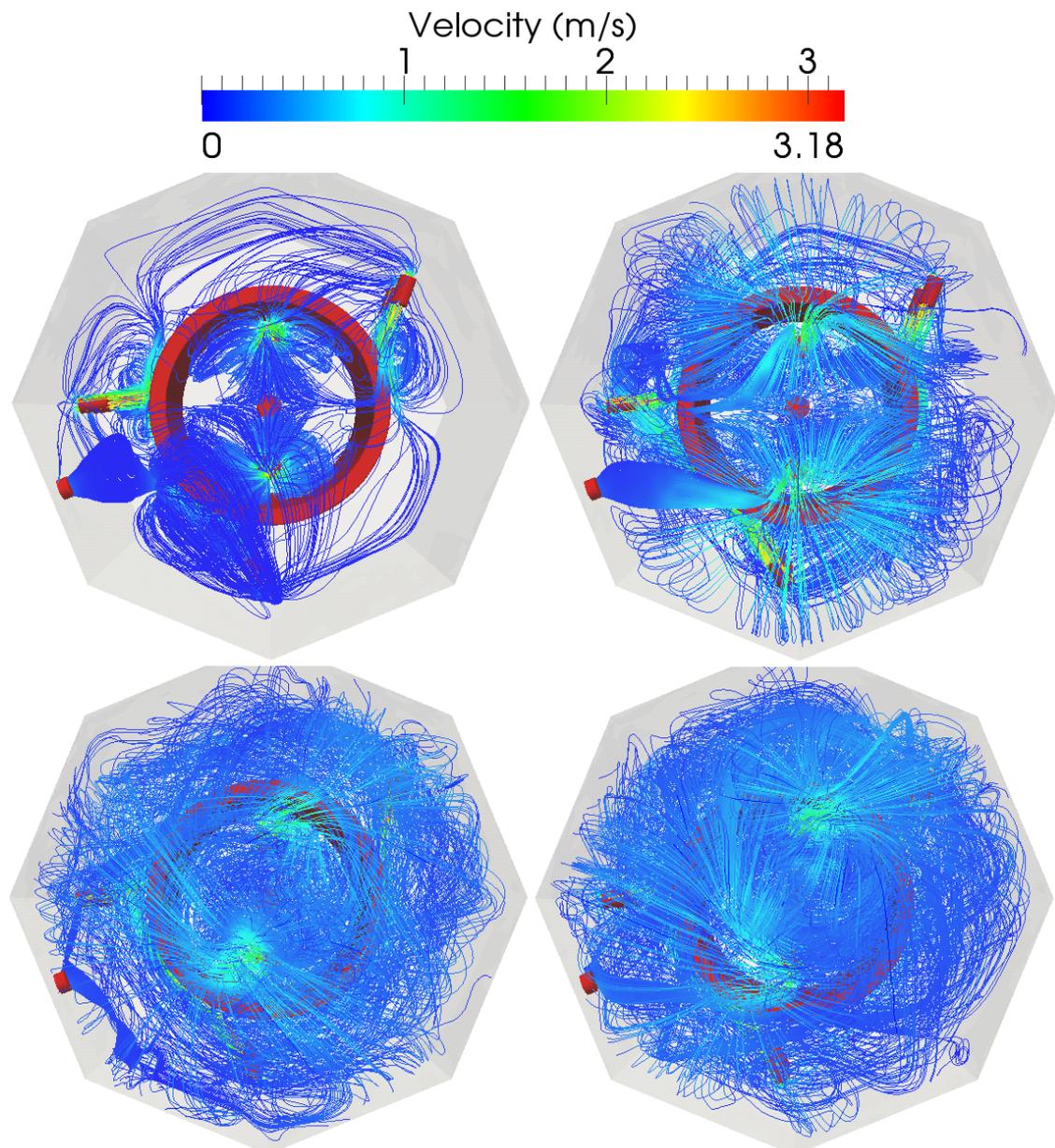


Figure 7.31 – Velocity streamlines during the cooling phase inside the quenching chamber at different time instances.

are depicted in figure 7.33. The snapshots reveal the uniform and symmetric cooling

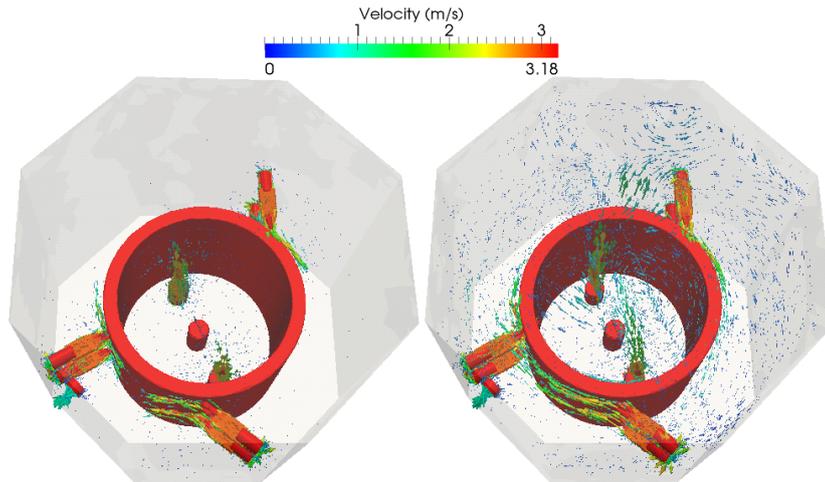


Figure 7.32 – Velocity field inside the quenching chamber during the cooling phase at two different time instances.

of the metal which is at the origin of a uniform change in the thermo-mechanical properties of the final product in terms of resistance, ductility and flexibility. We point out that the obtained solution is free of numerical oscillations reflecting the good tuning of the adopted stabilization approach. The anisotropic mesh generated while adapting on the hollow object, the injector, the turbines, the temperature field and the velocity field and norm is provided in figure 7.34. We can clearly identify how accurately the solid/fluid interface is represented yielding a good material distribution and thus accurate heat transfers over the computational domain. The nodes are optimally distributed in accordance with the physics of the problem providing a good control on the interpolation errors of the solution. During the experimental analysis, the hollow ingot was instrumented with thermocouples and thermal variations were reported for numerical comparisons. We present in figures 7.35 and 7.36 the thermal distribution at different sensors positioned inside and outside the workpiece. The variations obtained using the anisotropic mesh adaptation and time adaptive approaches are compared to the ones generated on a fixed mesh and using a fixed time-step size 0.1s and to the experimental data. It can be clearly identified that the experimental data are way from being well captured and that numerically a faster cool down is being predicted. This discrepancy can be associated with the absence of a boiling model as well as the diffusive character of the used  $P_1$  radiative model. Indeed, water, having a higher thermal conductivity than vapor, provides faster cooling rates. Moreover, since the specific heat capacity associated with water is higher than that of vapor, the former is able to absorb more heat than the latter for the same raise in their temperatures. Therefore, by ignoring the presence of vapor films in the surrounding of the hollow ingot and by assuming that this solid is in direct contact with the water, that presents higher quenching rates, a rapid decrease in its temperature is expected. This is exactly what was reflected by the numerical simulations. Furthermore, comparing the results obtained with the anisotropic mesh adaptation and time adaptive approach we detect a better capture of the thermal

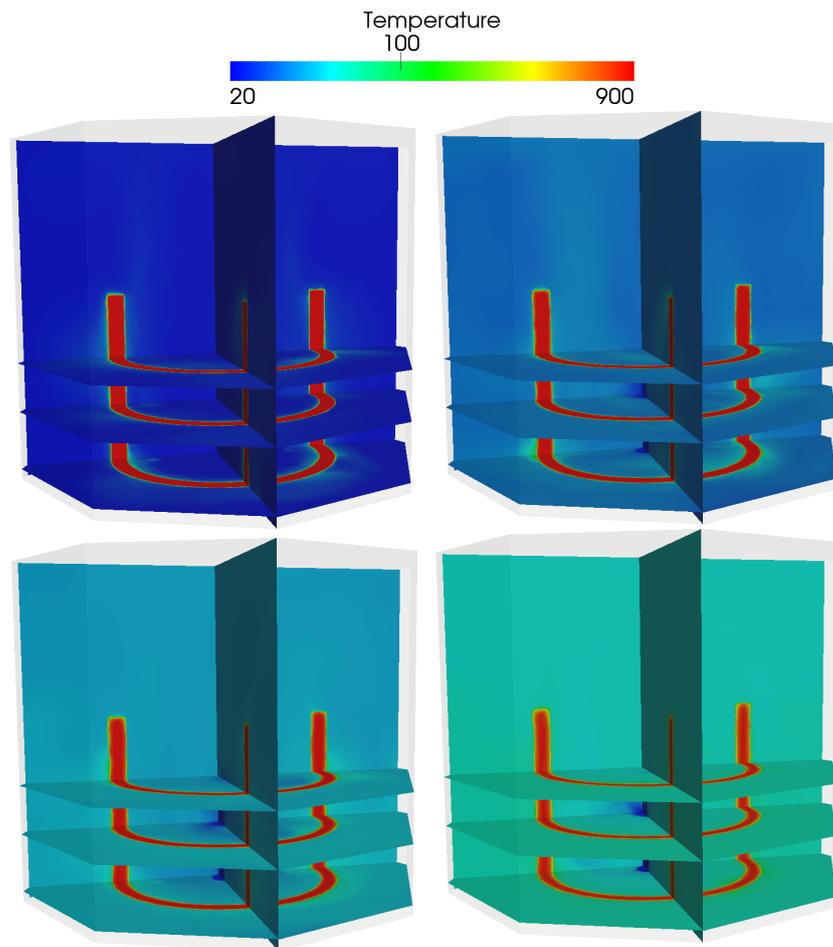


Figure 7.33 – Temperature distribution inside the quenching chamber during the cooling phase at two different time instances.

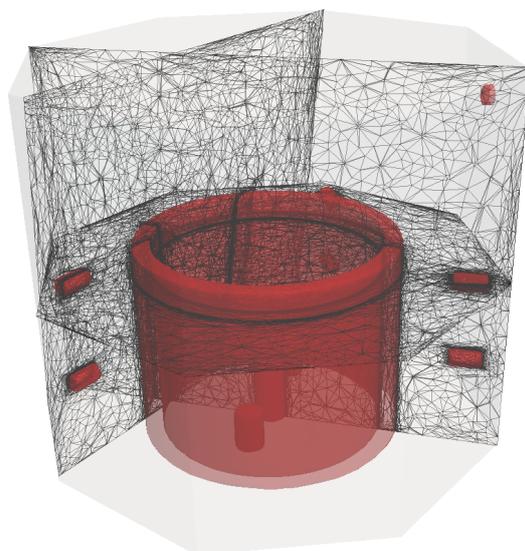


Figure 7.34 – Snapshot on the anisotropic mesh associated with the quenching chamber after immersing the hollow solid.

evolution on all the sensors. This proves once again that the developed adaptation tools improve the accuracy of the numerical solutions. Figure 7.37 presents a comparison

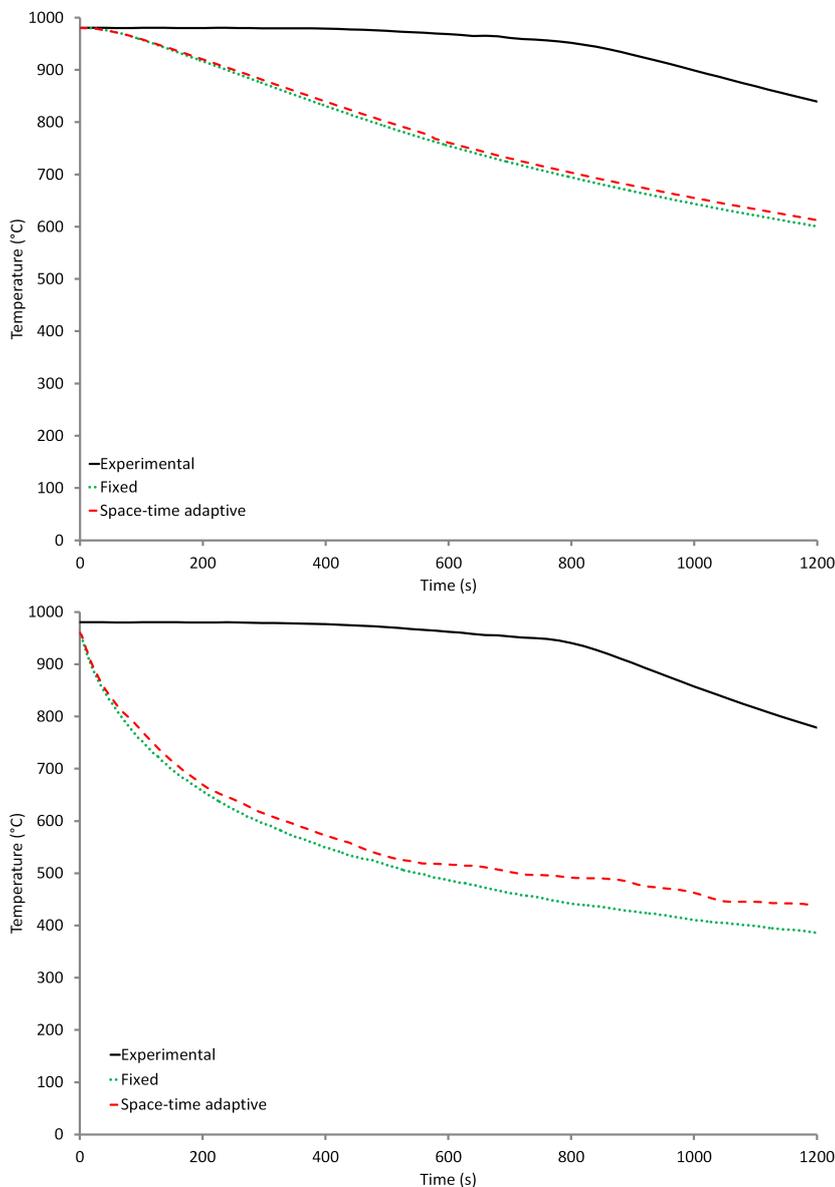


Figure 7.35 – Temperature variation at thermocouples C1 and C3 implanted on the surface of the workpiece.

of the thermal distribution on the different sensors implanted inside the workpiece. The obtained variations are in very good agreement with expectations while a very fast cooling is noted at sensor 3 that is the closest to the cold water injector, a slower rate is highlighted in the core of the workpiece. In addition, due to the heat transfer between the hot metal and the cold fluid, as the former gets cooler the latter gets hotter and hence presents lower cooling rates. This is revealed by the slow down of the cooling at sensor 3 after 200s. Furthermore, being close to the surface, thermocouples C1 and C5 reflect a quicker cooling than thermocouples C2 and C4 which are at the heart of the solid object. In the absence of the heat transfer coefficient, the IVM based on the new adaptive algorithm proves, once again, to be essential for the resolution of such a

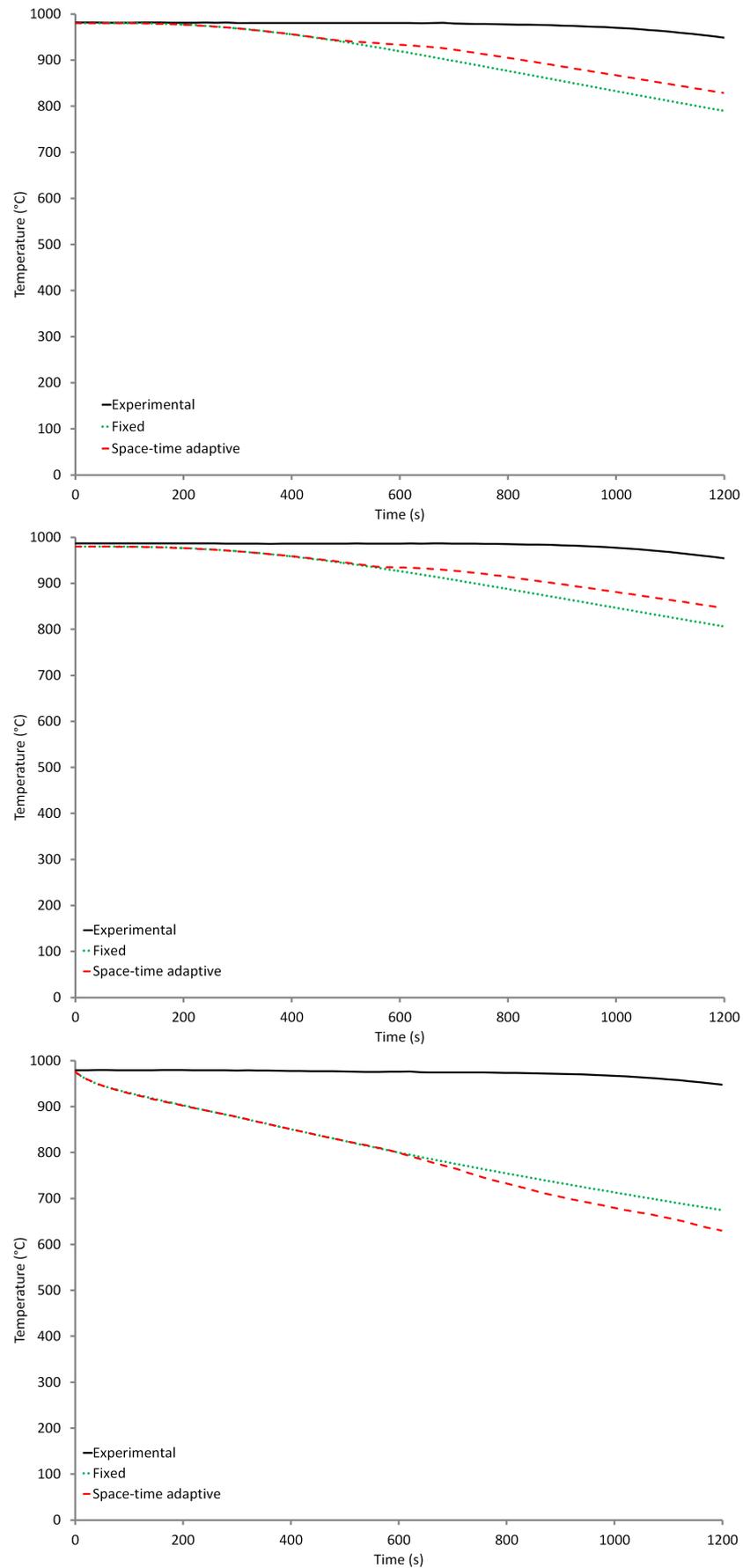


Figure 7.36 – Temperature variation at thermocouples C2, C4 and C5 implanted at the heart of the workpiece.

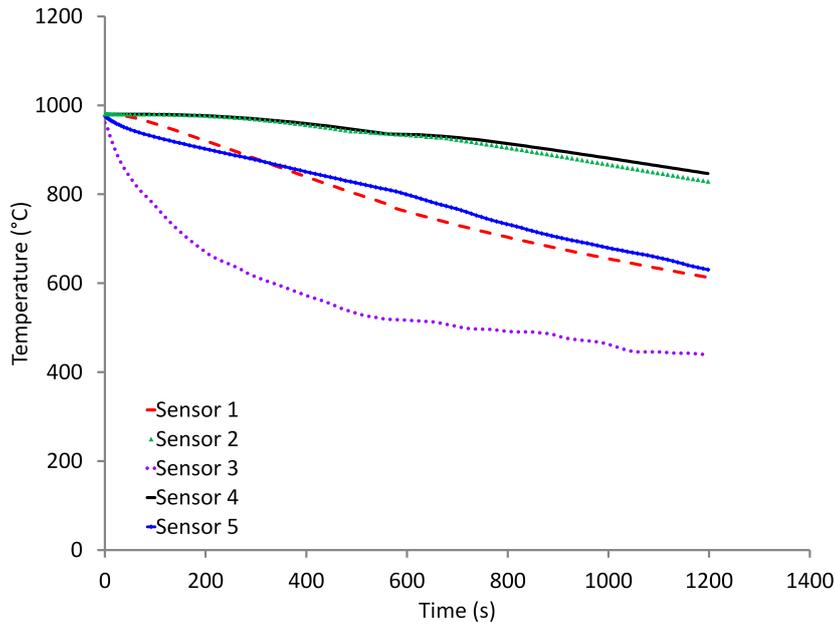


Figure 7.37 – Comparison of the thermal distribution at the different thermocouples implanted inside the workpiece.

complex conjugate heat transfer fluid/structure interaction problem. In order to quantify the accuracy of the computed solution, we compare the obtained results using the anisotropic mesh adaptation and the time adaptive method with the ones computed on a fixed mesh of around 100,000 nodes and a fixed time-step size 0.01s and the ones computed on a very fine mesh of 875,000 nodes and a time-step 0.005s. We can clearly see that the solution generated on an anisotropically adapted mesh is more accurate than the one computed on a fixed mesh having the same resolution.

As for the efficiency of computations, we summarize in table 7.2 the CPU time needed for generating each of the computed solutions. The 3D computations were run in parallel on 40, 2.4Ghz Opetron cores linked by an Infiniband network. Although using the anisotropic mesh adaptation tools induces an associated execution time, a faster convergence of the finite element solvers is noted when the meshes and time-step sizes are optimally predicted to control the interpolation error in the solution. The obtained data reflect the great advantage of the developed approach in terms of simulation time.

We note that in this test case the prediction of the boiling process is very important for determining the thermal variation and heat transfer at the fluid/solid interface. Indeed, during the cooling process, a vapor film is formed around the metal with quenching properties weaker than those of water and hence if taken into account they would result in a numerical solution that is closer to the experimental one. This modelling is not taken into account in the current study and will be the subject of future investigations. This explains the discrepancies between the numerical results and the experimental data.

|                                                 | CPU Time (h) |
|-------------------------------------------------|--------------|
| Fixed mesh 875,000 nodes & $\Delta t = 0.005$   | 458.6        |
| Fixed mesh 100,000 nodes & $\Delta t = 0.1$     | 134.8        |
| Adaptive space 100,000 nodes & $\Delta t = 0.1$ | 164.3        |
| Adaptive space & time 100,000 nodes             | 2.9          |

Table 7.2 – CPU time needed to simulate the conjugate heat transfers and fluid flows inside a quenching chamber.

## 7.6 RÉSUMÉ FRANÇAIS

Dans ce chapitre, nous nous sommes intéressé à l'application des outils développés et des méthodes numériques présentées dans les chapitres précédents à la modélisation et la simulation de procédés industriels tels que le traitement thermique de pièces métalliques dans des fours et la trempe dans des bûches ou des chambres de refroidissement. La complexité du procédé, des géométries, les ratio de propriétés physiques élevées rendent une telle simulation difficile en terme de modélisation et de coût de calcul. La compréhension des phénomènes physiques impliqués permettra à nos partenaires industriels de contrôler les séquences de traitement thermique en vue de réduire la consommation d'énergie et d'améliorer la qualité de leurs produits en termes de microstructure, flexibilité, ductilité et résistance.

Trois classes différentes de procédés industriels ont été traitées: refroidissement 3D par convection naturelle d'un lingot en forme de disque chapeau, chauffage 3D d'une pièce métallique à l'intérieur d'un four industriel, trempe 2D d'un lingot en forme de couronne par convection forcée de gaz et un cas de trempe 3D dans une bûche à eau. Il est important de signaler que ce travail constitue la première fois que de véritables simulations ont été abordées, avec un coût de calcul raisonnable, modélisant les procédés dans leur intégralité ce qui se pose comme une avancée majeure au vu de la complexité des cas considérés.

Dans tous les cas, en comparaison avec les profils expérimentaux, la tendance des solutions numériques obtenues reflète le potentiel des outils numériques en termes de méthodes de stabilisation et d'adaptation espace et temps. Bien que les données expérimentales n'ont pas pu être récupéré numériquement pour tous les exemples, les outils développés ont démontré leur efficacité et utilité à améliorer la qualité des solutions cherchées. La confrontation avec les données expérimentales a souligné la nécessité de développer des modèles numériques supplémentaire, tel que des modèles de combustion, d'ébullition, et des modèles de rayonnement moins diffusifs afin de mieux représenter les procédés envisagés.



# CONCLUSION AND OUTLOOK

"Seuls se félicitent d'être arrivés ceux qui se savent incapables d'aller plus loin."

Amin Maalouf

This thesis was motivated by the need to perform long time and large-scale simulations of complex fluid/structure interaction problems. These problems can exhibit challenges at the interface levels and variations at the small and the large scales, and in both cases it is highly desirable to capture those variations with good levels of accuracy and efficiency. To answer this objective, we aimed at developing a general space and time adaptation framework. The adopted approach responds to the anisotropic nature of the physical phenomena by constructing anisotropically adapted meshes that control the global interpolation error induced during the computations. The present work can be regarded as a sedimentation and cultivation plan in a fertile land.

We started in chapter 1 with the land preparation process where we introduced the numerical tools adopted for the modelling and simulation of conjugate heat transfer and turbulent flow problems. Since, on the one hand, standard Galerkin formulations exhibit numerical oscillations in convection or diffusion dominated regimes especially in the vicinity of sharp solution gradient, and, on the other hand, incompatibility restrictions on the choice of the finite element spaces are necessary for flow problems to ensure the stability of the formulation, we resort to the use of stabilization techniques in order to deal with numerical instabilities. In order to deal with the turbulent behavior of the flow, turbulence models were considered. The performance and potential of these numerical methods have been exposed on two and three dimensional applications reflecting their accuracy in simulating physical phenomena. In all the examples, a fixed fine mesh and a fixed small time-step size have been used. Although these tools resulted in precise computations, the investigation on complex three dimensional coupled problems reflected the need for further considerations tackling the computational efficiency to cope with the limitation in the available computer resources.

The novelty brought forth in this work resides in the development of space and time adaptive approaches capable of improving the accuracy and efficiency of the computations while respecting the available computational power. An anisotropic mesh adaptation technique, working under the constraint of a prescribed fixed number of nodes in the mesh has been developed, implemented and validated in chapter 2. The mesh generation process has been briefly described pointing out its capability to receive metric tensors and generate the corresponding adapted mesh. A detailed description of the adaptation framework has been provided. The mesh adaptation problem was reformulated as an optimization one whereby one seeks, at each node in the mesh, the metric tensor that best controls the interpolation error in the  $L^p$  norm. The tensor, at a certain node, prescribes optimal mesh sizes along the different directions connected to the node. Moreover, a natural way to obtain the latter direction was defined using

the length distribution tensor and the optimal mesh sizes were obtained by imposing stretching factors along the existing edges in the mesh. Two approaches for computing the stretching factors while respecting a fixed mesh complexity were proposed. The first one relies on the minimization of the total error over the edges in the mesh whereas the second intends to equi-distribute the error along these edges. In order to quantify this error, a new edge-based a posteriori error estimator was derived. It defines the error along a spatial edge as the projected change in the field's gradient along that edge. Furthermore, since for most physical problems the exact solution's gradient is not known analytically then a recovery process has been proposed through the resolution of a least square problem and resulted in a second order reconstruction. The theoretical and numerical validation on the control of the  $L^p$  norm of the interpolation error through the use of this edge-based error estimator have been provided. Further consideration of the metric derivation lead to the extension of the definition of the averaged length distribution tensor to a privileged distribution giving more weight to the edges in the gradient direction and yielding a faster and sharper representation of anisotropic features and a capture of small scale variations. The global accuracy and efficiency of the derived method were validated on analytical functions exhibiting sharp gradients, multiscale variations and steep discontinuities. In all the cases, the adaptation algorithm proved to improve the quality of the results and to maintain a reduced execution time.

In chapter 3, we aimed at testing the developed anisotropic mesh adaptation tool on CFD problems. We started with an a priori error analysis demonstrating that the control of the interpolation error on the convection-diffusion-reaction and the Navier-Stokes equation yields a control of the approximation error of these problems. Then definitions of the characteristic lengths involved in the solvers' stabilization parameters were provided accounting for highly stretched elements. Applications of the anisotropic mesh adaptation technique on CFD problems, provided next, reflected the capabilities of the developed approach to enhance the precision of computations and reduce the simulation time.

The theory of adaptation was then extended in chapter 4 to include a time-adaptive technique. The coupling between the spatial and temporal adaptations was respected using the equi-distribution principle. The convergence, accuracy and efficiency of the newly developed adaptation method, known as the classical approach or the single time-step adaptation, have been validated on two and three dimensional problems involving conjugate heat transfer and turbulent flows.

In the view of resolving fluid structure interaction problems, we have adopted in chapter 5 the immersed volume method and extended it to contain dynamic and automatic mesh adaptation. A full description of the method's components was emphasized. The method retains the use of a monolithic formulation whereby the fluid and solid parts of the problem were considered as a single medium with variable material properties. A levelset function was used to delimit the interface between the subdomains and to distribute the material properties using mixing laws. Combined with the

dynamic anisotropic mesh adaptation technique, the IVM lead to accurate simulation of FSI problems. By virtue of the monolithic formulation, the method does not need to impose a heat transfer coefficient at the interface level. This parameter is usually difficult to determine and requires experimental studies and hence constitutes a limiting factor for industrial applications. In this chapter, we have also described the simple and powerful multi-criteria adaptation that permits the generation of meshes adapted on several fields of interest. The numerical results on turbulent fluid flows and conjugate heat transfer problems highlighted the potential of the IVM and the classical adaptation technique to handle fluid-structure interaction problems with good levels of accuracy and efficiency. Nevertheless, when applied with a certain frequency in the context of problems involving moving geometries, the anisotropic adaptation showed to be slowing down the advancement of the objects over time raising the need for an adaptation that deals with slabs of time.

To overcome this computational difficulty and ensure full adaptation over the whole simulation time, the spatio-temporal anisotropic adaptation has been extended in chapter 6 to construct optimal meshes and time-step sizes for slabs of time. All the details on the derivation, structure and implementation of the method were discussed. The algorithm involves the generation of a set of optimal metric tensors and slab sizes followed by a 2D or a 3D remeshing in space and a 1D remeshing in time. Notably, the 1D remesher was developed and implemented in this work permitting the redistribution of the temporal nodes in an optimal configuration. The algorithm demonstrated its accuracy and efficiency on analytical functions, CFD application, and a fluid-structure interaction problem involving moving geometries.

The application of the developed numerical tools on 3D complex problems involving conjugate heat transfers and turbulent flows inside large scale geometries have been investigated in chapter 7. The obtained results reflected an improvement in the numerical accuracy and the efficiency of computations compared to the ones computed on a fixed fine mesh and a fixed small time-step size.

Therefore, we have prepared and sedimented the fertile land with nutrients and cultivated some of its fruits. At this stage, it is ready to receive new implements from successive endeavors. There is a series of short and long term prospects of this work. These can be divided into three main parts. The first one concerns the improvement of the existing solvers, the second deals with the adaptation algorithm and the third concerns the industrial applications.

- At the level of the solvers, we have used a  $P_1$  radiation model which is known to overestimate the radiative effects and to equally diffuse the temperature to the neighboring medium. The work has already started on the development of a more accurate radiation model, the  $M_1$  model, in the work of [Schmidt 16]. This model accounts for the radiative directions of propagation. In addition, to improve the accuracy of the computations, a first attempt would be to enrich the variational multiscale stabilization approach to account for dynamic variations of

the small scales and hence model the nonlinear effect of turbulence on the global flow. The second goal would be to extend the available numerical tools to deal with compressible flows. A third objective is to extend the implemented stabilized finite elements methods to account for higher order temporal discretization.

- The enhancement of the developed space and time adaptation techniques can be achieved by providing a control on the approximation error instead of the interpolation one. A second improvement would be to provide a problem oriented control instead of a field based adaptation. An extension to build up anisotropic meshes with a desired resolution at the boundary layers is currently under investigation in the thesis work of [Billon 16] and will in turn open the door to further considerations concerning the accuracy provided by the adaptation process.
- Concerning the industrial applications in the frame of the REALisTIC project, combustion and boiling models seem mandatory to reduce the discrepancies between the numerical solutions and the experimental data. This would permit the investigation of what-if scenarios and answer the general industrial objective of reducing pollutant emissions and energy consumption as well as improving the quality of the final product. Moreover, more accurate experimental data are highly demanded for conducting real simulations, these include the thermo-mechanical properties of the insulating walls, the chemical constituents and thus the combustion power of the injected gas. Until now, adhoc conditions were provided by our industrial partners based on observations and estimations. However, for accurate representation of the conjugate heat transfers, qualitative information are still needed. To do so, the industrials are currently setting up heating and quenching chambers that would only serve for experimental studies and thus for numerical validations.

On the long run, these developments can serve as the basis for building design optimization algorithms for aerodynamic simulations, automotive industry, heat treatment furnaces and many other applications.

## RÉSUMÉ FRANÇAIS

Dans cette thèse, de nouvelles méthodes d'adaptations anisotropes en espace et en temps ont été proposées. Une approche de construction de métriques basée sur un estimateur d'erreur par arrête a été développée. Le principe d'équidistribution de l'erreur en espace et en temps a permis le contrôle de l'erreur d'interpolation spatiale et temporelle avec un seul paramètre: le nombre fixe de nœuds dans le maillage. Cette restriction se pose comme un avantage essentiel de la méthode proposée et permet de réduire le temps de calcul. Les outils numériques proposés: solveurs éléments finis stabilisés, méthode d'immersion de volume, et adaptation anisotrope en espace-temps ont révélé une robustesse et une capacité à modéliser et simuler des problèmes fortement couplés d'écoulement de fluides turbulents et de transfert thermiques. Leur performance

a été confronté aux résultats obtenue sans adaptation et aux données expérimentales sur plusieurs problèmes industriels. Des améliorations de précision des résultats et des réductions des temps de calculs ont été notés. Cependant, dans cette dernière partie de la thèse, des propositions de développements supplémentaires ont été discutées au niveau de l'adaptation en espace-temps, des solveurs et des modélisations de procédés industriels.



# BIBLIOGRAPHY

- [AAIB 10] AAIB. *Report on the accident to Boeing 777-236ER, G-YMMM, at London Heathrow Airport on 17 January 2008*. Rapport technique, 2010. (Cited page 1.)
- [Abgrall 14] R. Abgrall, H. Beaugendrea & C. Dobrzynska. *An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques*. *Journal of Computational Physics*, vol. 257, pages 83 – 101, 2014. (Cited page 234.)
- [Adjerid 86] S. Adjerid & J. Flaherty. *A Moving Finite Element Method with Error Estimation and Refinement for One-Dimensional Time Dependent Partial Differential Equations*. *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pages 778–796, 1986. (Cited pages 7 and 80.)
- [Advani 87] S. Advani & C. Tucker. *The Use of Tensors to Describe and Predict Fiber Orientation in Short Fiber Composites*. *Journal of Rheology*, vol. 31, no. 8, pages 751–784, 1987. (Cited pages 87 and 88.)
- [Advani 90] S. Advani & C. Tucker. *A numerical simulation of short fiber orientation in compression molding*. *Polymer Composites*, vol. 11, no. 3, pages 164 – 173, 1990. (Cited page 93.)
- [Agouzal 99] A. Agouzal, K. Lipnikov & Y. Vassilevski. *Adaptive generation of quasi-optimal tetrahedral meshes*. *East-West J. Numer. Math.*, vol. 7, pages 223–244, 1999. (Cited page 99.)
- [Alauzet 07] F. Alauzet, P. J. Frey, P. L. George & B. Mohammadi. *3D Transient Fixed Point Mesh Adaptation for Time-dependent Problems: Application to CFD Simulations*. *Journal of Computational Physics*, vol. 222, no. 2, pages 592 – 623, 2007. (Cited page 272.)
- [Alauzet 10a] F. Alauzet. *Size Gradation Control of Anisotropic Meshes*. *Finite Elements in Analysis and Design*, vol. 46, nos. 1 – 2, pages 181 – 202, January 2010. (Cited pages 246 and 247.)
- [Alauzet 10b] F. Alauzet, W. Hassan & M. Picasso. *Goal oriented, anisotropic, A posteriori error estimates for the Laplace equation*. In *Numerical Mathematics and Advanced Applications 2009*, pages 47–58. Springer Berlin Heidelberg, 2010. (Cited pages 7, 9, and 81.)

- [Alauzet 10c] F. Alauzet & A. Loseille. *High-order sonic boom modeling based on adaptive methods*. *Journal of Computational Physics*, vol. 229, no. 3, pages 561 – 593, 2010. (Cited page 100.)
- [Alauzet 12] F. Alauzet, A. Belme & A. Dervieux. *Anisotropic Goal-Oriented Mesh Adaptation for Time Dependent Problems*. In *Proceedings of the 20th International Meshing Roundtable*, pages 99–121. Springer Berlin Heidelberg, 2012. (Cited pages 96 and 250.)
- [Alauzet 13] F. Alauzet. *Efficient Moving Mesh Technique Using Generalized Swapping*. In *Proceedings of the 21st International Meshing Roundtable*, pages 17–37. Springer Berlin Heidelberg, 2013. (Cited page 267.)
- [Albensoeder 05] S. Albensoeder & HC. Kuhlmann. *Accurate three-dimensional lid-driven cavity flow*. *Journal of Computational Physics*, vol. 206, pages 536 – 558, 2005. (Cited page 177.)
- [Allgower 78] E.L. Allgower & S.F. McCormick. *Newton's method with mesh refinements for numerical solution of nonlinear two-point boundary value problems*. *Numerische Mathematik*, vol. 29, no. 3, pages 237–260, 1978. (Cited pages 7 and 80.)
- [Allmaras 12] S. Allmaras & F. Johnson. *Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model*. In *7th International Conference on Computational Fluid Dynamics*, Big Island, Hawaii, 2012. (Cited page 51.)
- [Arney 90] D. C. Arney & J. E. Flaherty. *An adaptive mesh-moving and local refinement method for time-dependent partial differential equations*. *ACM Trans. Math. Softw.*, vol. 16, no. 1, pages 48 – 71, 1990. (Cited pages 7 and 81.)
- [Arnold 84] D.N. Arnold, F. Brezzi & M. Fortin. *A stable finite element for the stokes equations*. *CALCOLO*, vol. 21, no. 4, pages 337–344, 1984. (Cited pages 36, 37, 40, and 150.)
- [Aulisa 06] E. Aulisa, S. Manservigi & P. Seshaiyer. *A computational multilevel approach for solving 2D Navier-Stokes equations over non-matching grids*. *Computer Methods in Applied Mechanics and Engineering*, vol. 195, nos. 33 – 36, pages 4604 – 4616, 2006. (Cited page 231.)
- [Babuška 78] I. Babuška & W. C. Rheinboldt. *A-posteriori error estimates for the finite element method*. *International Journal for Numerical Methods in Engineering*, vol. 12, no. 10, pages 1597 – 1615, 1978. (Cited pages 9, 95, and 118.)

- [Badia 09] S. Badia & R. Codina. *On a multiscale approach to the transient Stokes problem: Dynamic subscales and anisotropic space-time discretization*. Applied Mathematics and Computation, vol. 207, no. 2, pages 415 – 433, 2009. (Cited pages 31 and 37.)
- [Baiocchi 93] C. Baiocchi, F. Brezzi & L. P. Franca. *Virtual bubbles and Galerkin-least-squares type methods (Ga.L.S.)*. Computer Methods in Applied Mechanics and Engineering, vol. 105, no. 1, pages 125 – 141, 1993. (Cited pages 21, 32, and 34.)
- [Balay 97] S. Balay, W. Gropp, L. McInnes & B. Smith. *Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries*. In Modern Software Tools in Scientific Computing, pages 163 – 202. Birkhauser Press, 1997. (Cited page 55.)
- [Bank 85] R. E. Bank & A. Weizer. *Some A Posteriori Error Estimators For Elliptic Partial Differential Equations*. mathematics of computation, vol. 44, pages 283 – 301, 1985. (Cited page 95.)
- [Bank 93] R. E. Bank & R. K. Smith. *A Posteriori Error Estimates Based On Hierarchical Bases*. Siam journal on numerical analysis, vol. 30, no. 4, pages 921 – 935, 1993. (Cited page 95.)
- [Bao 11] Y. Bao, D. Zhou, C. Huang, Q. Wu & X. Chen. *Numerical prediction of aerodynamic characteristics of prismatic cylinder by finite element method with Spalart-Allmaras turbulence model*. Computers & Structures, vol. 89, nos. 3 – 4, pages 325 – 338, 2011. (Cited pages 55, 56, 57, and 58.)
- [Baptista 95] A. M. Baptista, E. E. Adams & P. Gresho. *Benchmarks for the transport equation: The convection-diffusion forum and beyond*, volume 47. American Geophysical Union, 1995. (Cited page 214.)
- [Barth 92] T. J. Barth, editeur. *Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations*, 1992. (Cited page 81.)
- [Basset 06] O. Basset. *Simulation numérique d'écoulements multi-fluides sur grille de calcul*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2006. (Cited pages 37, 40, 155, and 159.)
- [Bathe 04] K-J. Bathe & H. Zhang. *Finite element developments for general fluid flows with structural interactions*. International Journal for Numerical Methods in Engineering, vol. 60, no. 1, pages 213 – 232, 2004. (Cited page 3.)

- [Bathe 09] K-J Bathe & H. Zhang. *A mesh adaptivity procedure for {CFD} and fluid-structure interactions*. *Computers & Structures*, vol. 87, nos. 11 – 12, pages 604 – 617, 2009. (Cited page 233.)
- [Batkam Hemo 02] S. Batkam Hemo. *Thermique multidomaines en simulation numérique du remplissage 3D*. thèse de doctorat, Ecole nationale supérieure des mines de Paris, 2002. (Cited page 244.)
- [Bazilevs 07] Y. Bazilevs, V.M. Calo, J.A. Cottrell, T.J.R. Hughes, A. Reali & G. Scovazzi. *Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows*. *Computer Methods in Applied Mechanics and Engineering*, vol. 197, nos. 1 – 4, pages 173–201, 2007. (Cited page 42.)
- [Bearman 82] P.W. Bearman & E.D. Obasaju. *An experimental study of pressure fluctuations on fixed and oscillating square-section cylinders*. *Journal of Fluid Mechanics*, vol. 119, pages 297 – 321, 1982. (Cited page 58.)
- [Becker 96] R. Becker & R. Rannacher. *A Feed-Back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples*. *East-West J. Numer. Math*, vol. 4, pages 237 – 264, 1996. (Cited page 96.)
- [Belme 11] A. Belme. *Aérodynamique instationnaire et méthode adjointe*. PhD thesis, Nice university, 2011. (Cited pages 8, 97, 145, and 194.)
- [Belme 12] A. Belme, A. Dervieux & F. Alauzet. *Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows*. *Journal of Computational Physics*, vol. 231, no. 19, pages 6323 – 6348, 2012. (Cited page 250.)
- [Berger 84] M. J. Berger & J. Oliger. *Adaptive mesh refinement for hyperbolic partial differential equations*. *Journal of Computational Physics*, vol. 53, no. 3, pages 484 – 512, 1984. (Cited pages 7, 80, and 81.)
- [Bernacki 08] M. Bernacki, Y. Chastel & T. Coupez. *Level set method for the numerical modelling of primary recrystallization in the polycrystalline materials*. *Scripta Materialia*, vol. 58, no. 12, pages 1129 – 1132, 2008. (Cited pages 3, 53, 232, 234, and 238.)
- [Berrone 09] S. Berrone & M. Marro. *Space-time adaptive simulations for unsteady Navier-Stokes problems*. *Computers & Fluids*, vol. 38, no. 6, pages 1132 – 1144, 2009. (Cited pages 8 and 189.)
- [Bieterman 86] M. Bieterman, J. Flaherty & P. Moore. *Adaptive Refinement Methods for Non-Linear Parabolic Partial Differential Equations*. *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, vol. 16, no. 1, pages 339 – 358, 1986. (Cited pages 7 and 81.)

- [Bikchentaev 12] A. Bikchentaev. *On invertibility of some operator sums*. Lobachevskii Journal of Mathematics, vol. 33, no. 3, pages 216–222, 2012. (Cited page 146.)
- [Billon 16] L. Billon. *Analyse et validation des couches limites pour les écoulements turbulents*. thèse de doctorat, Ecole nationale supérieure des mines de Paris, 2013 - 2016. (Cited pages 300 and 360.)
- [Borouchaki 01] H. Borouchaki, D. Chapelle, P.L. George, P. Laug & P.J. Frey. *Estimateurs d'erreur géométriques et adaptation de maillage*. Maillage et adaptation, P.L. George ed., Traité Mécanique et Ingénierie des Matériaux, Série Méthodes Numériques, Hermès Science, Paris, vol. 95, no. 2, pages 279–310, 2001. (Cited pages xii, 131, and 133.)
- [Botella 98] O. Botella & R. Peyret. *Benchmark spectral results on the lid-driven cavity flow*. Computers and Fluids, vol. 27, pages 421 – 433, 1998. (Cited pages 172 and 173.)
- [Bottasso 04] C. L. Bottasso. *Anisotropic mesh adaption by metric-driven optimization*. International Journal for Numerical Methods in Engineering, vol. 60, no. 3, pages 597 – 639, 2004. (Cited pages 7, 81, and 124.)
- [Bourgault 09] Y. Bourgault, M. Picasso, F. Alauzet & A. Loseille. *On the use of anisotropic a posteriori error estimators for the adaptative solution of 3D inviscid compressible flows*, 2009. (Cited page 95.)
- [Boussetta 06] R. Boussetta, T. Coupez & L. Fourment. *Adaptative Remeshing based on a posteriori error Estimation for Forging Simulation*. Comput. Methods Appl. Mech. Engrg., vol. 195, pages 6626–6645, 2006. (Cited pages xii, 7, 76, 81, and 126.)
- [Boussinesq 03] J. Boussinesq. *Théorie analytique de la chaleur mise en harmonie avec la thermodynamique et avec la théorie mécanique de la lumière: Refroidissement et échauffement par rayonnement, conductibilité des tiges, lames et masses cristallines, courants de convection, théorie mécanique de la lumière*. 1903. xxxii, 625, [1] p. Cours de physique mathématique de la Faculté des sciences. Gauthier-Villars, 1903. (Cited page 58.)
- [Brezzi 92] F. Brezzi, M.O. Bristeau, L. P. Franca, M. Mallet & G. Rogé. *A relationship between stabilized finite element methods and the Galerkin method with bubble functions*. Computer Methods in Applied Mechanics and Engineering, vol. 96, no. 1, pages 117 – 129, 1992. (Cited pages 21, 32, and 34.)

- [Brezzi 94] F. Brezzi & A. Russo. *Choosing bubbles for advection-diffusion problems*. *Mathematical Models and Methods in Applied Sciences*, vol. 04, no. 04, pages 571–587, 1994. (Cited page 21.)
- [Brezzi 96] F. Brezzi. *Stabilization techniques and subgrid scales capturing*. UCD/CCM report. University of Colorado at Denver, Center for Computational Mathematics, 1996. (Cited page 37.)
- [Brooks 82] T. J.R. Brooks A. N. and Hughes. *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*. *Computer Methods in Applied Mechanics and Engineering*, vol. 32, nos. 1 – 3, pages 199 – 259, 1982. (Cited pages xi, 20, 25, 27, and 28.)
- [Bruchon 09] J. Bruchon, H. Dignonnet & T. Coupez. *Using a signed distance function for the simulation of metal forming processes: Formulation of the contact condition and mesh adaptation*. *International Journal for Numerical Methods in Engineering*, vol. 78, no. 8, pages 980 – 1008, 2009. (Cited pages 3, 232, 234, 235, and 236.)
- [Burchard 74] H. G. Burchard. *Splines (with optimal knots) are better*. *Applicable Analysis*, vol. 3, no. 4, pages 309–319, 1974. (Cited page 118.)
- [Calderer 13] R. Calderer & A. Masud. *Residual-based variational multiscale turbulence models for unstructured tetrahedral meshes*. *Computer Methods in Applied Mechanics and Engineering*, vol. 254, no. 0, pages 238 – 253, 2013. (Cited pages 40 and 42.)
- [Cangiani 05] A. Cangiani & E. Süli. *Enhanced residual-free bubble method for convection-diffusion problems*. *International Journal for Numerical Methods in Fluids*, vol. 47, nos. 10 – 11, pages 1307 – 1313, 2005. (Cited page 149.)
- [Canuto 96] C. Canuto & V. Van Kemenade. *Bubble-stabilized spectral methods for the incompressible Navier-Stokes equations*. *Computer Methods in Applied Mechanics and Engineering*, vol. 135, nos. 1–2, pages 35 – 61, 1996. (Cited page 157.)
- [Cao 02] W. Cao, W. Huang & R. Russell. *A Moving Mesh Method Based on the Geometric Conservation Law*. *SIAM Journal on Scientific Computing*, vol. 24, no. 1, pages 118–142, 2002. (Cited page 118.)
- [Castaños 96] J. Castaños & J. Savage. *The Dynamic Adaptation of Parallel Mesh-Based Computation*. In *SIAM 7th Symposium on Parallel and Scientific Computation*, 1996. (Cited page 7.)

- [Castro-Díaz 97] M. J. Castro-Díaz, F. Hecht, B. Mohammadi & O. Pironneau. *Anisotropic unstructured mesh adaption for flow simulations*. International Journal for Numerical Methods in Fluids, vol. 25, no. 4, pages 475 – 491, 1997. (Cited pages 7 and 81.)
- [Causin 05] P. Causin, J.F. Gerbeau & F. Nobile. *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*. Computer Methods in Applied Mechanics and Engineering, vol. 194, nos. 42 – 44, pages 4506 – 4527, 2005. (Cited page 3.)
- [Cea 64] J. Cea. *Approximation variationnelle des problèmes aux limites*. Ann. Inst. Fourier, vol. 14, no. 2, pages 345 – 444, 1964. (Cited page 97.)
- [Celia 92] M. Celia & L. Abriola. *A mass conservative numerical solution for two phase flow porous media with application to unsaturated flow*. Water Resources Research, vol. 28, no. 10, pages 2819 – 2828, 1992. (Cited pages 8 and 190.)
- [Chen 04a] L. Chen, P. Sun & J. Xu. *Optimal anisotropic meshes for minimizing interpolation errors in  $l^p$ -norm*, 2004. (Cited page 118.)
- [Chen 04b] Z. Chen & J. Feng. *An Adaptive Finite Element Algorithm with Reliable and Efficient Error Control for Linear Parabolic Problems*. Mathematics of Computation, vol. 73, no. 247, pages 1167 – 1193, 2004. (Cited pages 8 and 189.)
- [Chen 08] L. Chen & J. Xu. *Stability and accuracy of adapted finite element methods for singularly perturbed problems*. Numerische Mathematik, vol. 109, no. 2, pages 167–191, 2008. (Cited page 118.)
- [Chew 89] L. Chew. *Guaranteed-quality triangular meshes*. Tech. rep., Dept. of Computer Science, 1989. (Cited pages 7 and 81.)
- [Christie 76] I. Christie, D. F. Griffiths, A. R. Mitchell & O. C. Zienkiewicz. *Finite element methods for second order differential equations with significant first derivatives*. International Journal for Numerical Methods in Engineering, vol. 10, no. 6, pages 1389 – 1396, 1976. (Cited page 20.)
- [Ciarlet 91] P.G. Ciarlet. *Error Estimates for Elliptic Problems, in Handbook of Numerical Analysis, volume II of Finite Element methods (Part 1)*., 1991. (Cited page 97.)
- [Claisse 11] A. Claisse & P. Frey. *A nonlinear {PDE} model for reconstructing a regular surface from sampled data using a level set formulation on triangular meshes*. Journal of Computational Physics, vol. 230, no. 12, pages 4636 – 4656, 2011. (Cited page 100.)

- [Codina 98] R. Codina. *Comparison of some finite element methods for solving the diffusion-convection-reaction equation*. *Computer Methods in Applied Mechanics and Engineering*, vol. 156, nos. 1 – 4, pages 185 – 210, 1998. (Cited page 27.)
- [Codina 00a] R. Codina. *Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods*. *Comput. Methods Appl. Mech. Engrg.*, vol. 190, nos. 13 – 14, pages 1579 – 1599, 2000. (Cited pages 20, 27, 28, 37, 43, 45, 148, 149, and 154.)
- [Codina 00b] R. Codina & J. Blasco. *Stabilized finite element method for the transient Navier-Stokes equations based on a pressure gradient projection*. *Computer Methods in Applied Mechanics and Engineering*, vol. 182, nos. 3 – 4, pages 277–300, 2000. (Cited page 37.)
- [Codina 01] R. Codina. *Pressure Stability in Fractional Step Finite Element Methods for Incompressible Flows*. *Journal of Computational Physics*, vol. 170, no. 1, pages 112 – 140, 2001. (Cited page 37.)
- [Codina 07] R. Codina & J. Principe. *Dynamic subscales in the finite element approximation of thermally coupled incompressible flows*. *International Journal for Numerical Methods in Fluids*, vol. 54, nos. 6 – 8, pages 707 – 730, 2007. (Cited pages 37 and 42.)
- [Compère 10] G. Compère, J-F. Remacle, J. Jansson & J. Hoffman. *A mesh adaptation framework for dealing with large deforming meshes*. *International Journal for Numerical Methods in Engineering*, vol. 82, no. 7, pages 843 – 867, 2010. (Cited page 267.)
- [Coupez 91] T. Coupez. *Grandes transformations et remaillage automatique*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 1991. (Cited pages 7, 10, 74, 75, 76, and 124.)
- [Coupez 96] T. Coupez. *Stable-stabilized finite element for 3D forming calculation*. Rapport technique, CEMEF, 1996. (Cited pages 37, 76, and 154.)
- [Coupez 00] Thierry Coupez. *Génération de maillage et adaptation de maillage par optimisation locale*. *Revue européenne des éléments finis*, vol. 9, no. 4, pages 403–423, 2000. (Cited pages xii, 75, 76, and 79.)
- [Coupez 09] T. Coupez, W. Zerguine, H. Digonnet & Y. Mesri. *Adaptive anisotropic mesh and interface capturing problem*. In *Symposium on Research in the Mathematical Sciences: Current status and Future perspectives by CAMS*, 2009. (Cited pages 87 and 238.)

- [Coupez 11] T. Coupez. *Metric construction by length distribution tensor and edge based error for anisotropic adaptive meshing*. *Journal of Computational Physics*, vol. 230, no. 7, pages 2391 – 2405, 2011. (Cited pages 90, 92, 95, 98, 116, 132, 140, 141, 199, and 294.)
- [Coupez 13a] T. Coupez & E. Hachem. *Solution of high-Reynolds incompressible flow with stabilized finite element and adaptive anisotropic meshing*. *Computer Methods in Applied Mechanics and Engineering*, vol. 267, pages 65–85, 2013. (Cited pages 170 and 247.)
- [Coupez 13b] T. Coupez, G. Jannoun, N. Nassif, H.C. Nguyen, H. Digonnet & E. Hachem. *Adaptive time-step with anisotropic meshing for incompressible flows*. *Journal of Computational Physics*, vol. 241, no. 0, pages 195 – 211, 2013. (Cited pages 95, 98, 203, and 247.)
- [Coupez 13c] T. Coupez, G. Jannoun, J. Veysset & E. Hachem. *Edge-based anisotropic mesh adaptation for CFD applications*. In *Proceedings of the 21st International Meshing Roundtable*, pages 567 – 583. Springer, 2013.
- [Coupez 14] T. Coupez, L. Silva & E. Hachem. *Implicit boundary and anisotropic meshing*. submitted to SEMA SIMAI Springer Series, 2014. (Cited page 239.)
- [Courty 05] B. Courty, T. Roy, B. Koobus, M. Vázquez & A. Dervieux. *Error analysis for  $P_1$ -exact schemes*. In *Finite Element for Flow Problems*, Swansea, UK, April 4-6, 2005, 2005. (Cited page 146.)
- [Courty 06] F. Courty, T. Roy, B. Koobus & A. Dervieux. *Mesh adaptation by Optimal Control of a continuous model*. Rapport de recherche, INRIA, 2006. (Cited pages 145 and 146.)
- [Craig 87] A. Craig. *Accuracy estimates and adaptive refinements in finite element computations, I*. *International Journal for Numerical Methods in Engineering*, vol. 24, no. 6, pages 1228 – 1228, 1987. (Cited pages 7 and 80.)
- [Cremonesi 14] F. Cremonesi, G. Jannoun, L. Billon, J. F. Geneste, Y. H. Grunevald & E. Hachem. *Application of anisotropic edge-based unsteady mesh adaptation to aerodynamics*. *International Journal for Numerical Methods in Fluids*, vol. submitted, 2014. (Cited pages 56 and 178.)
- [de Sampaio 93] P. de Sampaio, P. Lyra, K. Morgan & N. Weatherill. *Petrov-Galerkin solutions of the incompressible Navier-Stokes equations in primitive variables with adaptive remeshing*. *Computer Methods in Applied Mechanics and Engineering*, vol. 106, nos. 1 – 2, pages 143 – 178, 1993. (Cited page 268.)

- [Dedé 08] L. Dedé, S. Micheletti & S. Perotto. *Anisotropic error control for environmental applications*. Applied Numerical Mathematics, vol. 58, no. 9, pages 1320 – 1339, 2008. (Cited page 95.)
- [Delaunay 34] B. Delaunay. *Sur la sphere vide*. Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk, vol. 7, nos. 793 – 800, pages 1 – 2, 1934. (Cited pages 74 and 75.)
- [Demkowicz 84] L. Demkowicz, J. T. Oden & T. Strouboulis. *Adaptive finite elements for flow problems with moving boundaries. I - Variational principles and a posteriori estimates*. Computer Methods in Applied Mechanics and Engineering, vol. 46, pages 217–251, 1984. (Cited page 95.)
- [Digonnet 01] H. Digonnet. *Répartitionnement dynamique, mailleur parallèle et leurs applications á la simulation numérique en mise en forme des matériaux*. thèse de doctorat, Ecole nationale supérieure des mines de Paris, 2001. (Cited page 125.)
- [Digonnet 03] H. Digonnet & T. Coupez. *Object-oriented programming for fast and easy development of parallel applications in forming processes simulation*. 2nd MIT Conference on Computational Fluid and Solid Mechanics, Cambridge : Royaume-Uni, pages 1922–1924, 2003. (Cited pages 10, 19, 37, 55, and 181.)
- [Dobrzynski 07] C. Dobrzynski & J-F. Remarclé. *Parallel mesh adaptation*. In International Journal for Numerical Methods in Engineering, pages 177–194. 2007. (Cited page 7.)
- [Donea 03] J. Donea & A. Huerta. *Finite element methods for flow problems*. Finite Element Methods for Flow Problems. John Wiley & Sons, 2003. (Cited page 37.)
- [Dubois 99] T. Dubois, F. Jauberteau & R. Temam. *Dynamic multilevel methods and the numerical simulation of turbulence*. Cambridge University Press, Cambridge, 1999. (Cited page 42.)
- [Dvinsky 91] A. S. Dvinsky. *Adaptive grid generation from harmonic maps on Riemannian manifolds*. Journal of Computational Physics, vol. 95, no. 2, pages 450 – 476, 1991. (Cited page 118.)
- [Dwight 08] R. P. Dwight. *Goal-oriented mesh adaptation for finite volume methods using a dissipation-based error indicator*. International Journal for Numerical Methods in Fluids, vol. 56, no. 8, pages 1193 – 1200, 2008. (Cited page 250.)

- [Eijkhout 91] V. Eijkhout & P. Vassilevski. *The Role of the Strengthened Cauchy-Buniakowskii-Schwarz Inequality in Multilevel Methods*, 1991. (Cited page 95.)
- [Elman 05] H.C. Elman, D.J. Silvester & A.J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical mathematics and scientific computation. Oxford University Press, 2005. (Cited page 168.)
- [Farhat 95] C. Farhat, M. Lesoinne & N. Maman. *Mixed explicit/implicit time integration of coupled aeroelastic problems: Three-field formulation, geometric conservation and distributed solution*. International Journal for Numerical Methods in Fluids, vol. 21, no. 10, pages 807 – 835, 1995. (Cited page 232.)
- [Farhat 98] C. Farhat, C. Degand, B. Koobus & M. Lesoinne. *Torsional springs for two-dimensional dynamic unstructured fluid meshes*. Comput. Methods Appl. Mech. Eng., vol. 163, pages 231–245, 1998. (Cited page 267.)
- [Farhat 06] C. Farhat, K. van der Zee & P. Geuzaine. *Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*. Computer Methods in Applied Mechanics and Engineering, vol. 195, nos. 17 – 18, pages 1973 – 2001, 2006. (Cited page 2.)
- [Farrell 00] P.A. Farrell. *Robust computational techniques for boundary layers*. Applied mathematics. Chapman & Hall/CRC, 2000. (Cited pages 161, 162, and 166.)
- [Farrell 04] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O’Riordan & G.I. Shishkin. *Global maximum norm parameter-uniform numerical method for a singularly perturbed convection-diffusion problem with discontinuous convection coefficient*. Mathematical and Computer Modelling, vol. 40, nos. 11 – 12, pages 1375 – 1392, 2004. (Cited page 161.)
- [Fedkiw 99] R. P. Fedkiw, T. Aslam, B. Merriman & S. Osher. *A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method)*. Journal of Computational Physics, vol. 152, no. 2, pages 457 – 492, 1999. (Cited page 232.)
- [Fernández 05] M-A Fernández & M. Moubachir. *A Newton method using exact jacobians for solving fluid-structure coupling*. Computers & Structures, vol. 83, nos. 2 – 3, pages 127 – 142, 2005. (Cited page 3.)

- [Ferziger 96] J. H. Ferziger & M. Perić. *Computational methods for fluid dynamics*. Springer-Verlag, Berlin, Germany, 1996. (Cited pages [xi](#) and [46](#).)
- [Ferziger 02] J.H. Ferziger & M. Perić. *Computational methods for fluid dynamics*. Springer London, Limited, 2002. (Cited page [2](#).)
- [Fidkowski 11] K. Fidkowski & Y. Luo. *Output-based space-time mesh adaptation for the compressible Navier-Stokes equations*. *Journal of Computational Physics*, vol. 230, no. 14, pages 5753 – 5773, 2011. (Cited pages [8](#) and [190](#).)
- [Formaggia 01] L. Formaggia & S. Perotto. *New anisotropic a priori error estimates*. *Numerische Mathematik*, vol. 89, no. 4, pages 641–667, 2001. (Cited pages [7](#), [81](#), and [95](#).)
- [Formaggia 03] L. Formaggia & S. Perotto. *Anisotropic error estimates for elliptic problems*. *Numerische Mathematik*, vol. 94, no. 1, pages 67–92, 2003. (Cited pages [7](#), [81](#), and [96](#).)
- [Formaggia 04] L. Formaggia, S. Micheletti & S. Perotto. *Anisotropic mesh adaptation in computational fluid dynamics: Application to the advection-diffusion-reaction and the Stokes problems*. *Applied Numerical Mathematics*, vol. 51, no. 4, pages 511 – 533, 2004. (Cited pages [7](#), [27](#), and [81](#).)
- [Förster 09] Ch. Förster, W. A. Wall & E. Ramm. *Stabilized finite element formulation for incompressible flow on distorted meshes*. *International Journal for Numerical Methods in Fluids*, vol. 60, no. 10, pages 1103 – 1126, 2009. (Cited page [149](#).)
- [Franca 92a] L. P. Franca, S. L. Frey & T. J.R. Hughes. *Stabilized finite element methods: I. Application to the advective-diffusive model*. *Computer Methods in Applied Mechanics and Engineering*, vol. 95, no. 2, pages 253 – 276, 1992. (Cited pages [27](#) and [159](#).)
- [Franca 92b] L.P. Franca & S.L. Frey. *Stabilized finite element methods: II. The incompressible Navier-Stokes equations*. *Comput. Methods Appl. Mech. Engrg.*, vol. 99, pages 209 – 233, 1992. (Cited page [170](#).)
- [Franca 95] L.P. Franca & Farhat C. *Bubble functions prompt unusual stabilized finite element methods*. *Comput. Methods Appl. Mech. Engrg.*, vol. 123, pages 229 – 308, 1995. (Cited pages [20](#), [21](#), [32](#), [34](#), and [37](#).)
- [Franca 96] L.P. Franca & A. Russo. *Deriving upwinding, mass lumping and selective reduced integration by residual-free bubbles*. *Applied Mathematics Letters*, vol. 9, no. 5, pages 83 – 88, 1996. (Cited page [21](#).)

- [Franca 97] L.P. Franca, C. Farhat, A. Macedo & M. Lesoinne. *Residual-Free Bubbles for the Helmholtz Equation*. International Journal for Numerical Methods in Engineering, vol. 40, pages 4003 – 4009, 1997. (Cited pages 20 and 21.)
- [Franca 98] L.P. Franca, A. Nesliturk & M. Stynes. *On the stability of residual-free bubbles for convection-diffusion problems and their approximation by a two-level finite element method*. Computer Methods in Applied Mechanics and Engineering, vol. 166, nos. 1 – 2, pages 35 – 49, 1998. Advances in Stabilized Methods in Computational Mechanics. (Cited pages 21 and 159.)
- [Franca 00] L.P. Franca & F. Valentin. *On an improved unusual stabilized finite element method for the advective-reactive-diffusive equation*. Computer Methods in Applied Mechanics and Engineering, vol. 190, nos. 13 – 14, pages 1785 – 1800, 2000. (Cited page 27.)
- [Franca 01] L. P. Franca & A. Nesliturk. *On a two-level finite element method for the incompressible Navier-Stokes equations*. International Journal for Numerical Methods in Engineering, vol. 52, no. 4, pages 433 – 453, 2001. (Cited pages 36 and 37.)
- [Franca 02] L. P. Franca & L. Tobiska. *Stability of the Residual Free Bubble Method for Bilinear Finite Elements on Rectangular Grids*. IMA J. Numer. Anal, vol. 22, no. 1, pages 73 – 87, 2002. (Cited pages 157 and 160.)
- [Frey 01] P. J. Frey. *YAMS: A fully automatic adaptive isotropic surface remeshing procedure*. Rapport technique 0252, INRIA, 2001. (Cited pages 74, 75, 124, and 127.)
- [Frey 05] P. Frey & F. Alauzet. *Anisotropic mesh adaptation for CFD computations*. Comput. Methods Appl. Mech. Engrg., vol. 194, pages 5068 – 5082, 2005. (Cited pages 7, 81, and 95.)
- [Frochte 09] J. Frochte & W. Heinrichs. *A splitting technique of higher order for the Navier-Stokes equations*. Journal of Computational and Applied Mathematics, vol. 228, no. 1, pages 373 – 390, 2009. (Cited pages 211 and 251.)
- [Galeão 88] A.C. Galeão & E.G. Dutra do Carmo. *A consistent approximate upwind Petrov-Galerkin method for convection-dominated problems*. Computer Methods in Applied Mechanics and Engineering, vol. 68, no. 1, pages 83 – 95, 1988. (Cited pages 29 and 30.)
- [Galeão 04] A.C. Galeão, R.C. Almeida, S.M.C. Malta & A.F.D. Loula. *Finite element analysis of convection dominated reaction-diffusion problems*.

- Applied Numerical Mathematics, vol. 48, no. 2, pages 205 – 222, 2004. (Cited pages 29 and 30.)
- [Gaston 97] L. Gaston. *Simulation numérique par éléments finis bidimensionnels du remplissage de moules de fonderie et étude expérimentale sur maquette hydraulique*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 1997. (Cited page 47.)
- [George 91] P.L. George, F. Hecht & M.G. Vallet. *Creation of internal points in Voronoi's type method. Control adaptation*. Advances in Engineering Software and Workstations, vol. 13, nos. 5 – 6, pages 303 – 312, 1991. (Cited pages 7, 81, and 85.)
- [George 99] P. L. George. *Tet Meshing: Construction, Optimization, And Adaptation*. In Proceedings, 8th International Meshing Roundtable, pages 133–141, South Lake Tahoe, CA, U.S.A., 1999. (Cited pages 74, 75, and 124.)
- [Georgiev 09] K. Georgiev, N. Kosturski, S. Margenov & J. Star. *On adaptive time stepping for large-scale parabolic problems: Computer simulation of heat and mass transfer in vacuum freeze-drying*. Journal of Computational and Applied Mathematics, vol. 226, no. 2, pages 268 – 274, 2009. Special Issue: Large scale scientific computations. (Cited pages 8 and 190.)
- [Gerbeau 03] J-F. Gerbeau & M. Vidrascu. *A quasi-newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows*. Mathematical Modelling and Numerical Analysis, vol. 37, no. 4, pages 631–647, 2003. (Cited page 3.)
- [Geuzaine 09] C. Geuzaine & 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, pages 1309 – 1331, 2009. (Cited page 124.)
- [Ghia 82] U. Ghia, K. N. Ghia & C. T. Shin. *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method*. Journal of Computational Physics, vol. 48, no. 3, pages 387–411, 1982. (Cited page 170.)
- [Glowinski 95] R. Glowinski, T-W. Pan, A. J. Kearsley & J. Periaux. *Numerical simulation and optimal shape for viscous flow by a fictitious domain method*. International Journal for Numerical Methods in Fluids, vol. 20, nos. 8 – 9, pages 695 – 711, 1995. (Cited page 232.)

- [Gravemeier 06] V. Gravemeier. *Variational Multiscale Large Eddy Simulation of Turbulent Flows Using a Two-Grid Finite Element or Finite Volume Method*. In *Numerical Mathematics and Advanced Applications*, pages 788–795. Springer Berlin Heidelberg, 2006. (Cited page 37.)
- [Grotjans 98] H. Grotjans & F. Menter. *Wall functions for general application CFD codes*. In *ECCOMAS 98, Proceedings of the 4th Computational Fluid Dynamics Conference*, pages 1112 – 1117, 1998. (Cited page 50.)
- [Gruau 05] C. Gruau & T. Coupez. *3D Tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric*. *Comput. Methods Appl. Mech. Engrg.*, vol. 194, pages 4951–4976, 2005. (Cited pages 7, 74, 76, 81, 95, 124, 125, 192, and 238.)
- [Guégan 10] D. Guégan, O. Allain, A. Dervieux & F. Alauzet. *An  $L_\infty$ - $L_p$  mesh-adaptive method for computing unsteady bi-fluid flows*. *International Journal for Numerical Methods in Engineering*, vol. 84, no. 11, pages 1376 – 1406, 2010. (Cited pages 293 and 294.)
- [Guermond 11] J. L. Guermond & P. D. Mineev. *Start-up flow in a three-dimensional lid-driven cavity by means of a massively parallel direction splitting algorithm*. *International Journal for Numerical Methods in Fluids*, vol. <http://dx.doi.org/10.1002/fld.2583>, page online, 2011. (Cited pages 172 and 173.)
- [Hachem 09] E. Hachem. *Stabilized Finite Element Method for Heat Transfer and Turbulent Flows inside Industrial Furnaces*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2009. (Cited pages xi, 4, 5, 6, 19, 28, 29, 37, 40, 44, 46, 49, 50, 152, 154, 156, 181, 244, and 323.)
- [Hachem 10a] E. Hachem, H. Digonnet, E. Massoni & T. Coupez. *Enriched finite element spaces for transient conduction heat transfer*. *Applied Mathematics & Computations*, vol. 217, pages 3929 – 3943, 2010. (Cited pages 3, 20, 232, and 234.)
- [Hachem 10b] E. Hachem, B. Rivaux, T. Klozcko, H. Digonnet & T. Coupez. *Stabilized finite element method for incompressible flows with high Reynolds number*. *Journal of Computational Physics*, vol. 229, pages 8643 – 8665, 2010. (Cited pages 2, 34, 35, 45, 170, 172, 174, 205, and 297.)
- [Hachem 10c] E. et. al Hachem. *Enriched finite element spaces for transient conduction heat transfer*. *Applied Mathematics and Computation*, vol. 217, no. 8, pages 3929 – 3943, 2010. (Cited pages 21, 31, and 32.)

- [Hachem 13] E. Hachem, G. Jannoun, J. Veysset, M. Henri, R. Pierrot, I. Poitroult, E. Massoni & T. Coupez. *Modeling of heat transfer and turbulent flows inside industrial furnaces*. Simulation Modelling Practice and Theory, vol. 30, no. 0, pages 35 – 53, 2013. (Cited pages 3, 53, 232, and 234.)
- [Hachem 14] E. Hachem, G. Jannoun, J. Veysset & T. Coupez. *On the stabilized finite element method for steady convection-dominated problems with anisotropic mesh adaptation*. Applied Mathematics and Computation, vol. 232, no. 0, pages 581 – 594, 2014. (Cited page 162.)
- [Han 97] Z. Han & R. Reitz. *A temperature wall function formulation for variable-density turbulent flows with application to engine convective heat transfer modeling*. International Journal of Heat and Mass Transfer, vol. 40, no. 3, pages 613 – 625, 1997. (Cited pages 325 and 326.)
- [Harari 92] I. Harari & T. J. R. Hughes. *Galerkin/least-squares finite element methods for the reduced wave equation with non-reflecting boundary conditions in unbounded domains*. Computer Methods in Applied Mechanics and Engineering, vol. 98, no. 3, pages 411 – 454, 1992. (Cited pages 20 and 149.)
- [Harari 94] I. Harari & T. J. R. Hughes. *Stabilized finite element methods for steady advection-diffusion with production*. Computer Methods in Applied Mechanics and Engineering, vol. 115, pages 165 – 191, 1994. (Cited page 20.)
- [Hassan 07] O. Hassan, K. Morgan & N. Weatherill. *Unstructured mesh methods for the solution of the unsteady compressible flow equations with moving boundary components*. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 365, no. 1859, pages 2531–2552, 2007. (Cited page 267.)
- [Hecht 14] Frédéric Hecht & Raphaël Kuate. *An Approximation of Anisotropic Metrics from Higher Order Interpolation Error for Triangular Mesh Adaptation*. J. Comput. Appl. Math., vol. 258, pages 99 – 115, 2014. (Cited pages 7 and 81.)
- [Heiligenstaedt 71] W. Heiligenstaedt. *Thermique appliquée aux fours industriels*. Paris: Dunod, 1971. (Cited page 64.)
- [Heinrich 77] J. C. Heinrich, P. S. Huyakorn, O. C. Zienkiewicz & A. R. Mitchell. *An 'upwind' finite element scheme for two-dimensional convective transport equation*. International Journal for Numerical Methods in Engineering, vol. 11, no. 1, pages 131 – 143, 1977. (Cited page 20.)

- [Hoffman 09] J. Hoffman & C. Johnson. *The Mathematical Secret of Flight*, 2009. (Cited page 2.)
- [Hoffman 14] J. Hoffman. *Towards a parameter-free method for high Reynolds number turbulent flow simulation based on adaptive finite element approximation*. Rapport technique, Computational Technology Laboratory Department of High Performance Computing and Visualization School of Computer Science and Communication Stockholm, Sweden, 2014. (Cited page 2.)
- [Huang 05] Weizhang Huang. *Metric tensors for anisotropic mesh generation*. J. Comput. Phys., vol. 204, no. 2, pages 633 – 665, 2005. (Cited pages 7, 81, 162, and 163.)
- [Hughes 86a] T. J.R. Hughes, M. Mallet & M. Akira. *A new finite element formulation for computational fluid dynamics: II. Beyond {SUPG}*. Computer Methods in Applied Mechanics and Engineering, vol. 54, no. 3, pages 341 – 355, 1986. (Cited page 29.)
- [Hughes 86b] Thomas J.R. Hughes & Michel Mallet. *A new finite element formulation for computational fluid dynamics: III. The generalized streamline operator for multidimensional advective-diffusive systems*. Computer Methods in Applied Mechanics and Engineering, vol. 58, no. 3, pages 305 – 328, 1986. (Cited pages 20 and 27.)
- [Hughes 86c] T.J.R. Hughes, L.P. Franca & M. Balestra. *A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations*. Comput. Methods Appl. Mech. Engrg., vol. 59, pages 85 – 99, 1986. (Cited page 170.)
- [Hughes 89] T. J. R. Hughes, L. Franca & G. Hulbert. *A new finite element formulation for uid dynamics: Viii. The Galerkin/least-squares method for the advection-diffusive equations*. Comput. Methods Appl. Mech. Engrg., vol. 73, pages 173 – 189, 1989. (Cited page 20.)
- [Hughes 95] T. J.R. Hughes. *Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods*. Computer Methods in Applied Mechanics and Engineering, vol. 127, nos. 1 – 4, pages 387 – 401, 1995. (Cited pages 20, 37, and 40.)
- [Hughes 98] T. J. R. Hughes, G. R. Feijóo, L. Mazzei & J. N. Quincy. *The variational multiscale method - a paradigm for computational mechanics*. Comput. Methods Appl. Mech. Engrg., vol. 166, pages 3–24, 1998. (Cited pages xi, 37, and 40.)

- [Hughes 01] T. J. R. Hughes, A.A. Oberai & L. Mazzei. *Large eddy simulation of turbulent channel flows by the variational multiscale method*. *Physics of Fluids*, vol. 13, pages 1784–1799, 2001. (Cited pages 37 and 40.)
- [Ilinca 11] F. Ilinca & J-F. Hetu. *A Finite element immersed boundary method for fluid flow around rigid objects*. *International Journal for Numerical Methods in Fluids*, vol. 65, pages 856–875, 2011. (Cited page 234.)
- [Ishii 98] T. Ishii, C. Zhang & S. Sugiyama. *Numerical Modeling Of An Industrial Aluminium Melting Furnace*. *Journal Of Energy Resources Technology*, vol. 120, pages 276 – 284, 1998. (Cited page 52.)
- [Jannoun 11] G. Jannoun. *Time adaptive stabilized finite element method for long time simulations of heat transfer and turbulent flows inside industrial furnaces*. *Rapport technique, first year annual report, cemef, Mines-Paristech*, 2011. (Cited page 189.)
- [Jannoun 12] G. Jannoun, J. Veysset, E. Hachem & T. Coupez. *Adaptive Time-step with Anisotropic Meshing for Incompressible Flows*. In *41st National Congress on Numerical Analysis*, May 2012.
- [Jannoun 13a] G. Jannoun, E. Hachem & T. Coupez. *Stabilized finite element method for incompressible flows using new space and time anisotropic adaptive technique*. In *17th International Conference on Finite Elements in Flow Problems*, May 2013.
- [Jannoun 13b] G. Jannoun, J. Veysset, E. Hachem, J-F Zaragoci & T. Coupez. *Fully space-time metric based anisotropic mesh adaptation for unsteady problems*. In *Adaptive Modeling and Simulation 2013-Proceedings of the 6th International Conference on Adaptive Modeling and Simulation, ADMOS 2013*, 2013.
- [Jannoun 14a] G. Jannoun. *Time accurate anysotropic edge-based mesh adaptation for unsteady flows*. *Finalist at the 25th Annual Robert J. Melosh Medal Competition for the Best Student Paper on Finite Element Analysis*, April 2014.
- [Jannoun 14b] G. Jannoun, E. Hachem, J. Veysset & T. Coupez. *Adaptive time-step with anisotropic meshing for unsteady convection-dominated problems*. *Applied Mathematical Modelling*, vol. Accepted, 2014. (Cited page 214.)
- [Jannoun 14c] G. Jannoun, E. Hachem, J. Veysset & T. Coupez. *Immersed volume method with anisotropic mesh adaptation and time-stepping control for fluid structure interaction and heat transfer applications*. *Oral communication at the 11th World Congress on Computational Mechanics WCCM XI*, July 2014.

- [John 48] F. John. *Extremum Problems with Inequalities as Subsidiary Conditions*. In K. Friedrichs, O. Neugebauer & J. Stoker, editeurs, *Studies and Essays: Courant Anniversary Volume*. Wiley-Interscience, 1948. (Cited page 247.)
- [Johnson 99] A. Johnson & T. Tezduyar. *Advanced mesh generation and update methods for 3D flow simulations*. *Computational Mechanics*, vol. 23, no. 2, pages 130–143, 1999. (Cited page 267.)
- [Johnston 04] H. Johnston & J-G. Liu. *Accurate, stable and efficient Navier-Stokes solvers based on explicit treatment of the pressure term*. *Journal of Computational Physics*, vol. 199, no. 1, pages 221 – 259, 2004. (Cited page 187.)
- [Jones 72] W. P. Jones & B. E. Launder. *The predictions of laminarisation with a two equation model of turbulence*. *Int J Heat Mass Transfer*, vol. 15, no. 301, 1972. (Cited pages 48 and 53.)
- [Kast 13] S. Kast & K. Fidkowski. *Output-based mesh adaptation for high order Navier-Stokes simulations on deformable domains*. *Journal of Computational Physics*, vol. 252, no. 0, pages 468 – 494, 2013. (Cited pages 8 and 190.)
- [Kavetski 02] D. Kavetski, P. Binning & S. Sloan. *Adaptive backward Euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow*. *International Journal for Numerical Methods in Engineering*, vol. 53, no. 6, pages 1301–1322, 2002. (Cited pages 8 and 189.)
- [Kolmogorov 91] A. N. Kolmogorov. *The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers*. *Proceedings: Mathematical and Physical Sciences*, vol. 434, no. 1890, pages 9 – 13, 1991. (Cited page 46.)
- [Koobus 05] B. Koobus, L. Hascoët, F. Alauzet, A. Loseille, Y. Mesri & A. Dervieux. *“Continuous mesh adaptation models for CFD”*. In *Recent Trends in Aerospace Design and Optimization*, pages 3–11. Tata-McGraw Hill, New Delhi, 2005. SAROD-2005, Hyderabad, India. (Cited page 97.)
- [Kopteva 05] N. Kopteva, N. Madden & M. Stynes. *Grid equidistribution for reaction-diffusion problems in one dimension*. *Numerical Algorithms*, vol. 40, no. 3, pages 305–322, 2005. (Cited pages 25 and 118.)
- [Kunert 00] G. Kunert. *An a posteriori residual error estimator for the finite element method on anisotropic tetrahedral meshes*. *Numerische Mathematik*, vol. 86, no. 3, pages 471–490, 2000. (Cited pages 7 and 81.)

- [Kuzmin 07] D. Kuzmin, O. Mierka & S. Turek. *On the implementation of the  $k$ -epsilon turbulence model in incompressible flow solvers based on a finite element discretization*. International Journal of Computing Science and Mathematics, vol. 2/3/4, pages 193–206, 2007. (Cited page 64.)
- [Ladyzhenskaya 63] O. A. Ladyzhenskaya. *The Mathematical Theory of Viscous Incompressible Flow*. Gordon and Breach, 1963. (Cited page 36.)
- [Laemmer 97] L. Laemmer. *Parallel Mesh Generation*. In Proceedings of the 4th International Symposium on Solving Irregularly Structured Problems in Parallel, IRREGULAR '97, pages 1 – 12. Springer-Verlag, 1997. (Cited page 7.)
- [Lakehal 02] D. Lakehal, M. Meier & M. Fulgosi. *Interface tracking towards the direct simulation of heat and mass transfer in multiphase flows*. International Journal of Heat and Fluid Flow, vol. 23, no. 3, pages 242 – 257, 2002. (Cited pages 3, 232, and 233.)
- [Lauder 74] B.E. Lauder & D.B. Spalding. *The numerical computation of turbulent flows*. Computer Methods in Applied Mechanics and Engineering, vol. 3, no. 2, pages 269 – 289, 1974. (Cited pages 325 and 326.)
- [Le Quéré 05] P. Le Quéré, H. Paillère, E. Dick, R. Becker, M. Braack & J. Locke. *Modelling of natural convection flows with large temperature differences: A benchmark problem for low mach number solvers. part 1. reference solutions*. ESAIM: Mathematical Modelling and Numerical Analysis, vol. 39, no. 3, pages 609–616, 2005. (Cited page 59.)
- [Le Tallec 01] P. Le Tallec & J. Mouro. *Fluid structure interaction with large structural displacements*. Computer Methods in Applied Mechanics and Engineering, vol. 190, nos. 24 – 25, pages 3039 – 3067, 2001. (Cited page 3.)
- [Lee 03] L. Lee & R. Leveque. *An immersed interface method for incompressible navier-stokes equations*. SIAM Journal on Scientific Computing, vol. 25, pages 832 – 856, 2003. (Cited page 232.)
- [Löhner 89] R. Löhner. *Adaptive remeshing for transient problems*. Computer Methods in Applied Mechanics and Engineering, vol. 75, nos. 1 – 3, pages 195 – 214, 1989. (Cited pages 7 and 81.)
- [Löhner 92] R. Löhner & J. Baum. *Adaptive  $h$ -refinement on 3D unstructured grids for transient problems*. International Journal for Numerical Methods in Fluids, vol. 14, no. 12, pages 1407 – 1419, 1992. (Cited page 268.)

- [Li 05] X Li, M. S. Shephard & M. W. Beall. *3D anisotropic mesh adaptation by mesh modification*. Computer Methods in Applied Mechanics and Engineering, vol. 194, nos. 48 – 49, pages 4915 – 4950, 2005. Unstructured Mesh Generation. (Cited page 124.)
- [Linβ 85] T. Linβ. *Layer-adapted meshes for reaction-convection-diffusion problems*. Springer Berlin Heidelberg, 1985. (Cited page 161.)
- [Linβ 99] T. Linβ & M. Stynes. *A hybrid difference scheme on a Shishkin mesh for linear convection-diffusion problems*. Applied Numerical Mathematics, vol. 31, no. 3, pages 255 – 270, 1999. (Cited page 161.)
- [Lo 85] S. H. Lo. *A new mesh generation scheme for arbitrary planar domains*. International Journal for Numerical Methods in Engineering, vol. 21, no. 8, pages 1403 – 1426, 1985. (Cited page 74.)
- [Lo 90] S. H. Lo. *An adaptive mesh-moving and local refinement method for time-dependent partial differential equations*. ACM Transactions on Mathematical Softwares, vol. 16, no. i, pages 48 – 79, 1990. (Cited pages 7 and 80.)
- [Loseille 09] A. Loseille. *Anisotropic 3D hessian-based multi-scale and adjoint-based mesh adaptation for Computational fluid dynamics Application to high fidelity sonic boom prediction*. PhD thesis, Paris VI, 2009. (Cited pages 87, 95, 138, and 247.)
- [Loseille 12] A. Loseille & R. Löhner. *Robust Boundary Layer Mesh Generation*. In Proceeding in 21th International Meshing Roundtable, pages 493 – 511. Springer-Verlag, 2012. (Cited page 7.)
- [Lörcher 07] F. Lörcher, G. Gassner & C.-D. Munz. *A Discontinuous Galerkin Scheme Based on a Space-Time Expansion. I. Inviscid Compressible Flow in One Space Dimension*. Journal of Scientific Computing, vol. 32, no. 2, pages 175–199, 2007. (Cited pages 8 and 188.)
- [Lörcher 08] F. Lörcher & C-D. Gassner G. an Munz. *An explicit discontinuous Galerkin scheme with local time-stepping for general unsteady diffusion equations*. Journal of Computational Physics, vol. 227, no. 11, pages 5649 – 5670, 2008. (Cited pages 8 and 188.)
- [Marcum 95] D. L. Marcum & N. P. Weatherill. *Unstructured grid generation using iterative point insertion and local reconnection*. AIAA Journal, vol. 33, pages 1619–1625, 1995. (Cited page 81.)
- [Masud 04] A. Masud & R. A. Khurram. *A multiscale/stabilized finite element method for advection-diffusion equation*. Comput. Methods Appli.

- Mech. Engrg., vol. 193, pages 1997 – 2018, 2004. (Cited pages 33, 41, 43, 44, 155, and 158.)
- [Mavriplis 90] D. J. Mavriplis. *Adaptive mesh generation for viscous flows using Delaunay triangulation*. J. Comput. Phys., vol. 90, no. 2, pages 271 – 291, 1990. (Cited pages 7 and 81.)
- [Micheletti 02] S. Micheletti, S. Perotto & M. Picasso. *Some remarks on the stability coefficients and bubble stabilization of FEM on anisotropic meshes*. Computing and Visualization in Science, 2002. (Cited pages 149 and 152.)
- [Micheletti 04] S. Micheletti, S. Perotto & M. Picasso. *Stabilized Finite Elements on Anisotropic Meshes: A Priori Error Estimates for the Advection-Diffusion and the Stokes Problems*. SIAM Journal on Numerical Analysis, vol. 41, pages 1131 – 1162, 2004. (Cited pages 149 and 161.)
- [Micheletti 06] S. Micheletti & S. Perotto. *Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator*. Computer Methods in Applied Mechanics and Engineering, vol. 195, nos. 9 – 12, pages 799 – 835, 2006. (Cited pages 7, 81, and 95.)
- [Micheletti 08a] S. Micheletti & S. Perotto. *Output Functional Control for Nonlinear Equations Driven by Anisotropic Mesh Adaption: The Navier-Stokes Equations*. SIAM Journal on Scientific Computing, vol. 30, no. 6, pages 2817–2854, 2008. (Cited pages 7, 81, and 268.)
- [Micheletti 08b] S. Micheletti & S. Perotto. *Output Functional Control for Nonlinear Equations Driven by Anisotropic Mesh Adaption: The Navier-Stokes Equations*. SIAM J. Sci. Comput., vol. 30, no. 6, pages 2817 – 2854, 2008. (Cited page 191.)
- [Micheletti 08c] S. Micheletti & S. Perotto. *Space-Time Adaption for Advection-Diffusion-Reaction Problems on Anisotropic Meshes*. In K. Kunisch, G. Of & O. Steinbach, editeurs, Numerical Mathematics and Advanced Applications, pages 49–56. Springer Berlin Heidelberg, 2008. (Cited pages 8 and 191.)
- [Mitchell 78] A. Mitchell & D. Griffiths. *Generalised Galerkin methods for second order equations with significant first derivative terms*. In G. Watson, editeur, Numerical Analysis, volume 630 of *Lecture Notes in Mathematics*, pages 90–104. Springer Berlin / Heidelberg, 1978. (Cited page 27.)

- [Mittal 00] S. Mittal. *On the performance of high aspect ratio elements for incompressible flows*. *Comput. Methods Appl. Mech. Engrg.*, vol. 188, pages 269 – 287, 2000. (Cited page 149.)
- [Modest 93] Michael F. Modest. *Radiative heat transfer*. McGraw-Hill, New-York, 1993. (Cited pages 53 and 54.)
- [Moës 99] N. Moës, J. Dolbow & T. Belytschko. *A finite element method for crack growth without remeshing*. *International Journal for Numerical Methods in Engineering*, vol. 46, no. 1, pages 131 – 150, 1999. (Cited page 232.)
- [Moin 98] P. Moin & K. Mahesh. *Direct numerical simulation: a tool in turbulence research*. *Annual Review of Fluid Mechanics*, vol. 30, no. 1, pages 539 – 578, 1998. (Cited page 46.)
- [Morton 96] K. W. Morton. *Numerical solution of convection-diffusion problems*. Chapman & Hall, 1996. (Cited page 160.)
- [Müller 01] J.D. Müller & M.B. Giles. *Solution Adaptive Mesh Refinement Using Adjoint Error Analysis*. AIAA Paper, pages 2004 – 4536, 2001. (Cited page 250.)
- [Nguyen 09] H. Nguyen, M. Gunzburger, L. Ju & J. Burkardt. *Adaptive anisotropic meshing for steady convection-dominated problems*. *Computer Methods in Applied Mechanics and Engineering*, vol. 198, nos. 37 – 40, pages 2964 – 2981, 2009. (Cited pages 21, 22, 27, 161, 162, 163, and 166.)
- [Nieckele 04] A. O. Nieckele, M. F. Naccache & M. S. P. Gomes. *Numerical Modeling Of An Industrial Aluminium Melting Furnace*. *Journal Of Energy Resources Technology*, vol. 126, pages 72 – 81, 2004. (Cited page 52.)
- [Nobile 01] F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*. PhD thesis, 2001. (Cited page 3.)
- [Oden 99] J. T. Oden & S. Prudhomme. *Goal-Oriented Error Estimation and Adaptivity for the Finite Element Method*. *Comput. Math. Appl*, vol. 41, pages 735 – 756, 1999. (Cited page 96.)
- [Oliver 08] T. A. Oliver. *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, 2008. (Cited page 52.)

- [Oliver 09] T. Oliver & D. Darmofal. *Impact of Turbulence Model Irregularity on High-Order Discretizations*. In 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition, Orlando, Florida, 2009. (Cited page 51.)
- [Paillère 05] H. Paillère, P. Le Quéré, C. Weisman, J. Vierendeels, E. Dick, M. Braack, F. Dabbene, A. Beccantini, E. Studer, T. Kloczko, C. Corre, V. Heuveline, M. Darbandi & S. Hosseinizadeh. *Modelling of Natural Convection Flows with Large Temperature Differences: A Benchmark Problem for Low Mach Number Solvers. Part 2. Contributions to the June 2004 conference*. ESAIM: Mathematical Modelling and Numerical Analysis, vol. 39, pages 617–621, 5 2005. (Cited pages 59, 61, 62, and 220.)
- [Pain 01] C.C. Pain, A.P. Umpleby, C.R.E. de Oliveira & A.J.H. Goddard. *Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations*. Computer Methods in Applied Mechanics and Engineering, vol. 190, nos. 29 – 30, pages 3771 – 3796, 2001. (Cited pages 7 and 81.)
- [Paniconi 91] C. Paniconi, A. Aldama & E. Wood. *Numerical evaluation of iterative and noniterative methods for the solution of nonlinear Richards equation*. Water Resources Research, vol. 27, no. 6, pages 1147 – 1163, 1991. (Cited pages 8 and 190.)
- [Park 83] K. Park. *Stabilization of partitioned solution procedure for pore fluid-soil interaction analysis*. International Journal for Numerical Methods in Engineering, vol. 19, no. 11, pages 1669 – 1673, 1983. (Cited page 2.)
- [Patankar 80] S. V. Patankar. *Numerical heat transfer and fluid flow*. Series In Computational And Physical Processes In Mechanics And Thermal Sciences. Taylor & Francis, 1980. (Cited page 244.)
- [Peraire 87] J. Peraire, M. Vahdati, K. Morgan & O.C. Zienkiewicz. *Adaptive remeshing for compressible flow computations*. Journal of Computational Physics, vol. 72, no. 2, pages 449 – 466, 1987. (Cited pages 7 and 81.)
- [Perchat 00] E. Perchat. *MINI-élément et factorisation incomplètes pour la parallélisation d'un solveur de Stokes 2D : application au forgeage*. PhD thesis, Ecole Nationale Supérieure des Mines de Paris, 2000. (Cited page 37.)
- [Peskin 77] C. S. Peskin. *Numerical analysis of blood flow in the heart*. Journal of

- Computational Physics, vol. 25, no. 3, pages 220–252, 1977. (Cited page 231.)
- [Peter 12] J. Peter, M. Nguyen-Dinh & P. Trontin. *Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates - With applications to Euler flows*. Computers & Fluids, vol. 66, no. 0, pages 194 – 214, 2012. (Cited pages 7 and 81.)
- [Picasso 03] M. Picasso. *Anisotropic error indicator based on Zienkiewicz-Zhu error estimator: Application to elliptic and parabolic problems*. Siam Journal on Scientific Computing, vol. 24, no. 4, pages 1328 – 1355, 2003. (Cited pages 7 and 81.)
- [Picasso 06] M. Picasso. *Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives*. Computer Methods in Applied Mechanics and Engineering, vol. 196, nos. 1 – 3, pages 14 – 23, 2006. (Cited pages 7 and 81.)
- [Picasso 09] M. Picasso & V. Prachittham. *An adaptive algorithm for the Crank-Nicolson scheme applied to a time-dependent convection-diffusion problem*. Journal of Computational and Applied Mathematics, vol. 233, no. 4, pages 1139 – 1154, 2009. (Cited pages 161, 214, 215, 216, 217, and 268.)
- [Picasso 11] M. Picasso, F. Alauzet, H. Borouchaki & P.L. George. *A numerical study of some hessian recovery Techniques on isotropic and anisotropic meshes*. Siam Journal On Scientific Computing, vol. 33, pages 1058 – 1076, 2011. (Cited page 101.)
- [Quan 14] D-L. Quan, T. Toulorge, E. Marchandise, J.-F. Remacle & G. Bricteux. *Anisotropic mesh adaptation with optimal convergence for finite elements using embedded geometries*. Computer Methods in Applied Mechanics and Engineering, vol. 268, pages 65 – 81, 2014. (Cited page 234.)
- [Rathfelder 94] K. Rathfelder & L. Abriola. *Mass conservation numerical solutions of the head-based Richards equation*. Water Resources Research, vol. 30, no. 9, pages 2579 – 2586, 1994. (Cited pages 8 and 190.)
- [Rausch 92] R. Rausch, J. Batina & H. Yang. *Spatial adaptation procedures on tetrahedral meshes for unsteady aerodynamic flow calculations*. AIAA Journal, vol. 30, pages 1243 – 1251, 1992. (Cited page 268.)
- [Reilly 05] M.J. O Reilly & E. O Riordan. *A Shishkin mesh for a singularly perturbed Riccati equation*. Journal of Computational and Applied Mathematics, vol. 182, no. 2, pages 372 – 387, 2005. (Cited page 161.)

- [Rodi 80] W. Rodi. *Turbulence models and their application in hydraulics - A state of the art review*. NASA STI/Recon Technical Report A, vol. 81, 1980. (Cited page 48.)
- [Rosenker 09] M. Rosenker. *NTSB Safety Recommendations A-09-17 and -18*. Rapport technique, NTSB, 2009. (Cited page 1.)
- [Rumsey 13] C Rumsey. *NASA turbulence modeling resource page*, 2013. <http://turbmodels.larc.nasa.gov/>. (Cited pages 51, 178, and 179.)
- [Russo 96] A. Russo. *Bubble stabilization of finite element methods for linearized incompressible Navier-Stokes equations*. *Comput. Methods Appl. Mech. Engrg.*, vol. 132, pages 335 – 343, 1996. (Cited pages 157 and 159.)
- [Sahin 03] M. Sahin & R. G. Owens. *A novel fully-implicit finite volume method applied to the lid-driven cavity problem. Part I: high Reynolds number flow calculations*. *International Journal for Numerical Methods in Fluids*, vol. 42, pages 57–77, 2003. (Cited page 170.)
- [Saksono 07] P. H. Saksono, W. G. Dettmer & D. Perić. *An adaptive remeshing strategy for flows with moving boundaries and fluid-structure interaction*. *International Journal for Numerical Methods in Engineering*, vol. 71, no. 9, pages 1009 – 1050, 2007. (Cited pages 267 and 268.)
- [Schäfer 96] M. Schäfer & S. Turek. *The benchmark problem - Flow around a cylinder*. In *Flow Simulation with High- Performance Computers II*. Notes on Numerical Fluid Mechanics, vol. 52, pages 547 – 566, 1996. (Cited pages xiv, 211, 212, 213, 251, and 253.)
- [Schlichting 68] H. Schlichting & J. Kestin. *Boundary-layer theory*, volume 539. McGraw-Hill New York, 1968. (Cited pages 56 and 178.)
- [Schmidt 16] Q. Schmidt. *Méthode éléments finis stabilisés pour la simulation du transfert radiatif dans les fours industriels*. thèse de doctorat, Ecole nationale supérieure des mines de Paris, 2013 - 2016. (Cited pages 54, 323, and 359.)
- [Schmitt 07] F. G. Schmitt. *About Boussinesq's turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity*. *Comptes Rendus Mécanique*, vol. 335, nos. 9 – 10, pages 617 – 627, 2007. (Cited page 48.)
- [Scovazzi 07] G. Scovazzi. *A discourse on Galilean invariance, SUPG stabilization, and the variational multiscale framework*. *Comput. Methods Appl. Mech. Engrg.*, vol. 196, nos. 4 – 6, pages 1108 – 1132, 2007. (Cited page 20.)

- [Selmin 92] V. Selmin & L. Formaggia. *Simulation of hypersonic flows on unstructured grids*. International Journal for Numerical Methods in Engineering, vol. 34, no. 2, pages 569 – 606, 1992. (Cited pages 7 and 81.)
- [Shakib 91] F. Shakib & T. J.R. Hughes. *A new finite element formulation for computational fluid dynamics: IX. Fourier analysis of space-time Galerkin/least-squares algorithms*. Computer Methods in Applied Mechanics and Engineering, vol. 87, no. 1, pages 35 – 58, 1991. (Cited pages 28 and 148.)
- [Shimada 02] K. Shimada & T. Ishihara. *Application of a modified  $k - \epsilon$  model to the prediction of aerodynamic characteristics of rectangular cross-section cylinders*. Journal of Fluids and Structures, vol. 16, no. 4, pages 465–485, 2002. (Cited page 58.)
- [Shishkin 92] G. I. Shishkin. *Discrete approximation of singularly perturbed elliptic and parabolic problems (in Russian)*. Russian Academy of Sciences, Ural Section, 1992. (Cited page 161.)
- [Shishkin 04] G. I. Shishkin. *Discrete Approximations of Solutions and Derivatives for a Singularly Perturbed Parabolic Convection-diffusion Equation*. J. Comput. Appl. Math., vol. 166, no. 1, pages 247 – 266, 2004. (Cited page 161.)
- [Sidwell 05] V. Sidwell & D. Darmofal. *The impact of blade-to-blade variability and assembly on turbine cooling performance*. ASME Journal of Turbomachinery, vol. 127, no. 4, pages 763 – 770, 2005. (Cited page 2.)
- [Siegel 02] R. Siegel & J. Howell. *Thermal radiation heat transfer*. Taylor & Francis, New-York, 2002. (Cited pages 54 and 331.)
- [Sloan 99] S. Sloan & A. Abbo. *Biot consolidation analysis with automatic time stepping and error control Part 1: theory and implementation*. International Journal for Numerical and Analytical Methods in Geomechanics, vol. 23, no. 6, pages 467–492, 1999. (Cited pages 8 and 189.)
- [Sohankar 00] Ahmed Sohankar, Lars Davidson & Christoffer Norberg. *Large eddy simulation of flow past a square cylinder: comparison of different subgrid scale models*. Journal of Fluids Engineering, vol. 122, no. 1, pages 39–47, 2000. (Cited page 58.)
- [Song 97] G. Song, T. Bjorge, J. Holen & B. F. Magnussen. *Simulation of Fluid Flow and Gaseous Radiation Heat Transfer in a Natural Gas-Fired Furnace*. International Journal of Numerical Methods for Heat and Fluid Flow, vol. 7, pages 169 – 182, 1997. (Cited page 52.)

- [Spalart 94] P. Spalart & S. Allmaras. *A one-equation turbulence model for aerodynamic flows*. La Recherche Aérospatiale, vol. 1, pages 5–25, 1994. (Cited page 51.)
- [Stanley 88] O Stanley & J. Sethian. *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*. Journal of Computational Physics, vol. 79, no. 1, pages 12 – 49, 1988. (Cited page 235.)
- [Takizawa 12] K. Takizawa, K. Schjodt, A. Puntel, N. Kostov & T. Tezduyar. *Patient-specific computer modeling of blood flow in cerebral arteries with aneurysm and stent*. Computational Mechanics, vol. 50, no. 6, pages 675 – 686, 2012. (Cited page 2.)
- [Tam 00] A. Tam, D. Ait-Ali-Yahia, M.P. Robichaud, M. Moore, V. Kozel & W.G. Habashi. *Anisotropic mesh adaptation for 3D flows on structured and unstructured grids*. Computer Methods in Applied Mechanics and Engineering, vol. 189, no. 4, pages 1205 – 1230, 2000. (Cited pages 7 and 81.)
- [Tam 03] C.K.W. Tam & K.A. Kurbatskii. *Multi-size-mesh Multi-time-step Dispersion-relation-preserving Scheme for Multiple-scales Aeroacoustics Problems*. International Journal of Computational Fluid Dynamics, vol. 17, no. 2, pages 119–132, 2003. (Cited pages 8 and 188.)
- [Temam 77] Roger Temam. *Navier Stokes Equations*. Amsterdam-New York-Oxford, 1977. (Cited page 36.)
- [Tezduyar 86] T.E. Tezduyar & Y.J. Park. *Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations*. Computer Methods in Applied Mechanics and Engineering, vol. 59, no. 3, pages 307 – 325, 1986. (Cited pages 29 and 30.)
- [Tezduyar 92] S Tezduyar T.E. and. Mittal, S.E. Ray & R. Shih. *Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements*. Computer Methods in Applied Mechanics and Engineering, vol. 95, no. 2, pages 221 – 242, 1992. (Cited page 37.)
- [Tezduyar 00] T. E. Tezduyar & Y. Osawa. *Finite element stabilization parameters computed from element matrices and vectors*. Comput. Methods Appl. Mech. Engrg., vol. 190, pages 411–430, 2000. (Cited pages 43, 45, 148, and 150.)
- [Tezduyar 06] T.E. Tezduyar, S. Sathe, R. Keedy & K. Stein. *Space-time finite element techniques for computation of fluid-structure interactions*. Com-

- puter Methods in Applied Mechanics and Engineering, vol. 195, nos. 17 – 18, pages 2002 – 2027, 2006. (Cited page 193.)
- [Thangam 91] S. Thangam & N. Hur. *A highly-resolved numerical study of turbulent separated flow past a backward-facing step*. International Journal of Engineering Science, vol. 29, no. 5, pages 607 – 615, 1991. (Cited page 50.)
- [Tremel 07] U. Tremel, K. A. Sorensen, S. Hitzel, H. Rieger, Oubay Hassan & Nigel P. Weatherill. *Parallel remeshing of unstructured volume grids for CFD applications*. International Journal for Numerical Methods in Fluids, vol. 53, no. 8, pages 1361 – 1379, 2007. (Cited page 7.)
- [Valette 09] V. Valette, T. Coupez, C. David & B. Vergnes. *A Direct 3D Numerical Simulation Code for Extrusion and Mixing Processes*. International Polymer Processing XXIV, vol. 2, pages 141 – 147, 2009. (Cited pages 3, 53, 232, and 234.)
- [van der Pijl 05] S.P. van der Pijl, A. Segal, C. Vuik & P. Wesseling. *A mass-conserving Level-Set method for modelling of multi-phase flows*. International Journal for Numerical Methods in Fluids, vol. 47, pages 339–361, 2005. (Cited page 244.)
- [Van Maele 06] K. Van Maele & B. Merci. *Application of two buoyancy-modified - turbulence models to different types of buoyant plumes*. Fire Safety Journal, vol. 41, no. 2, pages 122 – 138, 2006. (Cited page 49.)
- [Venditti 02] D.A. Venditti & D. L. Darmofal. *Grid adaptation for functional outputs: application to two-dimensional inviscid flows*. J. COMPUT. PHYS, vol. 176, pages 40 – 69, 2002. (Cited page 250.)
- [Venditti 03] D.A. Venditti & D. L. Darmofal. *Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows*. Journal of Computational Physics, vol. 187, no. 1, pages 22 – 46, 2003. (Cited pages 7, 81, 96, and 250.)
- [Veysset 14a] J. Veysset. *Immersed nurbs methods for solving conjugated heat and turbulent flow problems inside large domains*. thèse de doctorat, Ecole nationale supérieure des mines de Paris, 2014. (Cited page 219.)
- [Veysset 14b] J. Veysset, G. Jannoun, E. Hachem & T. Coupez. *Immersed NURBS for CFD applications*. accepted in SEMA SIMAI Springer Series, 2014. (Cited page 238.)
- [Volker 01] J. Volker & G. Matthies. *Higher-order finite element discretizations in a benchmark problem for incompressible flows*. International Journal

- for Numerical Methods in Fluids, vol. 37, no. 8, 2001. (Cited pages 211, 212, and 251.)
- [Volker 04a] J. Volker. *Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder*. International Journal for Numerical Methods in Fluids, vol. 44, no. 7, pages 777 – 788, 2004. (Cited pages 56 and 57.)
- [Volker 04b] J. Volker. *Reference values for drag and lift of a two-dimensional time-dependent flow around a cylinder*. International Journal for Numerical Methods in Fluids, vol. 44, no. 7, pages 777 – 788, 2004. (Cited pages 211, 212, 251, and 253.)
- [Volker 07] J. Volker & Petr K. *On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations: Part I - A review*. Computer Methods in Applied Mechanics and Engineering, vol. 196, nos. 17 – 20, pages 2197 – 2215, 2007. (Cited page 29.)
- [Von Schéele 10] J. Von Schéele. *Oxyfuel Combustion in the Steel Industry: Energy Efficiency and Decrease of CO<sub>2</sub> Emissions*. Energy Efficiency, 2010. (Cited page 321.)
- [Wall 00] W. A. Wall, M. Bischoff & E. Ramm. *A deformation dependent stabilization technique, exemplified by EAS elements at large strains*. Comput. Methods Appl. Mech. Engrg., vol. 188, no. 4, pages 859–871, 2000. (Cited page 43.)
- [Was 14] *Waste Heat Reduction and Recovery for Improving Furnace Efficiency, Productivity and Emissions Performance*. Rapport technique, A Sourcebook for Industry is a development of the Best Practices initiative under the U.S. Department of Energy (DOE) Industrial Technologies Program (ITP) and the Industrial Heating Equipment Association (IHEA), 2014. (Cited pages xviii and 322.)
- [Xenophontos 03] C. Xenophontos. *A note on the convergence rate of the finite element method for singularly perturbed problems using the Shishkin mesh*. Applied Mathematics and Computation, vol. 142, nos. 2 – 3, pages 545 – 559, 2003. (Cited page 161.)
- [Yano 12] M. Yano & D. Darmofal. *An optimization-based framework for anisotropic simplex mesh adaptation*. Journal of Computational Physics, vol. 231, no. 22, pages 7626 – 7649, 2012. (Cited pages 7 and 81.)
- [Yerry 83] Mark A Yerry & Mark S Shephard. *A modified quadtree approach to finite element mesh generation*. Computer Graphics and Applications, IEEE, vol. 3, no. 1, pages 39 – 46, 1983. (Cited page 75.)

- [Zaragoci 13] J-F Zaragoci, E. Hachem, G. Jannoun, J. Veysset & T. Coupez. *Méthode d'immersion de pièces mobiles dans des fours industriels*. In 11e Colloque National en Calcul des Structures, CSMA, 2013.
- [Zhang 03] Z. Zhang. *Finite element superconvergence on shishkin mesh for 2D convection-diffusion problems*. *Mathematics of Computation*, vol. 72, pages 1147 – 1177, 2003. (Cited page [162](#).)
- [Zhang 04] Z. Zhang. *Polynomial preserving recovery for anisotropic and irregular grids*. *Journal of Computational Mathematics*, vol. 22, pages 331 – 340, 2004. (Cited page [102](#).)
- [Zhang 05] Z. Zhang & A. Naga. *A New Finite Element Gradient Recovery Method: Superconvergence Property*. *SIAM J. Sci. Comput.*, vol. 26, no. 4, pages 1192 – 1213, 2005. (Cited page [102](#).)
- [Zhou 97] G. Zhou. *How accurate is the streamline diffusion finite element method?* *Math. Comp*, vol. 66, pages 31 – 44, 1997. (Cited page [27](#).)
- [Zienkiewicz 87] O. C. Zienkiewicz & J. Z. Zhu. *A simple error estimator and adaptive procedure for practical engineering analysis*. *International Journal for Numerical Methods in Engineering*, vol. 24, no. 2, pages 337 – 357, 1987. (Cited pages [95](#), [96](#), and [268](#).)
- [Zienkiewicz 92] O. C. Zienkiewicz & J. Z. Zhu. *The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique*. *International Journal for Numerical Methods in Engineering*, vol. 33, no. 7, pages 1331 – 1364, 1992. (Cited pages [95](#), [96](#), [100](#), and [268](#).)
- [Zienkiewicz 94] O. C. Zienkiewicz & J. Wu. *Automatic directional refinement in adaptive analysis of compressible flows*. *International Journal for Numerical Methods in Engineering*, vol. 37, no. 13, pages 2189 – 2210, 1994. (Cited pages [7](#) and [81](#).)
- [Zohdi 07] T. Zohdi. *An adaptive-recursive staggering strategy for simulating multifield coupled processes in microheterogeneous solids*. *International Journal for Numerical Methods in Engineering*, vol. 53, pages 1511 – 1532, 2007. (Cited pages [8](#), [9](#), [190](#), [191](#), and [272](#).)



*... and it will never end!*





## Adaptation anisotrope précise en espace-temps et méthodes d'éléments finis stabilisées pour la résolution de problèmes de mécanique des fluides instationnaires

**RESUMÉ:** Aujourd'hui, avec l'amélioration des puissances de calcul informatique, la simulation numérique est devenue un outil essentiel pour la prédiction des phénomènes physiques et l'optimisation des procédés industriels. La modélisation de ces phénomènes pose des difficultés scientifiques car leur résolution implique des temps de calcul très longs malgré l'utilisation d'importantes ressources informatiques. Dans cette thèse, nous nous intéressons à la résolution de problèmes complexes couplant écoulements et transferts thermiques. Les problèmes physiques étant fortement anisotropes, il est nécessaire d'avoir un maillage avec une résolution très élevée pour obtenir un bon niveau de précision. Cela implique de longs temps de calcul. Ainsi il faut trouver un compromis entre précision et efficacité. Le développement de méthodes d'adaptation en temps et en espace est motivé par la volonté de mettre en place des applications réelles et de limiter les inconvénients inhérents aux méthodes de résolution non adaptatives en terme de précision et d'efficacité. La résolution de problèmes multi-échelles instationnaires sur un maillage uniforme avec un nombre de degrés de liberté limité est souvent incapable de capturer les petites échelles, nécessite des temps de calcul longs et peut aboutir à des résultats incorrects. Ces difficultés ont motivé le développement de méthodes de raffinement local avec une meilleure précision aux endroits adéquats. L'adaptation en temps et en espace peut donc être considérée comme une composante essentielle de ces méthodes. L'approche choisie dans cette thèse consiste en l'utilisation de méthodes éléments finis stabilisées et le développement d'outils d'adaptation espace-temps pour améliorer la précision et l'efficacité des simulations numériques. La dérivation de la méthode adaptative est basé sur un estimateur d'erreur sur les arrêtes du maillage afin de localiser les régions du domaine de calcul présentant de forts gradients ainsi que les couches limites. Ensuite une métrique décrivant la taille de maille en chaque noeud dans les différentes directions est calculée. Afin d'améliorer l'efficacité des calculs, la construction de cette métrique prend en compte un nombre fixe de noeuds et aboutit à une répartition et une orientation optimale des éléments du maillage. Cette approche est étendue à une formulation espace-temps où les maillages et les pas de temps optimaux sont prédits sur des intervalles de temps en vue de contrôler l'erreur d'interpolation sur le domaine de calcul.

**Mots clés:** Adaptation de maillage anisotrope, Adaptation de temps, Méthodes éléments finis stabilisées, Résolution de transferts thermiques, Résolution de écoulements turbulents, Fours industriels

## Space-Time accurate anisotropic adaptation and stabilized finite element methods for the resolution of unsteady computational fluid dynamics problems

**ABSTRACT:** Nowadays, with the increase in computational power, numerical modeling has become an intrinsic tool for predicting physical phenomena and developing engineering designs. The modeling of these phenomena poses scientific complexities the resolution of which requires considerable computational resources and long lasting calculations. In this thesis, we are interested in the resolution of complex long time and large scale heat transfer and fluid flow problems. When the physical phenomena exhibit sharp anisotropic features, a good level of accuracy requires a high mesh resolution, hence hindering the efficiency of the simulation. Therefore a compromise between accuracy and efficiency shall be adopted. The development of space and time adaptive techniques was motivated by the desire to devise realistic configurations and to limit the shortcomings of the traditional non-adaptive resolutions in terms of lack of solution's accuracy and computational efficiency. Indeed, the resolution of unsteady problems with multi-scale features on a prescribed uniform mesh with a limited number of degrees of freedom often fails to capture the fine scale physical features, have excessive computational cost and might produce incorrect results. These difficulties brought forth investigations towards generating meshes with local refinements where a higher resolution was needed. Space and time adaptations can thus be regarded as essential ingredients in this recipe. The approach followed in this work consists in applying stabilized finite element methods and the development of space and time adaptive tools to enhance the accuracy and efficiency of the numerical simulations. The derivation process starts with an edge-based error estimation for locating the regions, in the computational domain, presenting sharp gradients, inner and boundary layers. This is followed by the construction of nodal metric tensors that prescribe, at each node in the spatial mesh, mesh sizes and the directions along which these sizes are to be imposed. In order to improve the efficiency of computations, this construction takes into account a fixed number of nodes and generates an optimal distribution and orientation of the mesh elements. The approach is extended to a space-time adaptation framework, whereby optimal meshes and time-step sizes for slabs of time are constructed in the view of controlling the global interpolation error over the computation domain.

**Keywords:** Anisotropic mesh adaptation, Time Adaptation, Stabilized finite element methods, Resolution of heat transfers, Resolution of turbulent flows, Industrial furnaces