UCLouvain

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ÉCOLE POLYTECHNIQUE DE LOUVAIN
INSTITUTE OF MECHANICS,
MATERIALS AND CIVIL ENGINEERING

COUPLING A VORTEX PARTICLE-MESH METHOD FOR COMPRESSIBLE FLOWS WITH A DISCONTINUOUS GALERKIN SOLVER

DOCTORAL DISSERTATION PRESENTED BY

PHILIPPE PARMENTIER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF DOCTOR IN ENGINEERING SCIENCES

THESIS COMMITTEE:

Prof. Philippe Chatelain, Université catholique de Louvain (Advisor)
Prof. Grégoire Winckelmans, Université catholique de Louvain (Advisor)
Prof. Jeff D. Eldredge, University of California, Los Angeles
Dr. Koen Hillewaert, Cenaero
Prof. Jean-François Remacle, Université catholique de Louvain
Prof. Renaud Ronsse, Université catholique de Louvain (President)

Louvain-la-Neuve, November 2018

ACKNOWLEDGEMENTS

L'adage selon lequel l'important n'est pas la destination mais le voyage lui-même revêt tout son sens dans la réalisation d'une thèse de doctorat. Ce voyage fut à la fois extrêmement enrichissant d'un point de vue scientifique et professionnel mais également d'un point de vue personnel. Celui-ci n'aurait pu avoir lieu sans le support inconditionnel de mes promoteurs, Grégoire Winckelmans et Philippe Chatelain. Je les remercie de m'avoir permis de réaliser les travaux présentés dans ce manuscript et de m'avoir guidé et conseillé durant toute la durée de ce doctorat. Leur rigueur et leur complémentarité auront été des atouts indéniables pour le succès de ce projet. Je remercie également les membres de mon jury, Philippe Chatelain, Grégoire Winckelmans, Jeff D. Eldredge, Koen Hillewaert, Jean-François Remacle et Renaud Ronsse, pour leur examen minutieux de ce document ainsi que pour leurs commentaires et remarques avisés lors de la défense privée.

Je remercie Jeff D. Eldredge d'avoir partagé avec moi son expérience en matière de développement de méthode vortex pour l'étude des écoulements compressibles, ce qui constitue indubitablement une niche dans une niche. Je remercie Koen Hillewaert, Jean-Sébastien Cagnone et Pierre Schrooyen de m'avoir initié aux méthodes DG et pour leurs conseils éclairés lors du développement du couplage avec le code Argo développé à Cenaero. J'en profite pour remercier Cenaero pour cette collaboration fructueuse.

Je remercie le F.R.S.-FNRS pour l'octroi d'une bourse FRIA qui aura financé 4 années de recherches. Les résultats présentés dans ce manuscript ont été obtenus grâce aux ressources de calcul fournies par les installations de calcul intensif de l'Université catholique de Louvain (CISM) ainsi que celles du Consortium des Equipements de Calcul Intensif en Fédération Wallonie Bruxelles (CECI), également financé par le Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) sous la convention 2.5020.11. Les présents travaux ont également bénéficié de moyens de calcul mis à disposition sur le supercalcu-

lateur Tier-1 de la Fédération Wallonie-Bruxelles, infrastructure financée par la Région wallonne sous la convention $n^{\circ}1117545$.

Durant cette thèse, j'ai eu l'occasion d'évoluer aux côtés de personnes d'une rare intelligence, dans une ambiance décontractée et bienveillante qui définit si bien le pôle TFL. Je remercie tout particulièrement les Furieux, Olivier Thiry (OT), Benjamin Berger (BB) et Pierre Schrooyen (PBS) ainsi que les Jeunes Furieux qui leur ont succédé et en particulier Olivier Lamberts, Thomas (TG) et Denis-Gabriel (but you may call him deegee or DG). Je tiens à tous les remercier pour les nombreuses discussions sérieuses et moins sérieuses mais aussi pour les tranches de rire qui auront égayé mes années au TFL! Je remercie également les personnes qui m'auront permis de garder un esprit sain dans un corps sain, les infatigables de la running team du TFL: Hervé Jeanmart (Lalalala lalalalaaaa), Véronique, Maxime ainsi qu'Elise et Maud. Au-delà de l'aspect sportif, nos sorties dans les bois environnants auront aussi été l'occasion de partager de très bons moments. J'aimerais mettre ici en avant Olivier Thiry, Benjamin, Maud, TG et DG, sans qui cette aventure n'aurait pas eu la même saveur! Je tiens à remercier Isabelle Hennau, dont le travail de l'ombre est reconnu de tous, ainsi que Frank, Lili et Julien. Je tiens finalement à remercier Matthieu Duponcheel, le joker du TFL. Mes remerciements vont également à Yves qui, à bien des égards, aura positivement influé sur le cours de ma thèse. Je remercie aussi Ariane, Sophia, Caro, Gauthier, Nini, Victor, Billu, Michele, Will, Constantin et la nouvelle génération, Denis, Ignace, Maxime et Marion!

Je tiens également à remercier mes amis, ainsi que mes parents et mon frère pour leur soutien inébranlable. Je n'ai pas ici la possibilité de remercier nommément toutes les personnes qui m'auront inspiré, motivé ou amusé mais elles sont nombreuses! Merci à vous!

Au fil du temps, j'ai pris l'habitude d'également féliciter les compagnons ou les compagnes de mes collègues docteurs à l'issue de leur soutenance publique. En effet, ceux-ci prennent souvent sur leurs épaules une partie du poids que peut parfois représenter une thèse, que ce soit en temps ou en énergie, et notre travail est finalement un peu le leur également. Mes remerciements les plus chaleureux vont donc à Claudia pour son soutien indéfectible, pour sa capacité à me faire rire et sourire en toute occasion et finalement, parce que, sans en avoir l'air, elle a compris comment faire ressortir le meilleur de moi-même.

CONTENTS

Contents **i**

1 Introduction **1**
1.1 Motivation 1
1.2 Overview 6
1.3 General approach 9
1.3.1 Governing equations 9
1.3.2 Coupling strategy 10

2 A Vortex Particle-Mesh method for subsonic compressible flows **15**
2.1 Introduction 16
2.2 Methodology 20
2.2.1 Governing equations 20
2.2.2 Vortex Particle-Mesh method 22
2.2.3 Vortex Particle-Mesh method for unbounded compressible flows 24
2.2.4 Remeshing and Filtering 27
2.3 Numerical tests 31
2.3.1 Radiation of an acoustic pulse 32
2.3.2 Co-rotating vortex pair 33
2.4 Conclusions 48

3 A VPM solver with a Brinkman penalization for compressible flows **51**
3.1 Introduction 51
3.2 Methodology 54
3.2.1 Vortex Particle-Mesh method and Brinkman penalization for incompressible flows 54
3.2.2 Vortex Particle-Mesh method and Brinkman penalization for compressible flows 56
3.3 Numerical tests 62

3.3.1 Impulsively started and incompressible flow past a cylinder at low Reynolds number (Re = 550) 62
3.3.2 2-D cylindrical acoustic benchmark problem 65
3.4 Conclusions 74

4 Coupling a VPM solver and an Eulerian DG solver for compressible flows 77
4.1 Introduction 77
4.2 Discontinuous Galerkin solver 81
4.3 Coupling boundary conditions for the Eulerian solver 88
4.4 Correction of the Lagrangian solution in the Eulerian subdomain 93
4.4.1 Computation of the correction using quadrature rules on the Lagrangian particles 94
4.4.2 Computation of the correction using a quadrature on the Eulerian elements 98
4.4.3 Blending of the correction and of the current Lagrangian solution 107
4.5 Additional features 109
4.5.1 Sponge layer 109
4.5.2 Force computation 111
4.6 Conclusions 112

5 Validation of the coupling methodology 115
5.1 Introduction 115
5.2 Co-rotating vortex pair 117
5.3 2-D cylindrical acoustic benchmark problem 121
5.3.1 Wide Eulerian subdomain 122
5.3.2 Narrow Eulerian subdomain 126
5.4 Flow past a cylinder at Re = 100 and M = 0.1 137
5.4.1 Simulation using the DG solver alone 137
5.4.2 Simulation using the DG-VPM solver 142
5.5 Flow past a cylinder at Re = 550 and M = 0.1 155
5.6 Conclusions 158

6 Conclusions and perspectives 163
6.1 Conclusions 163
6.2 Perspectives 169

A A 2-D incompressible VPM method applied to the lid-driven cavity 173

B Second and fourth order "isotropic" finite differences schemes 179

C Numerical dispersion and dissipation errors of a 1-D redistributed and filtered Lagrangian method **189**

C.1 Contribution of the filtering to the numerical dissipation and dispersion errors 190

C.2 Performance of the E_6 scheme and of the combination M_6^* scheme - discrete filter 194

D Analytical expression for the divergence of the velocity field for the acoustic pulse problem **199**

E Enforcement of global conservation on the correction **203**

Bibliography **207**

CHAPTER 1

INTRODUCTION

1.1 Motivation

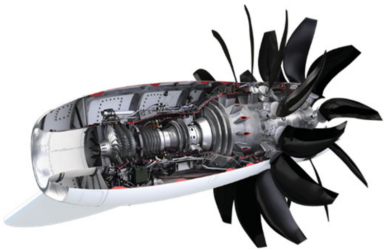
Wakes formed downstream of complex shaped bodies always spurred research because of the crucial issues that are at stake such as the safety and the community noise reduction in the case of an aircraft wake. Several projects have been dedicated to the study of aircraft wakes in order to fully uncover and control the impact of aircraft wake turbulence. Such projects include AWIATOR [45], Far-Wake [96], WakeNet3-Europe [75], GreenWake, etc., all co-financed by the European Commission Framework Programmes. As stated in [96], the large-scale vortices that develop behind an aircraft represent a significant hazard for a following aircraft, even more so during landing and take-off phases. In terms of air traffic management, this represents a challenge both in terms of operation safety and operational efficiency. Air traffic growth contributes to the congestion in airports and requires creative solutions such as more precise or even dynamic separation minima between two aircraft while maintaining operation safety [75]. Besides, wake encounters may also occur mid-flight [76]. They are referred to as *En Route wake vortex encounters* and might affect the passenger's comfort in the best-case scenario. During that flight regime, modern commercial airplanes are operating at a flight speed such that compressibility effects are important in the close vicinity of the airplane while the wake is, to a large extent, an essentially incompressible flow. If

one wants to fully uncover the mechanisms at play in the near and far wakes, a good understanding of the near-wall physics is also important.

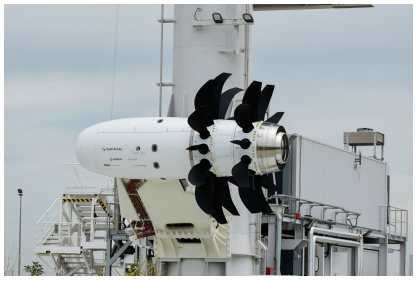
Aircraft operations near airports constitute one of the major sources of *community noise* in these areas. According to a report prepared for the World Health Organization [12], the associated adverse effects are numerous and includes effects on sleep, cardiovascular and psychophysiological systems as well as effects on performance and productivity, to name but a few. As also stated in [12], the main source of noise of a jet aircraft is the turbulence generated by the mixing layer between the high-velocity jet exhaust and the surrounding air. This jet noise on modern high-bypass turbofan engines is partially mitigated by the use of a fan, which surrounds the high velocity jet by a low velocity airflow.

In an effort to reduce fuel consumption, both for environmental and economic reasons, the Counter-Rotating Open Rotor (CROR) is again seen as a potentially relevant alternative to modern high-bypass turbofan engines. A first effort to develop such a technology was conducted by the NASA and the US industry in the late 1970s and 1980s [66], but was undermined by the return of the oil price to its level prior to the 1973 oil crisis as explained in [148, 157]. In Europe, the effort is made within the framework of the Clean Sky 1 and the ongoing Clean Sky 2 EU-funded projects [145]. In that context, Safran Aircraft Engines performed the first ground test of its demonstrator engine in May 2017 (Fig. 1.1). This concept takes to an extreme the idea behind the high-bypass technology by operating unshrouded counter rotating blades. The unshrouded blades installed on slotted discs in close proximity have an important noise signature, which originates from the complex blade interactions and from the fact that the source of noise is not shielded by a nacelle [145]. One of the challenges is thus to keep the aircraft interior noise at an acceptable level. Here again, the community noise during landing and take-off phases also constitutes a major concern. The increased noise, blade and hub failure as well as the interaction of the flow with the supporting airframe structure in the conventional engine-to-wing configuration forced the designers to look into innovative integration [66, 43, 148]. Besides the performance of the blades and the engine itself, the engine-airframe integration comes with its share of challenges and open questions.

In the context of rotorcraft aerodynamics, compressibility effects are important in the vicinity of the blades due to the high rotation speed, similarly



(a) Contra-rotating open rotor (Safran Aircraft Engines). Image taken from [17].



(b) Open-air test rig located in Istres (France) with the geared Open Rotor demonstrator. Credit: Safran Aircraft Engines.

Figure 1.1: Geared pusher Open Rotor engine developed in the Clean Sky project [145].

to the open rotor system considered above. In the case of helicopter blades, the weakly compressible interaction between the vortex generated at the tip of a blade (tip vortex) with the following blade, as illustrated in Figure 1.2, can give rise to substantial unsteady loadings and is involved in the generation of sound and vibrations [124]. This phenomenon is known as the parallel blade-vortex interactions (BVIs) and may impair the helicopter performance [124]. As stated by Peng and Gregory [124] and in previous studies, the accurate capture of the vortex dynamics during close interactions is essential to comprehend the fundamental interaction mechanisms coming into play in BVIs. The mitigation of the undesired effects of blade-vortex interactions still constitutes a very active research area.

The objective of this thesis is to develop a numerical approach allowing the realistic and efficient simulation of compressible flows past bodies such as the systems considered above. The resulting numerical method should then be able to simulate complex compressible and viscous flows in the subsonic regime (possibly also with some transonic region) from the shaped system to far downstream in the wake. To achieve this goal, the advantages of two different numerical methods introduced hereunder will be combined.

Vortex Particle (VP) and Vortex Particle-Mesh (VPM) methods are well suited to accurately capture advection-dominated flows. This is mainly attributable to the Lagrangian treatment of the advection, which results in relaxed stability constraints and in a reduced dispersion error. They are also capable of imposing unbounded domain boundary conditions. As a conse-

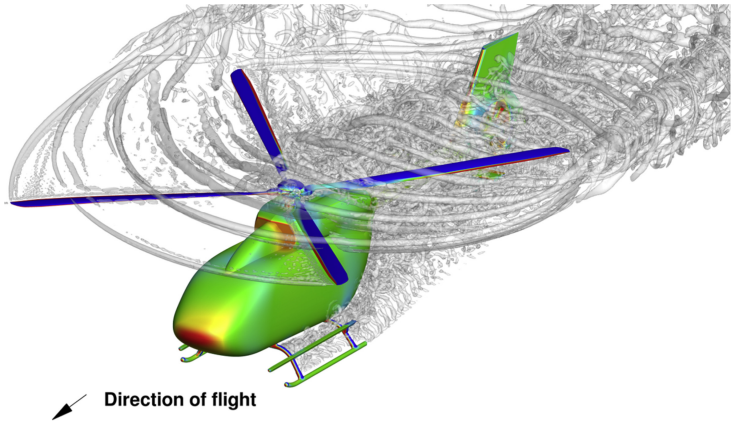


Figure 1.2: Vortex visualization around an Airbus H145 helicopter in descent flight exhibiting blade vortex interactions. The surfaces are colored by the pressure contours. Image reproduced from [89].

quence, VP methods are excellent candidates to capture the turbulent vortical structures that develop in the near and far wake of a solid obstacle. This capability, as well as its computational efficiency, has been demonstrated in numerous studies [159, 27, 135, 26, 8], also using multi-resolution techniques [44, 100, 136]. It was also recently demonstrated in [21]: Figure 1.3 shows a Large Eddy Simulation of the wake (up to 30 diameters) behind a simplified helicopter rotor as obtained using a VPM solver for incompressible flows coupled with a simple model to account for the blades [7]. The VPM solver was, furthermore, coupled to a multi-body solver [46] to account for a rigid flapping of the blades. The simulation illustrated in Fig. 1.3 demonstrates the capability of VPM solvers to accurately capture small scale vortical structures, and thus also turbulence, over large distances.

Through the years, several methodologies allowing a VP or a VPM method to take into account a solid obstacle have been developed. These methodologies vary in range from the least computationally invasive *panel-based methods* [88, 128, 129] and *penalization* [36, 135, 132, 56, 67, 105, 59] to the more invasive *immersed interface methods* [103, 104, 60]. Regardless of the methodology, VPM methods do not easily account for complex flow dynamics in the near-wall region. The complex near-wall phenomena are associated to the characteristics of the boundary layer (laminar, transitional or turbulent) and its behavior (transition from a laminar to a turbulent boundary layer, laminar separation possibly followed by a turbulent reattachment, turbulent separation, etc.). All of these complex flow features are strongly anisotropic whereas

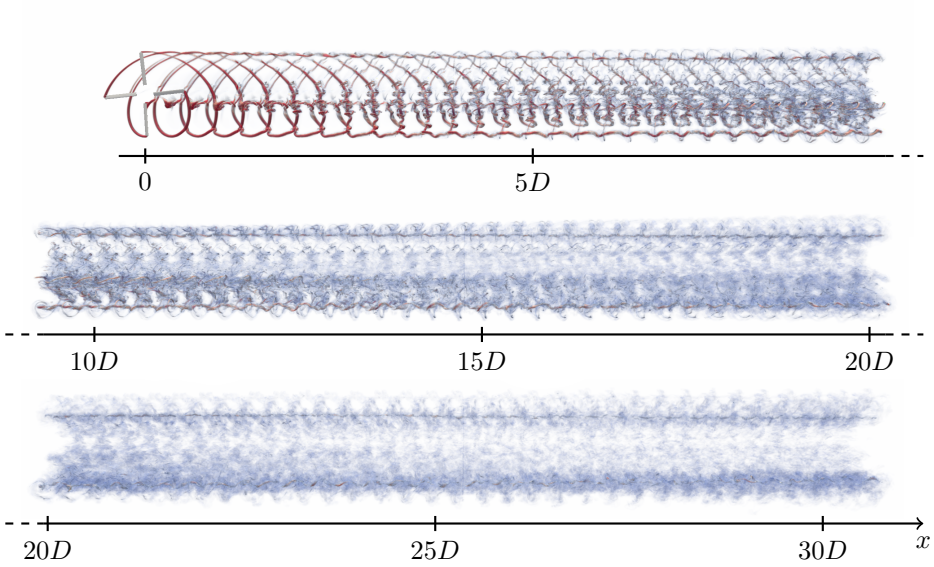


Figure 1.3: Near to far wake of an advancing simple rotor model, from the initial vortex generation to the roll-up and the establishment of a mainly turbulent wake [21]. This Large Eddy Simulation was performed using a 3-D VPM solver for incompressible flows using a sub-grid scale model as implemented in [31] together with an Immersed Lifting line method [7] to account for the blades. The VPM solver was also coupled to a multi-body solver [46] to account for a rigid flapping motion of the blades. Image reproduced from [21].

the remeshed VP and VPM methods rely on an underlying isotropic grid (for the necessary remeshing operation but also for the efficient evaluation of different quantities on the grid in a VPM solver). One of the possible strategies to improve the quality of the solution in the near-wall region is to use a second type of solver that does not exhibit such shortcoming. Eulerian solvers are precisely capable of using anisotropic computational elements. Moreover, they are intrinsically adapted to be used in conjunction with a body-fitted approach, which allows a straightforward implementation of the wall boundary conditions. However, using an Eulerian solver to solve complex turbulent flows over large distance, as in the example above, is expensive. An approach combining an Eulerian solver to capture the complex near-wall dynamics and a Lagrangian solver to handle the flow dynamics in the wake of the obstacle would allow to make the most of both methods. This approach requires what is generally referred to as a *coupling* of the solvers. Ideally, the Eulerian subdomain should be limited to the narrow region surrounding the obstacle where anisotropic flow structures / strong gradients are found. In practice,

this region will be larger than the near-wall region and will be referred to as the *near-body* region.

The coupling of these two approaches would allow studying, more efficiently and more accurately, problems in which one needs to capture complex anisotropic near-wall phenomena as well as the complex vortex dynamics far in the wake. It would then prove useful for the development and integration of the systems considered in this introduction; wings, CROR and helicopter blades. As an example, this work paves the way to a hybrid Eulerian-Lagrangian solver capable of simulating the wake shown in Fig. 1.3 while better accounting for the complex flow dynamics in the near-wall region, i.e the anisotropic flow structures and the compressibility effects. The purpose of this coupling methodology is thus different from the one pursued when combining a CFD (Computational Fluid Dynamics) code, such as those described above, with a CAA (Computational Aeroacoustics) solver in order to propagate the aerodynamically generated sound to the far-field: in such a coupling, the CFD code is used to solve the relevant flow structures in order to provide the acoustic source terms to the CAA solver, which is responsible for the sound propagation.

1.2 Overview

The developments leading to the coupling methodology will be progressively introduced in this manuscript.

The existing VPM solvers are mainly designed to study incompressible flows, whereas the above examples have at least a region of compressible flow near the wall and in a part of the near field as well. As a consequence, the first main objective of this work will be to develop a VPM method for compressible flows. This will allow dealing with viscous and unsteady compressible flows. The first two chapters will be devoted to this effort:

- **Chapter** 2 presents the elements allowing to extend a VPM method for incompressible flows to the compressible regime. It relies on existing and proven techniques for incompressible flows when possible, and adapt the VP method for compressible flows developed in [50] to benefit from the underlying grid, which allows to efficiently perform a series of operations. The developments will lead to a solver capable of performing accurate simulations of viscous and unsteady compressible flows in an unbounded domain. It will be here restricted to the subsonic regime as

no shock capturing methodology is implemented, yet. The method will also be applied to aeroacoustics problems. The method will be validated against analytical solutions when available, and against reference results from the literature. This work was published in the *Journal of Computational Physics* [120].

- **Chapter** 3 will allow the VPM solver for compressible flows to take solid obstacles into account. The Brinkman penalization technique for compressible flows introduced in [99] will here be adapted to fit in the VPM method. Two different implementations will be presented. Benchmark test cases will be performed and the results will be compared to analytical solutions and reference results from the literature. The addition of such a capability is by itself interesting for the VPM solver of Chapter 2 used alone but it also proved useful in the coupling methodology, even if the Eulerian solver is in charge of the accurate resolution of the near-body region.

Then, to be able to study flows at higher Reynolds number and eventually also transonic flows, and thus extend the application range of the code, the second main objective will be to make the most of two different worlds by coupling the newly developed VPM solver for compressible flows with an Eulerian solver. We here elected to use the high order solver, named Argo, developed at Cenaero (Gosselies, Belgium). Argo relies on a high order Discontinuous Galerkin (DG) method and is being developed to maintain a high computational efficiency and a high versatility. It is a multi-physics CFD code, which has already been applied to a large variety of challenging problems ranging from aeroelastic, (aero)acoustics [71] and aerothermal problems (also with ablation) [139, 140], to welding [19] and additive manufacturing [138, 137]. In this work, the developments will only rely on the module for compressible flows of Argo. More specifically,

- **Chapter** 4 introduces the coupling methodology. It will rely on existing techniques for incompressible flows when possible. Indeed, the idea already received some attention for this type of solver [37, 65, 113, 44, 100, 102] and it still constitutes an active research area. Only a few contributions exist for compressible solvers [114, 147, 118, 116]. A review of the existing techniques will be provided and the method introduced in this manuscript will be motivated. The objectives of this chapter are twofold. The first objective is to succinctly present the near-body Eulerian solver; the formulation used in the solver, as well as the temporal and spatial discretizations, indeed need to be taken into account in the

coupling methodology. This presentation will allow identifying its requirements but also the capabilities that may benefit the coupled solver. The second objective will be to introduce the implementation of the different ingredients that are necessary in each solver to obtain a robust hybrid solver.

- **Chapter** 5 will be devoted to the validation of the resulting hybrid DG-VPM solver. It will be applied to a series of test cases, some of them already considered in Chapter 2 and Chapter 3. Here again, the results will be compared to analytical and reference results. Improvements brought by the accurate resolution of the near-body region by the Eulerian solver will be demonstrated and discussed. The limitations of the solver will also be discussed.

The developments and results obtained during this doctoral thesis were communicated to the scientific community through oral presentations in different international conferences [121, 122, 119, 161] and through a journal paper [120].

The next section quickly lays the foundations of the coupling methodology. They will motivate and support the developments made in the different chapters of this manuscript. Indeed, although the developments made in Chapter 2 and Chapter 3 for the VPM for compressible flow form an independent effort, some aspects will also find motivations from the coupling methodology.

1.3 General approach

1.3.1 Governing equations

Most of the phenomena mentioned here above are governed by the Navier-Stokes equations for compressible flows, which read as the following system of PDEs

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{C}} + \nabla \cdot \boldsymbol{\mathcal{D}} + \boldsymbol{\mathcal{S}} = 0 \tag{1.1}$$

with $\mathbf{U}$ the conservative variables, $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{D}}$ the convective and diffusive fluxes respectively, and $\boldsymbol{\mathcal{S}}$ a source term

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix} \quad , \quad \boldsymbol{\mathcal{C}}(\mathbf{U}) = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}\mathbf{u} + p\mathbf{I} \\ \rho H\mathbf{u} \end{pmatrix},$$

$$\boldsymbol{\mathcal{D}}(\mathbf{U}, \nabla\mathbf{U}) = \begin{pmatrix} 0 \\ -\tau \\ -\tau \cdot \mathbf{u} + \mathbf{q} \end{pmatrix} \quad , \quad \boldsymbol{\mathcal{S}}(\mathbf{U}) = \begin{pmatrix} 0 \\ -\rho\mathbf{f} \\ -\rho\mathbf{f} \cdot \mathbf{u} \end{pmatrix}$$

with ρ the density of the fluid, $\mathbf{u}$ the velocity vector, E the specific total internal energy, p the pressure, H the specific total enthalpy, and $\mathbf{I}$ the identity tensor. Newtonian fluids are considered and the Stokes hypothesis is assumed (thus neglecting the bulk viscosity). The viscous stress tensor τ is then defined as:

$$\tau = 2\mu(T)\left(\mathbf{S} - \frac{1}{3}\left(\text{tr}\,\mathbf{S}\right)\mathbf{I}\right) \tag{1.2}$$

with $\mathbf{S} = \frac{1}{2}\left((\nabla\mathbf{u}) + (\nabla\mathbf{u})^T\right)$ the rate-of-strain tensor and $\mu(T)$ the dynamic viscosity. The viscous stress tensor is the deviatoric part of the stress tensor σ

$$\sigma = -p\mathbf{I} + \tau. \tag{1.3}$$

The pressure, the density and the temperature are linked by the perfect gas law

$$p = \rho R_s T \tag{1.4}$$

with $R_s = c_p - c_v$ the specific gas constant, and c_p and c_v the specific heats at constant pressure and constant volume, respectively; their ratio being $\gamma = c_p/c_v$. The specific total energy E is obtained as the sum of the specific internal energy e and the specific kinetic energy $\frac{1}{2}\mathbf{u} \cdot \mathbf{u}$. The conserved quantity is the total energy per unit volume which can thus be written as

$$\rho E = \rho(e + \frac{1}{2}\mathbf{u} \cdot \mathbf{u}) \tag{1.5}$$

while the specific total enthalpy H is obtained as

$$H = h + \frac{1}{2}\mathbf{u}\cdot\mathbf{u} = E + \frac{p}{\rho} \tag{1.6}$$

with h the specific enthalpy. The Fourier Law is here used to account for the thermal conduction

$$\mathbf{q} = -k(T)\nabla T, \tag{1.7}$$

where $\mathbf{q}$ is the heat flux, with $k(T)$ the thermal conductivity coefficient. The Navier-Stokes equations written in this conservative form will be solved by the Eulerian near-body solver. The Vortex-Particle Mesh method for compressible flows introduced in Chapter 2 will solve a corresponding form using the vorticity field, $\boldsymbol{\omega} = \nabla\times\mathbf{u}$, and the divergence of the velocity field, $\theta = \nabla\cdot\mathbf{u}$, as primary variables. The divergence of the velocity field is also called the dilatation field as it corresponds to the relative rate of change of volume for a material volume

$$\nabla\cdot\mathbf{u} = \frac{1}{V}\frac{dV}{dt}. \tag{1.8}$$

1.3.2 Coupling strategy

In the context of Computer Science, the coupling methodology implemented in this work belongs to the class of algorithms denoted as *co-simulations* and to the subclass of Continuous-Time (CT) based co-simulation. As defined in [63], a co-simulation is the collection of combined simulations produced by the coupled *simulation units* (SUs), which are independent black boxes coupled by an *orchestrator*. The simulation units in this work are the near-body Eulerian solver on the one hand, and the Lagrangian solver on the other hand, which may indeed operate independently. The orchestrator is responsible for the time-management in each SU and for the transfers of information from one SU to another following a *co-simulation scenario*, i.e the information management scheme ensuring a correct co-simulation. This is illustrated in Figure 1.4. As explained in [63], the challenges of CT-based co-simulation include

- the consistent initialization of the simulation units,
- the compositional convergence or the error control of the co-simulation; the compositional error includes the error made in both SUs as well as the error made during the transfers of information between the SUs.
- the compositional stability,
- the compositional continuity, i.e the guarantee that the successive inputs of a SU do not present significant or non-physical changes.

The goal of this work is thus to compute relevant information that needs to be communicated to the other simulation unit and to design a proper co-simulation scenario while keeping these challenges in mind. Implementation efforts will thus take place in each simulation unit and at the orchestrator-level. The remainder of this section will be devoted to identifying the information that needs to be exchanged between the solvers. The time management scheme of the orchestrator will also be presented. The key elements of the coupling strategy will thus be succinctly introduced to put all chapters of this document into perspective.

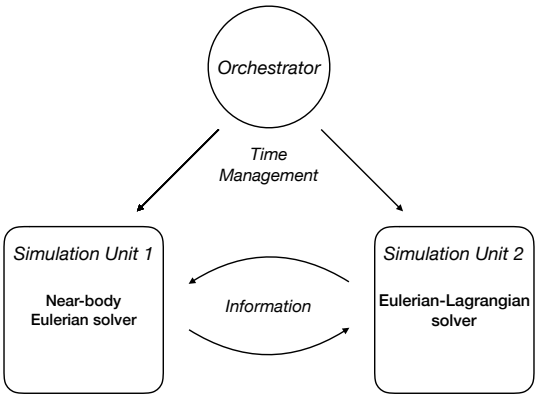


Figure 1.4: Continuous-Time based co-simulation

Domain decomposition technique

The coupling algorithm will rely on a completely overlapping domain decomposition technique initially introduced by Daeninck [44] and depicted in Figure 1.5. The key element of this technique is the use of the Lagrangian solver in the whole computational domain. Despite its purposely insufficient resolution in the near-body region, the Lagrangian solver is then responsible for providing proper boundary conditions to the near-body Eulerian solver. This is made possible by ensuring that the Lagrangian solver has a good integral representation of the solution. One of the elements allowing it is the correction of the near-body Lagrangian solution by the Eulerian solution. The need for an iterative procedure between the solvers is here suppressed due to the combination of the global view of the Lagrangian solver and the correction from the Eulerian solver ensuring a good integral representation in the near-body region. As in every co-simulation, the challenge lies in the quality of

the information exchanged by the solvers, i.e. the boundary conditions and the correction. Failing to provide good quality information for one of these exchanges would swiftly lead to irretrievably damaged solutions; the error would then enter into a self-excited loop between the solvers.

The size of the Eulerian subdomain should be chosen as the locus of the points surrounding the near-body region where the Eulerian and Lagrangian solvers represent the solution with a similar effective resolution (see Chapter 4). Of course, the choice is rather on how that locus is reached as the near-body Eulerian solver comes with a highly flexible discretization of the solution; the mesh resolution can be adapted (*h-adaptivity*) and/or the accuracy of the representation of the solution itself can be adapted element by element in some solvers (*p-adaptivity*). As a consequence, the choice of parameters associated with the near-body region comes down to the classical trade-off between the accuracy in the near-body region and the computational cost.

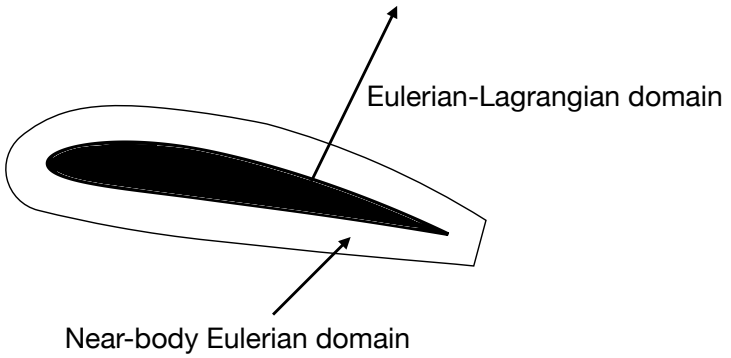


Figure 1.5: Completely overlapping domain decomposition technique: the Eulerian solver is used to capture the strongly anisotropic flow structures in the near-body region. The Lagrangian solver is used in the whole domain and is responsible for properly capturing the flow physics in the wake of the airfoil.

Global time stepping

The Eulerian and Lagrangian solvers will be allowed to use different time stepping strategies, as well as different time steps, to make optimum use of both solvers. When choosing the time integration strategy and the time step used in each solver, the stability of each co-simulation is the first element to take into consideration. The time step in each solver will thus be prescribed

in order to comply with the stability criteria. In this work, an implicit time integration technique allowing to use an arbitrary time step will be used in the Eulerian solver. However, besides the stability, the accuracy of the resulting simulation also drives the choice of the time steps. For the applications considered in this work, the stability and accuracy requirements will be more stringent in the near-body region. As a consequence, the time step in the Eulerian solver will be smaller than its counterpart in the Lagrangian solver. Figure 1.6 schematically depicts the global time-stepping of the coupling methodology. Starting from $t = t_E = t_L$, the Eulerian solver requires the boundary conditions from the Lagrangian solver at different times in the interval $\left[t_L^{n_L},\, t_L^{n_L+1}\right]$. As the Lagrangian solver uses a time step meeting its own criteria, the boundary conditions at these intermediate times (from the point of view of the Eulerian solver) will be linearly reconstructed using the boundary conditions at time $t_L^{n_L}$ and $t_L^{n_L+1}$. These boundary conditions are computed using the following procedure (see Fig. 1.6): at time $t = t_E = t_L$, the accurate solution in the Eulerian region is first used to correct the Lagrangian solution in that region (step 1). The boundary conditions that were computed at the end of the previous time step of the Lagrangian solver are recomputed using the corrected near-body solution (step 2). A single time step is then performed to update the Lagrangian solution (step 3). The boundary conditions at the outer boundary of the Eulerian solver are then computed at the end of the time step (i.e. at $t = t_L^{n_L+1}$) (step 4). In order to ensure the accuracy of the simulation, the Lagrangian time step will be chosen keeping in mind that the whole coupling strategy relies on the quality of the boundary conditions it provides. This implies that the quality of the Lagrangian solution should remain at an acceptable level between two correction steps, despite the under-resolution in the near-body region. The time step of the Lagrangian solver should thus also be chosen so that its solution is adjusted often enough by the Eulerian solver in the near-body region. Moreover, in order to allow the correction of the Lagrangian solution without the need for an additional temporal interpolation of corrections, the time steps are chosen so that $\frac{\Delta t_L}{\Delta t_E} = r$, with r a non-zero integer.

In this work, an identical time step will be used in both solvers. As an implicit time integration scheme is used in the Eulerian solver, it will thus be chosen in order to comply with the stability constraints of the Lagrangian solver and to ensure the temporal accuracy of the simulation. The global time stepping thus reduces to what is depicted in Fig. 1.7. In this case, recomputing the boundary conditions after the correction step is not necessary.

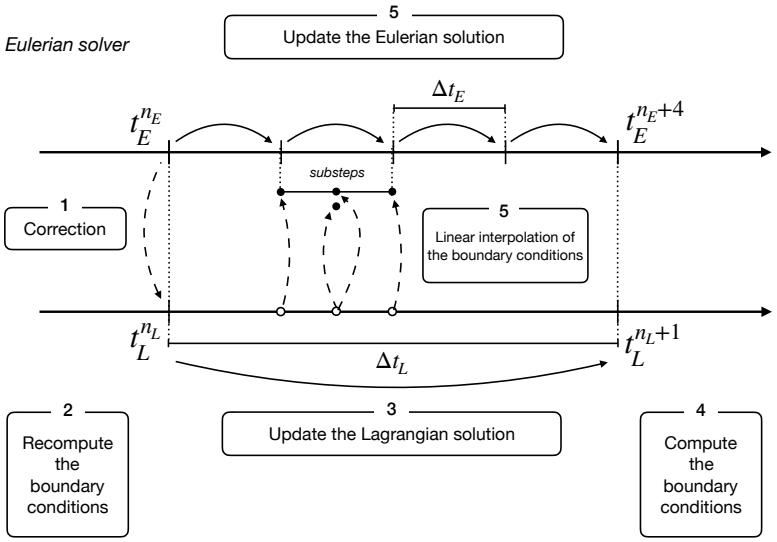


Figure 1.6: Global time stepping strategy with $\frac{\Delta t_L}{\Delta t_E} = 4$.

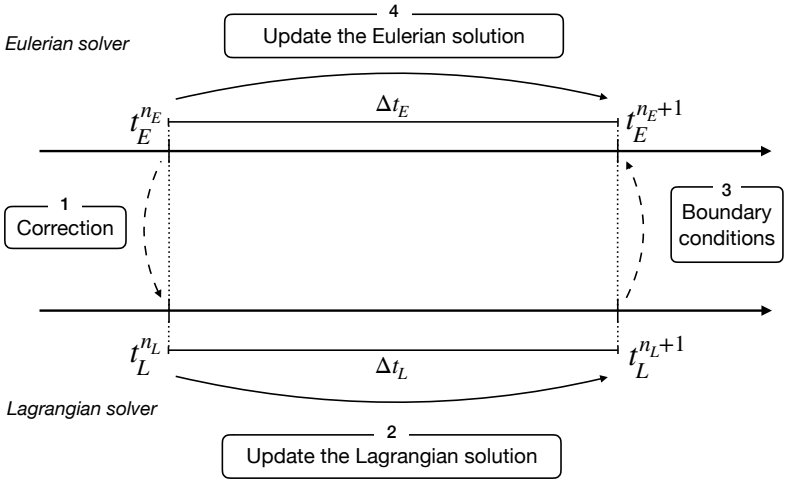


Figure 1.7: Global time stepping strategy when using an implicit time stepping technique in the Eulerian solver and identical time steps in both solvers ($\frac{\Delta t_L}{\Delta t_E} = 1$).

CHAPTER 2

A VORTEX PARTICLE-MESH METHOD FOR SUBSONIC COMPRESSIBLE FLOWS

The content of this chapter was published as a journal article entitled "A Vortex Particle-Mesh method for subsonic compressible flows" [120], with two coauthors, G. Winckelmans and P. Chatelain. It is essentially reproduced here as it was published with some additional developments presented in Appendices A, C, and D. The content of Appendix B was already presented in [120].

This chapter presents the implementation and validation of a remeshed Vortex Particle-Mesh (VPM) method capable of simulating complex compressible and viscous flows. It is supplemented with a radiation boundary condition in order for the method to accommodate the radiating quantities of the flow. The efficiency of the methodology relies on the use of an underlying grid; it allows the use of a FFT-based Poisson solver to calculate the velocity field, and the use of high-order isotropic finite differences to evaluate the non-advective terms in the Lagrangian form of the conservation equations. The Möhring analogy is then also used to further obtain the far-field sound produced by two co-rotating Gaussian vortices. It is demonstrated that the method is in excellent quantitative agreement with reference results that were obtained using a high-order Eulerian method and using a high-order remeshed Vortex Particle (VP) method.

2.1 Introduction

Vortex particle (VP) methods are known to be well suited for the study of convection dominated incompressible flows in an unbounded domain. This capability is attributable to different interesting properties either of the mathematical formulation itself or of the efficient implementations that follow from it. First, VP methods are Lagrangian methods solving the Navier-Stokes equations in vorticity-velocity formulation [40, 160]. Particles are used to discretize the vorticity field while being advected by the local velocity field. This Lagrangian treatment of the advection combined with high-order remeshing schemes – that are required to compensate for the Lagrangian distortion of the particles – has two expected advantages: relaxed stability constraints and a reduced dispersion error. Besides, because of their formulation, VP methods are inherently capable of taking into account unbounded boundary conditions. They also greatly benefit from the compactness of the vorticity field. Vortex Particle-Mesh (VPM) methods, also referred to as Vortex-in-Cell (VIC) methods, derive from it. The Vortex-in-Cell method was first introduced by Christiansen [30] and additional developments were made by others (e.g. [40, 160] as well as references therein and [113], [41], [31]). While keeping the advantages of VP methods, VPM methods also exploit the advantages of an underlying grid: it allows the use of efficient grid-based Poisson solvers (that are needed to calculate the velocity field) and the use of high-order finite difference operators to evaluate the non-advective terms. The high quality of the remeshed VPM methodology was demonstrated against the results obtained using a pseudo-spectral solver in [156]. The advantages of VP and VPM methods were exploited in numerous studies, both in 2-D and in 3-D, and efficient implementations were already achieved on massively parallel architectures both in the case of CPUs and GPUs [27, 135, 162, 163]. The major challenges of the VPM method and vortex methods in general most probably reside in the enforcement of the zero divergence of the vorticity field in 3-D simulations, on the one hand, and in the treatment of wall-bounded flows, on the other hand. For further and extensive information on vortex methods, one can refer to the book of Cottet and Koutmoutsakos [40] and the chapter of Winckelmans in [160], as well as to the references therein.

When compressibility effects can be neglected, numerous studies demonstrated the validity of the strategy of using an incompressible flow solver in conjunction with acoustic analogies so as to predict the far field sound. Of course, vortex methods have been no exception (e.g. Knio et al. [84], Pohtou et al. [130], Jeon et al. [81] and the extensive review of Colonius and Lele [35]) due

to their ability to accurately follow the evolution of the vorticity field, which is of primary importance in sound generation. We also note the work of Huberson et al. in the context of helicopter noise [77]. The strategy implemented consisted in employing a VPM methodology to solve the three-dimensional unsteady inviscid and incompressible flow past a body, the results of which being used as an input for the acoustic wave equation. A section-by-section 2-D compressible Euler correction was used when the compressibility effects could not be neglected (e.g. on the advancing blade).

Some efforts have already been made to bring the Vortex Particle method to the compressible regime: see e.g. the review of Winckelmans [160], the paper of Eldredge et al. [50], and references therein. We will here focus on the VP method for fully compressible flows of Eldredge et al. [50] and the existing VPM methods for compressible flows [153, 114, 118] as they are the most closely related to the methodology proposed in this chapter.

Eldredge et al. [50] developed a high-order remeshed VP method devoted to the simulation of two-dimensional compressible and viscous flows. It will be here referred to as the DVP solver, which stands for Dilating Vortex Particle solver, as in the original paper. The method follows the evolution of integrated quantities over a material volume of the vorticity, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, (as in an incompressible flow solver), the dilatation (i.e. the divergence of the velocity field, $\theta = \nabla \cdot \mathbf{u}$), the density, the enthalpy and the entropy fields. The Helmholtz decomposition is used to compute the velocity field, which is induced by the vorticity field, and the dilatation field, and with the further addition of an imposed far-field potential flow in some cases. Solving for velocity in this Helmholtz decomposition entails the resolution of Poisson equations. In [50], those are solved using a Fast Multipole Method (FMM) [64] adapted to also take into account the dilatation. The DVP solver also relied on an extension of Particle Strength Exchange (PSE) schemes to allow the high-order evaluation of arbitrary spatial differential operators [47]. One-sided stencils were also developed and allowed the enforcement of the required nonreflecting boundary condition needed to truncate the unbounded domain. Eldredge et al. used an eighth-order accurate kernel in the interior of the domain and a second-order accurate kernel to compute the one-sided derivatives at the boundaries. A sixth-order template function was used as the blob function and also as the particle redistribution kernel. This high-order kernel also exhibited a filtering capability. Those two features were necessary to preserve the smoothness of the enthalpy field during the remeshing procedure. The DVP solver was

used to simulate several complex fully compressible flows; the baroclinic generation of vorticity, the sound generation by a co-rotating vortex pair [50]. The resulting scheme was shown to be consistent and in very good agreement with the results of Mitchell et al. [106], who used an Eulerian solver with sixth-order Padé scheme to compute the spatial derivatives. The DVP solver was also used to carry out a thorough analysis of the dynamics and acoustics of viscous 2-D leapfrogging vortices [49].

Regarding the VPM methodology, Thirifay and Winckelmans developed a remeshed VP method dedicated to the study of reacting flows with heat release (and combustion problems especially), using the weakly compressible approximation [154, 160]; they also introduced mesh-based solvers to make it a VPM method referred to as a Vortex-in-Cell (VIC) method by the authors [153][1]. In the weakly compressible case and contrarily to the fully compressible case, the compactness of the fields – one of the most attractive features of vortex methods – can still be usefully exploited.

Following the work of Oxley [114], Papadakis and Voutsinas [118] coupled a 2-D low-order VPM solver for compressible flows with a near-body Eulerian code solving the compressible Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations. The extension of this hybrid solver to the 3-D case was later presented in another conference paper [116]. As it intrinsically and conveniently allows the use of a non-isotropic grid, the Eulerian solver is responsible for the near-body region, while the Eulerian-Lagrangian VPM method is exploited in the wake. The Poisson problems with unbounded boundary conditions were dealt with using a solver [9] based on the James-Lackner algorithm [79, 92] and the Method of Local Corrections [2, 1]. The viscous terms were accounted for using a PSE scheme on the grid. The solver was parallelized using a domain decomposition technique [9]. The particles were periodically remeshed using the third-order accurate M_4' scheme.

The PSE-like methodology exploited in [50] replaces the differential operators by integral operators that eventually require a quadrature to be applied over the particles. Such an implementation is based on nearest neighbours finding and is thus less computationally efficient than a mesh-based evaluation. The latter is adopted in the present work as it allows the use of an underlying grid giving access to efficient grid-based solvers and operators.

[1] For the sake of clarity, we here stick with the VPM naming.

Additionally, Eldredge et al. used for their redistribution a high-order and large support interpolation kernel that included some degree of smoothing. In this work, we wish to pursue an approach in which we regain separate controls over the explicit filtering of fields and the particle remeshing procedure, which does involve some implicit smoothing; this allows more flexibility for operations subject to different constraints on their application frequency. It also allows to benefit from recent advances in high-order interpolation kernels and in high-order grid-based filters.

In order to keep the computational cost relatively low, Papadakis and Voutsinas [118] elected to design a low-order VPM solver. In this work, on the contrary, we wish to retain the high-order capability that was sought by Eldredge et al. [50]. Although it is not explicitly stated in [118], Papadakis' PhD thesis [117] and the conference paper [116] suggest the use of a Gaussian damping function directly applied to the dilatation field. It is meant to avoid spurious reflections at the outer boundaries of the computational domain. In this chapter, however, we wish our boundary conditions to stick to a variant of the methodology successfully implemented in [50] relying on the solution of the linearized Euler equations.

In this chapter we bring the developments introduced by Eldredge et al. to the approach of Particle-Mesh methods. Our effort results in a Vortex Particle-Mesh solver for subsonic compressible flows, which greatly benefits from the mesh. We will refer to it as the VPM-C method in the remainder of this chapter. Efforts are made to end up with a solver that enables the accurate simulation of viscous unsteady compressible flows in an unbounded domain, also capable of simulating aeroacoustics problems. The solver is restricted to the subsonic regime as it does not implement any shock capturing strategy. As we develop the method, we assess and verify several of the numerical schemes employed. Finally, the solver is used to simulate the sound generation by a co-rotating vortex pair. This serves as a benchmark problem assessing the capability of the solver to accurately capture the flow dynamics and the sound generation process. The results are validated against the reference results found in [106, 50, 164].

The chapter is organized as follows. The methodology is introduced in Section 2.2. Starting from the classical VPM methodology for incompressible flows, the features allowing the simulation of unbounded viscous compressible flows are presented. Verification and validation numerical tests are then

carried out on benchmark unbounded problems and compared to analytical solutions or existing results in Section 2.3. Conclusions and perspectives are given in Section 2.4.

2.2 Methodology

In this section, we present the proposed methodology for the general three-dimensional (3-D) case; although it will here solely be applied to a series of two-dimensional (2-D) numerical tests in Section 2.3.

The starting point of the implementation presented in this section is a fast 2-D VPM solver for incompressible flows developed at UCLouvain. Appendix A showcases the capabilities of the solver using the developments made in [102] on the lid-cavity problem, a classical benchmark case that was not tested so far.

2.2.1 Governing equations

We solve the Navier-Stokes equations for an unsteady, compressible and viscous flow in their vorticity-dilatation-velocity formulation. The fluid is supposed to be a perfect gas. For simplicity, the bulk viscosity is here neglected (but our formulation can easily take it into account if it were required), and the dynamic viscosity and the thermal conductivity are assumed uniform. The governing equations for the vorticity ($\boldsymbol{\omega} = \nabla \times \mathbf{u}$) and the dilatation ($\theta = \nabla \cdot \mathbf{u}$) fields are obtained by taking the curl and the divergence of the momentum equation, respectively. The density ρ and the static temperature T are linked to the thermodynamic pressure through the equation of

$$p = \rho R_s T \tag{2.1}$$

with $R_s = c_p - c_v$, the specific gas constant, and c_p and c_v the specific heats at constant pressure and volume, respectively, their ratio being $\gamma = c_p/c_v$. Complementing the equations for $\boldsymbol{\omega}$ and θ with the continuity equation and

the energy equation, the complete set of equations is then

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.2}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\boldsymbol{\omega} \mathbf{u}) = (\nabla \mathbf{u}) \cdot \boldsymbol{\omega} + \frac{\nabla \rho \times \left(R_s \nabla (\rho T) \right)}{\rho^2} + \nabla \times \left(\frac{1}{\rho} \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \right) \tag{2.3}$$

$$\frac{\partial \theta}{\partial t} + \nabla \cdot (\theta \mathbf{u}) = \left(\theta^2 - \nabla \mathbf{u} : (\nabla \mathbf{u})^T \right) - \nabla \cdot \left(\frac{R_s \nabla (\rho T)}{\rho} \right) + \nabla \cdot \left(\frac{1}{\rho} \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \right) \tag{2.4}$$

$$\frac{\partial (\rho T)}{\partial t} + \nabla \cdot (\rho T \mathbf{u}) = (\gamma - 1) \left(\frac{\Phi}{R_s} - (\rho T)\, \theta \right) + \gamma \frac{\mu}{Pr} \nabla^2 T \tag{2.5}$$

The viscous dissipation term Φ is computed as the double contraction between the viscous stress tensor $\underline{\underline{\boldsymbol{\tau}}}$ and the rate-of-strain tensor $\underline{\underline{\boldsymbol{S}}}$ and is obtained as

$$\Phi = \mu\, \omega^2 + 2\mu \left(\nabla \mathbf{u} : (\nabla \mathbf{u})^T \right) - \frac{2}{3} \mu\, \theta^2 \tag{2.6}$$

with $\omega^2 \triangleq \boldsymbol{\omega} \cdot \boldsymbol{\omega}$, μ the dynamic viscosity, k the thermal conductivity and $Pr = \frac{\mu c_p}{k}$ the Prandtl number.

The velocity field $\mathbf{u}$ is calculated using the Helmholtz decomposition

$$\mathbf{u} = \nabla \times \boldsymbol{\Psi} + \nabla \phi + \nabla \varphi_\infty \tag{2.7}$$

with $\boldsymbol{\Psi}$ the streamfunction, ϕ the scalar potential and φ_∞ the imposed far-field potential flow, if any. Setting the gauge $\nabla \cdot \boldsymbol{\Psi} = 0$, the streamfunction and the scalar potential are calculated solving

$$\nabla^2 \boldsymbol{\Psi} = -\boldsymbol{\omega}, \tag{2.8}$$

$$\nabla^2 \phi = \theta. \tag{2.9}$$

The viscous terms in Eqs.(2.3) and (2.4) are rewritten as functions of the density, the vorticity and the dilatation field only

$$\nabla \times \left(\frac{1}{\rho} \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \right) = \tfrac{\mu}{\rho} \nabla^2 \boldsymbol{\omega} + \tfrac{\nabla \rho}{\rho} \times \left(\tfrac{\mu}{\rho} \nabla \times \boldsymbol{\omega} - \tfrac{4}{3} \tfrac{\mu}{\rho} \nabla \theta \right), \tag{2.10}$$

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \right) = \tfrac{4}{3} \tfrac{\mu}{\rho} \nabla^2 \theta + \tfrac{\nabla \rho}{\rho} \cdot \left(\tfrac{\mu}{\rho} \nabla \times \boldsymbol{\omega} - \tfrac{4}{3} \tfrac{\mu}{\rho} \nabla \theta \right). \tag{2.11}$$

Following [50], the equations are solved after nondimensionalization by an acoustic scaling using a characteristic length scale L, the speed of sound of the

far-field medium a_∞, its density ρ_∞, the dynamic viscosity μ, and the specific heat capacity at constant pressure c_p. The dimensionless numbers appearing in the dimensionless equations are then the acoustic Reynolds number $Re = \frac{\rho_\infty a_\infty L}{\mu}$, the Mach Number $M = \frac{U}{a_\infty}$ with U a chosen characteristic speed of the flow, and the Prandtl number Pr.

2.2.2 Vortex Particle-Mesh method

Vortex Particle-Mesh (VPM) methods are built using a hybrid Eulerian-Lagrangian approach involving particles and an underlying Cartesian mesh. A physical field f is discretized using particles characterized by their position $\mathbf{x}_p(t)$, their volume $V_p(t)$ and their strengths $F_p(t) = \int_{V_p} f(\mathbf{x},t)d\mathbf{x} \triangleq f_p(t)V_p(t)$. Their position evolves according to the local velocity field $\mathbf{u}_p(t) = \mathbf{u}(\mathbf{x}_p(t),t)$. Their strength evolve according to the discretized version of the conservation equation of f written in conservative form [160]. The latter reads as follows

$$\frac{\mathcal{D}f}{\mathcal{D}t} \triangleq \frac{\partial f}{\partial t} + \nabla\cdot(f\mathbf{u}) = \mathcal{L}(f,\mathbf{x},t) \tag{2.12}$$

with $\mathcal{L}$ the collection of all non-advective terms as in [104]. The discretization gives rise to a set of ODE's [160, 86]

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p, \tag{2.13}$$

$$\frac{dF_p}{dt} = \left(\frac{\mathcal{D}f}{\mathcal{D}t}\right)_p V_p = \mathcal{L}_p V_p, \tag{2.14}$$

$$\frac{dV_p}{dt} = (\nabla\cdot\mathbf{u})_p V_p = \theta_p V_p \tag{2.15}$$

where $\left(\frac{\mathcal{D}f}{\mathcal{D}t}\right)_p(t) \triangleq \left(\frac{\mathcal{D}f}{\mathcal{D}t}\right)(\mathbf{x}_p(t),t)$ and $\mathcal{L}_p \triangleq \mathcal{L}(f_p,\mathbf{x}_p,t)$. In the present formulation, this discretization is applied to Eqs. (2.2) to (2.5). Particles then carry integrated quantities for the vorticity, the dilatation, the density and the internal energy (divided by the specific heat at constant volume c_v).

A new regular set of particles has to be regularly generated so as to guarantee the convergence of the method [40, 85]. A distorted set of particles is replaced by a new set of particles positioned on a regular mesh and whose strengths are computed as

$$F_q = \sum_p F_p W\left(\frac{\mathbf{x}_q - \mathbf{x}_p}{h}\right) \tag{2.16}$$

where h is the mesh spacing, $\mathbf{x}_q$ is the new position of the particle and W is a moment-conserving interpolation kernel. This operation is conventionally

referred to as "remeshing" or "redistribution".

A Vortex Particle-Mesh method is a hybrid Eulerian-Lagrangian method. Besides the redistribution step that already confers an Eulerian character to a VPM methodology (just as it does for a VP method), the VPM method also uses the grid to efficiently perform several computationally intensive operations. First, it is exploited to perform the spatial differential operations required to evaluate the RHS of Eq. (2.14); i.e. here Eqs. (2.3) to (2.5). Second, the mesh allows the use of efficient grid solvers to solve the Poisson equations Eqs. (2.8) and (2.9). The velocity is then readily obtained on the grid using the Helmholtz decomposition, Eq. (2.7). This methodology requires the transfer of information from the particles to the grid, and conversely. The interpolation of the strengths of the particles at the grid nodes is conventionally referred to as the "particles to mesh" (P2M) operation. This operation allows to get the grid fields from the strengths of the particles and is performed in the same way as the redistribution operation

$$f(\mathbf{x}_{i,j,k}) = \sum_p \frac{F_p}{h^3} W\left(\frac{\mathbf{x}_{i,j,k} - \mathbf{x}_p}{h}\right) \tag{2.17}$$

where $\mathbf{x}_{i,j,k} \triangleq (ih, jh, kh)$. Its counterpart, i.e. the interpolation of grid quantities at the locations of the particles is referred to as the "mesh to particles" (M2P) operation. It is used to transfer the non-advective RHS terms evaluated on the grid, as well as the velocity field to the particles as required by Eqs. (2.13) and (2.14)

$$\mathcal{L}_p = \sum_{i,j,k} \mathcal{L}(\mathbf{x}_{i,j,k}) W\left(\frac{\mathbf{x}_p - \mathbf{x}_{i,j,k}}{h}\right) \tag{2.18}$$

This last operation then allows to advance the strengths, position and volume of the particles in time.

The vorticity field is a divergence-free vector by definition. However, as explained in [40, 160], in 3-D vortex methods, the divergence-free nature of the discretized vorticity field is not guaranteed due to the discretization method itself. The divergence of the discretized vorticity field may hence increase during the simulation, which impairs the quality of the solution. The common way to deal with the divergence problem consists in the periodic reprojection of the vorticity field onto a divergence-free vector field [40, 160]

$$\boldsymbol{\omega}^{\text{new}} = \boldsymbol{\omega} - \nabla Q \tag{2.19}$$

where Q is obtained by solving the Poisson equation $\nabla^2 Q = \nabla \cdot \boldsymbol{\omega}$. The 2-D implementation that will be used in Section 2.3 is evidently exempted to perform this reprojection operation as well as the evaluation of the vortex stretching term in Eq. 2.3.

2.2.3 Vortex Particle-Mesh method for unbounded compressible flows

In the previous section, the ingredients of a classical VPM method were presented. In this section, the features required to extend it to a VPM method for compressible flows are presented.

Dealing with an unbounded domain: high-order FFT-based Poisson solver and radiation condition

To solve a Poisson problem in an unbounded domain, one possible strategy is to compute the convolution of an unbounded integration kernel with the source term using a fast summation algorithm (FMM or FFT-based algorithm). We here use a Fourier-based algorithm which requires a domain doubling technique, as proposed by Hockney and Eastwood [74] and later extended in the context of the VPM method [28, 68]. The integration kernel can either be constructed using a regularized integration kernel or the Lattice Green's function method [61, 62, 97, 98]. We implemented regularized integration kernels derived by Hejlesen et al. [68] that have been shown to exhibit arbitrary high order of convergence. In this work, the kernel leading to a fourth-order Poisson solver is used. The smoothing radius ϵ that appears in the regularized integration kernel is also taken as $\epsilon = 2\,h$. This technique is applied to the Eqs. (2.8) and (2.9) using the FFTW library [54].

The velocity field is then evaluated using the method of spectral differentiation, thus maintaining the order of convergence. Another possibility would have been to directly evaluate the velocity field using regularized velocity kernels. The method of spectral differentiation was, however, deemed the "optimal choice" in [68] as it provides solutions of identical quality at a lower memory cost. The spatial differential operations (except the Lagrangian advection) in Eqs. (2.2) to (2.5) are performed on the grid using finite differences of corresponding order of convergence (see Section 2.2.3). Spectral differentiation would also be an option to evaluate the gradient of the velocity field that appears in Eq. (2.4). In 3-D, the vortex stretching term in Eq. (2.3) can be

evaluated in a similar fashion.

A radiation boundary condition has also been implemented in order to avoid spurious reflections of radiated perturbations at the numerical boundaries that correspond to unbounded directions. In 2-D, Tam and Webb [150] derived the radiation boundary condition Eq. (2.20) from the asymptotic resolution of the linearized Euler equation, in the presence of a uniform mean flow U_∞ in the x-direction (and thus $M_\infty = \frac{U_\infty}{a_\infty}$)

$$\left(\frac{1}{V(\vartheta)}\frac{\partial}{\partial t} + \frac{\partial}{\partial r} + \frac{1}{2r}\right)\begin{pmatrix}\rho_a\\ u_a\\ v_a\\ p_a\end{pmatrix} = 0 + \mathcal{O}\left(r^{-5/2}\right) \tag{2.20}$$

where $V(\vartheta) = a_\infty\left(M_\infty\cos\vartheta + \left(1 - M_\infty^2\sin^2\vartheta\right)^{1/2}\right)$ *is the effective velocity of propagation in the ϑ-direction* [150], r is the radial distance in polar coordinates. Equation (2.20) can be exploited to find a radiation boundary condition for dilatation disturbances. In absence of a mean flow (as in the subsequent sections of this chapter), it follows from Eq. (2.20) that the acoustic disturbances reaching the boundary ($\rho_a = \rho - \rho_\infty$, $\theta_a = \theta$ and $p_a = p - p_\infty$) evolve according to

$$\left(\frac{1}{a_\infty}\frac{\partial}{\partial t} + \frac{\partial}{\partial r} + \frac{1}{2r}\right)\begin{pmatrix}\rho_a\\ \theta_a\\ p_a\end{pmatrix} - \frac{1}{2r}\begin{pmatrix}0\\ \frac{v_r}{r}\\ 0\end{pmatrix} = 0 + \mathcal{O}\left(r^{-5/2}\right) \tag{2.21}$$

where v_r is the radial velocity[2].

Radiation boundary conditions require that only outgoing acoustic disturbances reach the numerical boundary, which implies using a well located and sufficiently large computational domain. As discussed in [50], the extent of the domain is also dictated by the necessity to limit the level of truncation imposed to the non-compact dilatation field as it is directly involved in the computation of the velocity field. As explained in [150], the location of the origin of the polar coordinate system may also be of importance due to the azimuthal

[2]Actually, to be correct, the left-hand side of the equation for the dilatation field in Eq. (2.21) should be complemented with the term $\frac{1}{r}\left[\sin^2\vartheta\frac{\partial u_a}{\partial x} - \cos\vartheta\sin\vartheta\left(\frac{\partial u_a}{\partial y} + \frac{\partial v_a}{\partial x}\right) + \cos^2\vartheta\frac{\partial v_a}{\partial y}\right] = \frac{1}{r^2}\left(-\sin\vartheta\frac{\partial u_a}{\partial\vartheta} + \cos\vartheta\frac{\partial v_a}{\partial\vartheta}\right)$. However, the simulations in this thesis were performed using Eq. (2.21). Nonetheless, both for the centered and the decentered acoustic pulses of Section 2.3.1, the results obtained with the corrected form of the radiation condition were almost indistinguishable to those obtained using Eq. (2.21), which are presented in Section 2.3.1.

dependence of the radiation boundary condition. The best performance will be achieved when the acoustic disturbances are normal to the boundary [51]. From a computational point of view, the boundary condition Eq. (2.21) is imposed using particles located in a buffer region outside the computational domain as in [50]. These buffer particles are allowed to move and change volume, their vorticity is maintained constant (and equal to zero) whereas their density and internal energy evolve according to

$$\frac{d}{dt}\left(f_p S_p\right) = -a_\infty \left(\frac{\partial f}{\partial r} + \frac{(f - f_\infty)}{2r}\right)_p S_p + f_p \theta_p S_p \tag{2.22}$$

and the dilatation evolves according to

$$\frac{d}{dt}\left(\theta_p S_p\right) = -a_\infty \left(\frac{\partial \theta}{\partial r} + \frac{\theta}{2r} - \frac{1}{r}\frac{v_r}{2r}\right)_p S_p + \theta_p^2 S_p \tag{2.23}$$

where S_p is the material surface associated to the particle in 2-D. It evolves according to the 2-D counterpart of Eq. (2.15). Equations (2.22) and (2.23) are obtained by first applying the product rule and then using Eq. (2.21).

The radial derivatives involved in Eqs. (2.22) and (2.23) are evaluated on the grid using third-order decentered finite differences. The performance of the radiation condition will be assessed in Section 2.3.1. In [50], only the enthalpy was subject to the first member of the Engquist-Majda hierarchy of conditions [51] for circular boundaries while the other transported quantities were maintained constant. In the absence of a uniform flow – as in the numerical tests of Section 2.3 –, the radiation condition Eq. (2.20) has the same form as the first member of the Engquist-Majda hierarchy.

The radiation boundary conditions derived in [150] were later extended to the case of non-uniform mean flow by Tam and Dong [151]. The extension to 3-D and in the presence of a non-uniform mean flow was introduced by Bogey and Bailly in [14] and could be adapted to fit in a 3-D implementation of the present methodology (as it was done hereinbefore with Eq. (2.20) in 2-D). Using spherical coordinates (r, ϑ, φ), it reads

$$\left(\frac{1}{v_g}\frac{\partial}{\partial t} + \frac{\partial}{\partial r} + \frac{1}{r}\right)\begin{pmatrix}\rho_a \\ \mathbf{u}_a \\ p_a\end{pmatrix} = 0 \tag{2.24}$$

with $v_g \mathbf{e}_r$ the speed of wave propagation. It is obtained as the vector sum of the mean sound speed $\bar{\mathbf{c}}$ and the mean flow velocity $\bar{\mathbf{u}}$ projected in the sound propagation direction $\mathbf{e}_r$.

Non-advective terms evaluation

As explained in [50] and references therein, the slight imbalance between the two terms $-\nabla^2 h$ and $-\nabla \mathbf{u} : (\nabla \mathbf{u})^T$ involved in Eq. (2.4) (recall that $\nabla \cdot \left(\frac{\nabla p}{\rho} \right) = \nabla^2 h - \nabla \cdot (T \nabla s)$) is of acute importance in the sound generation process. Various strategies of evaluation have been tested; only finite differences were considered to evaluate the first term whereas two different methods were considered for the second one: spectral differentiation and finite differences. Although each one of these methods can achieve an order of convergence that matches the one chosen for the Poisson solver, it turns out that the lower error level is achieved when the same technique is used to evaluate both terms. Both terms are then evaluated using finite differences.

In order to reduce the anisotropy of classical finite difference operators, fourth-order explicit schemes that are less sensitive to the grid orientation were considered, both for the Laplacian and for first order derivatives. They are presented in Appendix B together with their second-order counterparts. Their performances are compared with those of standard explicit schemes through modal analysis for the 2-D diffusion equation and the 2-D advection equation. In spite of having an error equal or greater to that of classical schemes (amplitude or phase error respectively), the presented schemes indeed exhibit a more isotropic error. Incidentally, we here mention schemes proposed by Lele [94], and further discussed in [35]. These schemes are compact explicit schemes that optimize their dispersion error. They also exhibit a lower anisotropy than standard explicit schemes and than the isotropic explicit schemes presented here. However, these interesting features come at a higher computational cost [35]. Let us also recall that the first order derivatives that need to be evaluated on the grid only appear in the non-advective terms of the conservation equations, the advection part being here treated in a Lagrangian fashion. We here elect to use the fourth-order explicit schemes in the remaining sections of this paper.

2.2.4 Remeshing and Filtering

As explained in Section 2.2.2, the particles need to be regularly redistributed. The physical fields whose integrated values are advected by those particles also need to be regularly filtered in order to remove the spurious high-wavenumber frequency modes appearing in the solution. In this section, our choices will be based on the learning of Eldredge et al. [50] while benefiting from the advantages of the new high-order redistribution schemes and of

high-order grid-based filters.

In [50], a sixth-order PSE-like interpolation kernel featuring smoothing properties was derived and used as the redistribution kernel instead of classical interpolation kernels such as the M_4' kernel [109]. The latter is third-order accurate, has a discontinuous second order derivative and was found to be more restrictive regarding the required resolution and the remeshing frequency. The sixth-order smoothed kernel was chosen as a member of a family of kernels constructed as the product of one-dimensional polynomial functions and a Gaussian function. The kernel is scaled by an smoothing radius σ that controls the filtering and a value $\sigma/h = 1.7$ was chosen in [50]. The 1-D scaled kernel has the following expression:

$$W_E^\sigma\left(\xi, \frac{\sigma}{h}\right) = \frac{1}{\sigma/h} W_E\left(\frac{\xi}{\sigma/h}\right) \tag{2.25}$$

with

$$W_E(\xi) = \frac{1}{\sqrt{\pi}}\left(\frac{15}{8} - \frac{5}{2}\xi^2 + \frac{1}{2}\xi^4\right) e^{-\xi^2} \tag{2.26}$$

The transfer function (Fourier transform) of the kernel is given by

$$\widehat{W}_E^\sigma(k) = \frac{1}{32}\left[\left(\frac{\sigma}{h}k\right)^4 + 8\left(\frac{\sigma}{h}k\right)^2 + 32\right] \exp\left(-\frac{1}{4}\left(\frac{\sigma}{h}k\right)^2\right) \tag{2.27}$$

with $\widehat{W}_E^\sigma(kh = 0) = 1$, $\widehat{W}_E^\sigma(kh = \pi) = f(\sigma/h)$ not necessarily equal to 0 – it is not the case with $\sigma/h = 1.7$ and the value derived here under.

A slight disadvantage of this kernel is that it does not exactly conserve the integral quantities for finite-width stencil although the departure rapidly decreases with the stencil width. As a consequence, the kernel does not have a strictly speaking compact support. For instance, using $\sigma/h = 1.7$ and a 12-point stencil, we get, for the zeroth order moment, $(1 - \sum_p W_\sigma(\mathbf{x}_p)) = \mathcal{O}(10^{-6})$ when the redistribution kernel is applied to already remeshed particles. Keeping the stencil width unchanged, the departure can be reduced to $\mathcal{O}\left(10^{-16}\right)$ using $\sigma/h = 1.5177720560756227$.

In this work, we elect to employ the M_6^* kernel derived in [11] for high-order calculations in combination with explicit discrete filters. It is defined

by:

$$M_6^*(\xi) = \begin{cases} 0 & \text{if } |\xi| > 3, \\ -\frac{1}{24}\left(|\xi|-2\right)\left(|\xi|-3\right)^3\left(5|\xi|-8\right) & \text{if } 2 < |\xi| \leq 3, \\ \frac{1}{24}\Big[\left(25|\xi|^3-114|\xi|^2+153|\xi|-48\right) & \\ \qquad \left(|\xi|-1\right)\left(|\xi|-2\right)\Big] & \text{if } 1 < |\xi| \leq 2, \\ -\frac{1}{12}\Big[\left(25|\xi|^4-38|\xi|^3-3|\xi|^2+12|\xi|+12\right) & \\ \qquad \left(|\xi|-1\right)\Big] & \text{if } |\xi| \leq 1. \end{cases} \tag{2.28}$$

This fourth-order accurate interpolation kernel has a 6-point support and is such that $M_6^* \in C^2(\mathbb{R}^3)$ as opposed to the M_4' kernel as mentioned above. van Rees et al. showed that the results obtained using a hybrid vortex particle-mesh method for incompressible flows using this kernel are in good agreement with those obtained using a pseudo-spectral method [156]. In their paper, they showed that, although the M_4' kernel already leads to results being in qualitative agreement with those of the pseudo-spectral code, the M_6^* was also in quantitative agreement. The distinction between the remeshing and the filtering steps also provides more flexibility to the methodology. It allows a better control on operations serving dissimilar purposes and subject to different constraints on their application frequency. The particles are remeshed every n_r time steps and filtered every $n_f = \alpha n_r$ time steps with α an integer – the freshly remeshed particles are filtered right after the remeshing step. The filtering operation can thus be performed more sparingly than the remeshing operation or even be deactivated if it is not necessary. The surface of the particles is kept unchanged during the filtering step. In this work, explicit discrete filters were considered. The 1-D second-order explicit filter is such that the filtered quantity $\bar{f}$ is obtained as

$$\bar{f}_i = f_i + \frac{1}{4}(f_{i-1} - 2f_i + f_{i+1}) \triangleq f_i - \left(-\frac{\delta^2}{4}\right) f_i \triangleq f_i - f_i^s \tag{2.29}$$

where f_i^s denotes the second-order small scales part of f_i and $-\frac{\delta^2}{4}$ the second order small scales operator. Its transfer function is given by

$$\mathcal{G}(kh) = 1 - \sin^2\left(kh/2\right). \tag{2.30}$$

Iterations performed on the second-order small scales operator $-\frac{\delta^2}{4}$ produce higher order filters [80]. A filter of order $2n$ is obtained by n iterations and its transfer function is

$$\mathcal{G}^{(n)}(kh) = 1 - \left(\sin^2\left(kh/2\right)\right)^n \tag{2.31}$$

and is still such that $\mathcal{G}^{(n)}(0) = 1$ and $\mathcal{G}^{(n)}(\pi) = 0$. Figure 2.1 shows the transfer function of the PSE-like interpolation kernel with $\sigma/h = 1.7$ and $\sigma/h = 1.517772$ and the transfer function of explicit discrete filters of order 4, 6 and 8. All of these operators are such that $G(0) = 1$ although the PSE-like interpolation kernel would require an infinite span to scrupulously respect this property. $G(\pi) = 0$ for the discrete explicit filters but not for the PSE-like interpolation kernel using the considered values for σ/h. From these trends, we expect that the filtering frequency will be required to be higher when using the PSE-like interpolation kernel with the corrected value of σ/h. The same remark holds for discrete filters of increasing order.

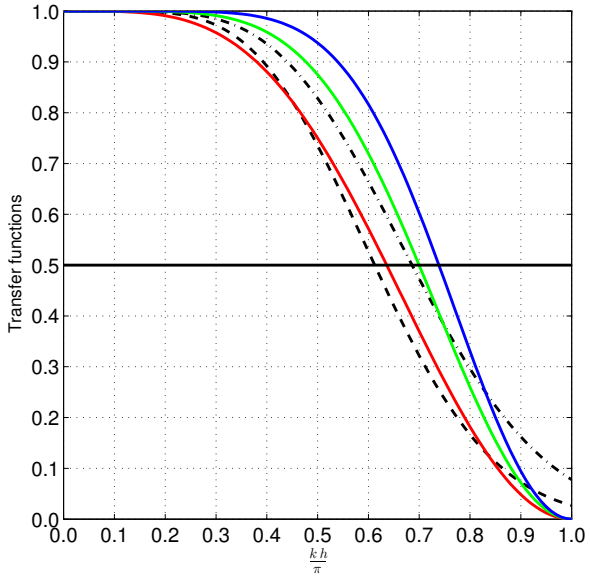


Figure 2.1: Transfer function of: explicit filters of order 4 ($n = 2$), order 6 ($n = 3$), and order 8 ($n = 4$) (solid red, green and blue respectively); sixth-order interpolation kernels developed in [50] : using the original value ($\sigma/h = 1.7$) (dashed), using the modified value ($\sigma/h = 1.517772$) (dashed-dotted)

The dispersion and dissipation properties of the redistribution scheme of Eldredge [50] are compared to those of the combination of the M_6^* scheme and a discrete filter in Appendix C. It is shown that the redistribution scheme of Eldredge exhibits excellent dispersion properties.

We note that efforts that attempted to address the issues of accuracy and isotropy were also put into redistribution schemes in [29]. In addition to other

interesting features, those schemes were shown in [29] to be more isotropic than schemes built using the tensor product of 1D schemes.

In the simulations performed in Section 2.3, the remeshing and filtering operations are performed periodically rather than triggered as a result of some conservative threshold overrun that would be evaluated during the simulation; it indeed proved difficult to establish firm criteria that could be used to determine when the particles needed to be relocated and/or filtered. In his PhD thesis [48], Eldredge developed a criterion to help determine a suitable remeshing-filtering frequency (as the two operations are combined in [50] and [48]) but it is emphasized that the choice remains "problem-specific"; a conclusion that was also drawn in this work. Indeed, the major difficulties in a compressible VP or VPM method are to accurately capture the dilatation field and to avoid the appearance of spurious oscillations that could impair the quality of the solution. The following procedure is thus applied to determine the time step, the remeshing and the filtering frequencies. The time step is first chosen in order to respect the classical criteria on $\|\boldsymbol{\omega}\|_\infty \Delta t$ and $\|\boldsymbol{S}\|_\infty \Delta t$ used for the simulations of incompressible flows. The zero-level of the dilatation field is then used as an indicator of the quality of the solution. The time between two filtering operations is shortened if spurious oscillations appear in the zero-level of the dilatation field. Of course, this may imply to increase the remeshing frequency (as $n_f = \alpha n_r$ with α an integer) and possibly to lower the time step itself. Following this procedure, the remeshing and filtering operations are performed as sparingly as possible, which is desirable as they also introduce numerical errors.

2.3 Numerical tests

The method designed in Section 2.2 is here verified and then validated against results from the literature. The radiation boundary condition is validated on the benchmark case of an acoustic pulse in an unbounded domain. The results will be compared to the analytical solution. The second part of this section is devoted to the verification and validation of the VPM-C solver on the basis of the simulation of co-rotating vortices evolving in a compressible medium. Spatial and temporal refinement studies will be made to verify that the expected orders of convergence are indeed achieved. A long-time simulation is then carried out. The results will be validated against reference simulations from the literature: the simulation of Eldredge et al. using the DVP solver [50]

(and thus the simulation of Mitchell et al. [106]) and the simulation of Zhang et al. using their WENO solver [164].

2.3.1 Radiation of an acoustic pulse

The purpose of this test is to assess the radiation condition described in Section 2.2 and it consists in the radiation of an acoustic pulse. The fluid is initially at rest, and the pressure and density fields exhibit a Gaussian excess zone:

$$u(0) = v(0) = 0, \tag{2.32}$$

$$\frac{p_a(0)}{\rho_\infty a_\infty^2} = \frac{\rho_a(0)}{\rho_\infty} = \varepsilon \exp\left(-\frac{r^2}{\sigma^2}\right) = \varepsilon \exp\left(-\ln 2\,\frac{r^2}{b^2}\right) \tag{2.33}$$

with $p_a = p - p_\infty$ the pressure disturbance, $\rho_a = \rho - \rho_\infty$ the density disturbance, $r = \left((x-x_c)^2 + (y-y_c)^2\right)^{1/2}$ the radial distance in polar coordinates, and b the half-width of the Gaussian. The computational domain Ω is a circular region of radius $R_\Omega = 4\sigma$ and centered at (x,y) = (0,0). The boundary zone $\partial\Omega$ is an annular region surrounding the interior domain and is a few particles wide. The pulse is here located at the center of the computational domain and its maximum value is $\varepsilon = 1.0 \cdot 10^{-3}$. As the amplitude of the perturbation is small, the flow is here governed by the linearized Euler equations. We, however, solve the full Navier-Stokes equations as in [50] in order to test the full solver. The full Navier-Stokes equations Eqs. (2.2) to (2.5) are thus solved in Ω whereas the boundary conditions Eqs. (2.22) and (2.23) are imposed in $\partial\Omega$. The fields are not filtered in the present test case. As advised in [34], a fine grid is used in order to isolate the error due to the boundary treatment from the truncation error in the domain; 40 particles lay across the half-width diameter $2b$. The particles are remeshed on the Cartesian grid using a M_6^* kernel every time step (i.e. $n_r = 1$), without filtering. The fourth-order regularized integration kernel derived in [68] is used to solve the Poisson problems Eqs. (2.8) and (2.9). The smoothing radius ϵ of the considered integration kernel was chosen such that $\epsilon/h = 2$. Spectral differentiation is carried out to obtain the velocity field. Other spatial derivatives are evaluated using isotropic fourth-order centered finite differences in the computational domain, and using third-order decentered finite differences for the radiation condition. The temporal integration is carried out using the fourth-order Runge-Kutta scheme using a time step $\Delta t\, a_\infty/\sigma = 0.020$.

Figure 2.2 shows that the pulse spreads as a cylindrical wave and then leaves the domain without noticeable reflection. Figure 2.3 shows the temporal evolution of the rms value and maximum value of the pressure and

dilatation fields. The results are compared to the analytical solution whose expression for the velocity and pressure disturbances were derived in [150]. The analytical solution for the dilatation is derived in Appendix D. The rms and maximum values of the analytical solution are obtained using the numerical evaluation of the solution at the grid points of the same Cartesian mesh. It is seen on Fig. 2.3 that the wave propagates at the correct speed up to the time at which it leaves the domain, $\frac{t\,a_\infty}{\sigma} \simeq 4$. A fast decay of the disturbances within the domain Ω then occurs after that. The numerical results are in excellent agreement with the exact solution until $t\,a_\infty/\sigma \simeq 6$, after which a small reflection is measured. A second test case considering a decentered pulse was also performed in order to gauge the sensitivity of the radiation boundary condition to the directionality of the acoustic disturbances. The amplitude of reflected waves is indeed expected to increase with the angle of incidence of the incoming waves [51]. The acoustic pulse was initially located at $\left(\frac{x}{\sigma}, \frac{y}{\sigma}\right) = (0.5, 0.5)$. The consequences of this non-optimal configuration manifested through a loss of central symmetry in the pressure contours as shown in Figure 2.4. However, the comparison with the analytical solution shown in Figure 2.5 demonstrates that the performance of the boundary condition is only slightly altered.

2.3.2 Co-rotating vortex pair

The second validation test case consists in two co-rotating vortices in a compressible medium and in an unbounded domain. The Gaussian-distributed vortices are initially separated by a distance $2R$ and each of them is such that

$$\omega(r) = \frac{\Gamma}{\pi\sigma^2} \exp\left(-\frac{r^2}{\sigma^2}\right) \tag{2.34}$$

with r the distance from the vortex center. The maximal self-induced azimuthal velocity U_0 is reached at $r = r_0$ such that $\frac{r_0^2}{\sigma^2} = \beta = 1.2564$. The present test case is such that $\frac{r_0}{R} = 0.15$. The circulation based Reynolds is such that $Re_\Gamma = \frac{\Gamma}{\nu} = 7500$. The Mach number based on the maximal azimuthal self-induced velocity is $M_0 = \frac{U_0}{a_\infty} = 0.56$; this is quite significant in terms of density variations and, subsequently, of the volume changes undergone by the particles. With such parameters, the co-rotation Mach number is obtained as $M_c = \frac{1}{\left(1-\exp\left(-\beta\right)\right)} \frac{r_0}{2R} M_0 = 0.059$. The ratio of specific heats and the Prandtl number are taken as $\gamma = 1.4$ and $Pr = 0.70$, respectively.

The fourth-order regularized integration kernel derived in [68] is used to solve the Poisson problems (Eqs. (2.8), (2.9) and Eq. (2.35)). The spatial deriva-

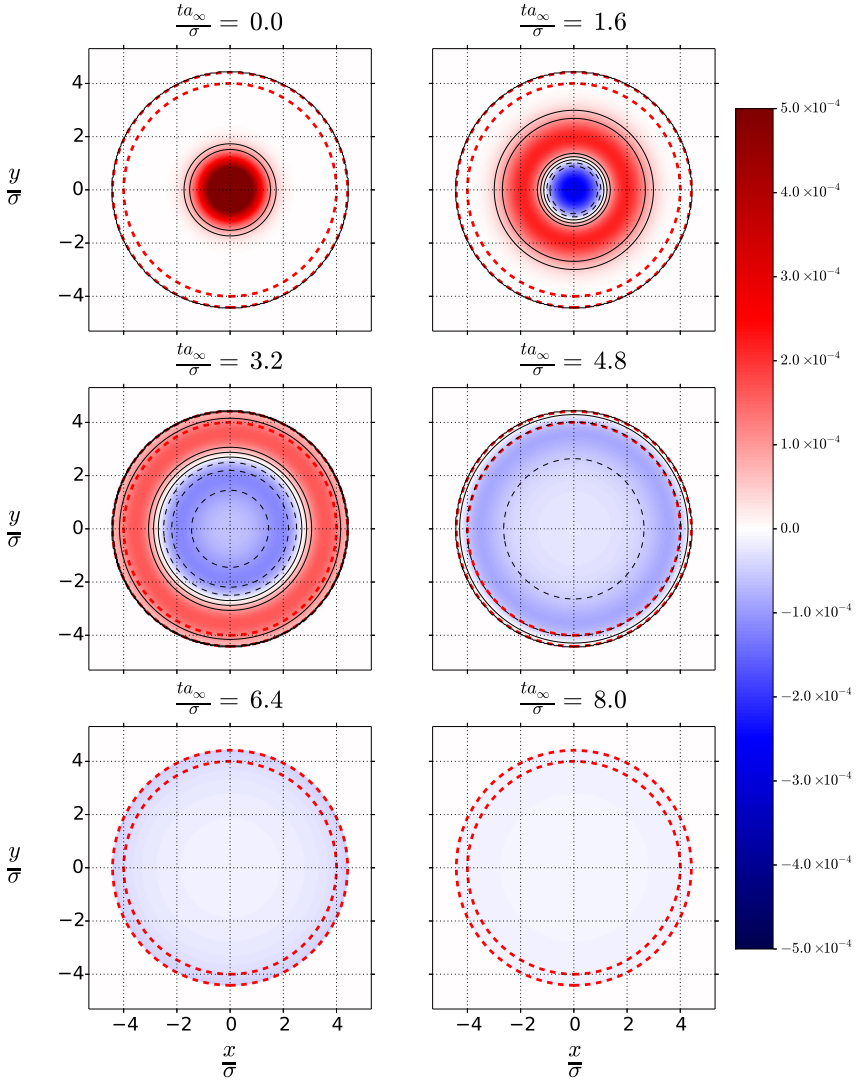


Figure 2.2: Acoustic pulse: Contours of pressure disturbances $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$. The contours are in the range $[-0.1\varepsilon, 0.1\varepsilon]$ in increments of 0.05ε (negative contours in thin dashed, positive contours and the zero level contour in thin solid). The radiation condition is applied in the region delimited by the thick dashed red lines.

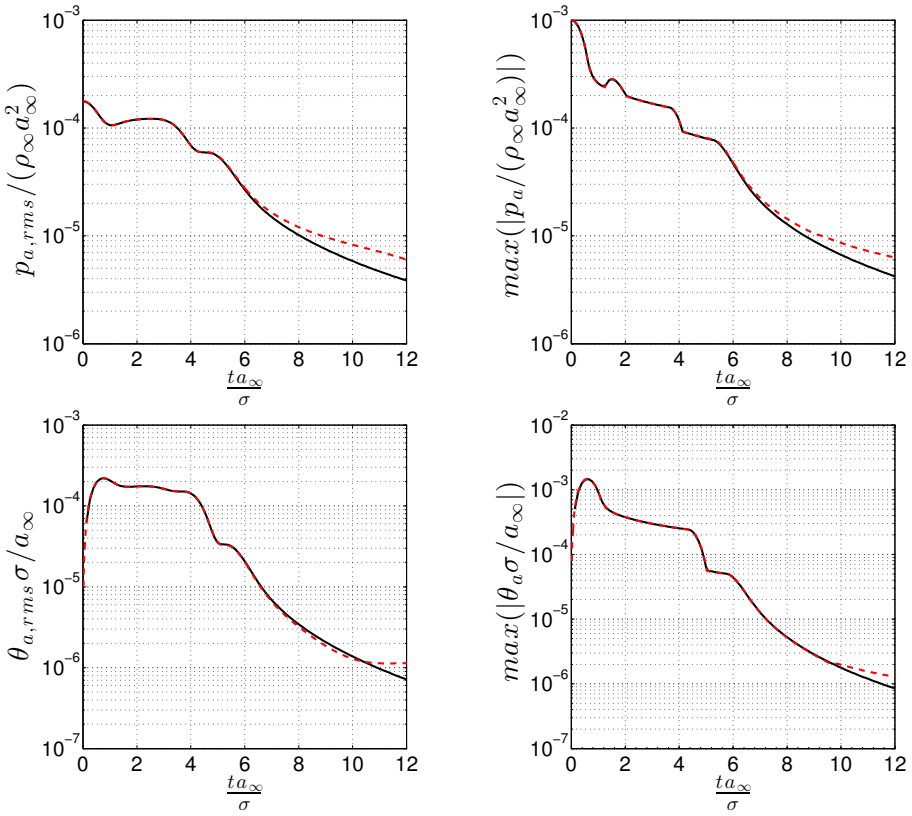


Figure 2.3: Acoustic pulse: Temporal evolution of the rms. value (left) and maximum value (right) of the pressure (top) and dilatation (bottom) fluctuations respectively. Exact solution : solid line, VPM-C: dashed line.

tives are evaluated using fourth-order isotropic centered finite differences in the computational domain and using third-order decentered finite differences for the radiation condition. The temporal integration is carried out using the fourth-order classical Runge-Kutta scheme. Unless otherwise stated, the radius of the computational domain Ω is chosen such that $R_\Omega = 4R$. As explained in Section 2.2.3, the accuracy of the radiation condition (Eqs. (2.22) and (2.23)) is sensitive to the choice of the origin of the polar coordinate system in which the boundary condition is written. In this case, the geometrical center of the vortical system is obviously the best choice. The particles are periodically remeshed using the M_6^* scheme presented in Section 2.2.4. When necessary, the remeshed particles are further filtered using the discrete filter of order 8. Depending on the test case, different choices of remeshing (n_r) and

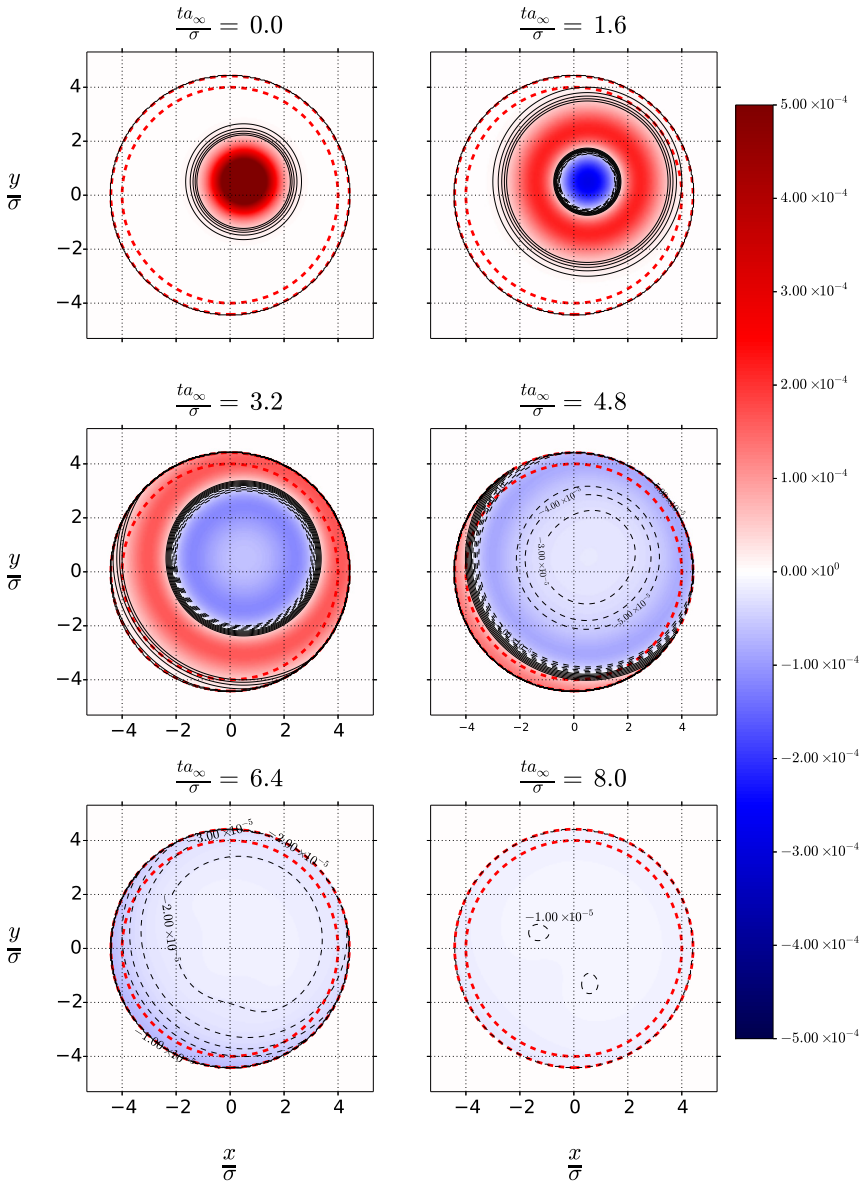


Figure 2.4: Decentered acoustic pulse: Contours of pressure disturbances $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$. The contours are in the range $[-0.05\varepsilon, 0.05\varepsilon]$ in increments of 0.01ε (negative contours in thin dashed, positive contours and the zero level contour in thin solid). The radiation condition is applied in the region delimited by the red thick dashed lines.

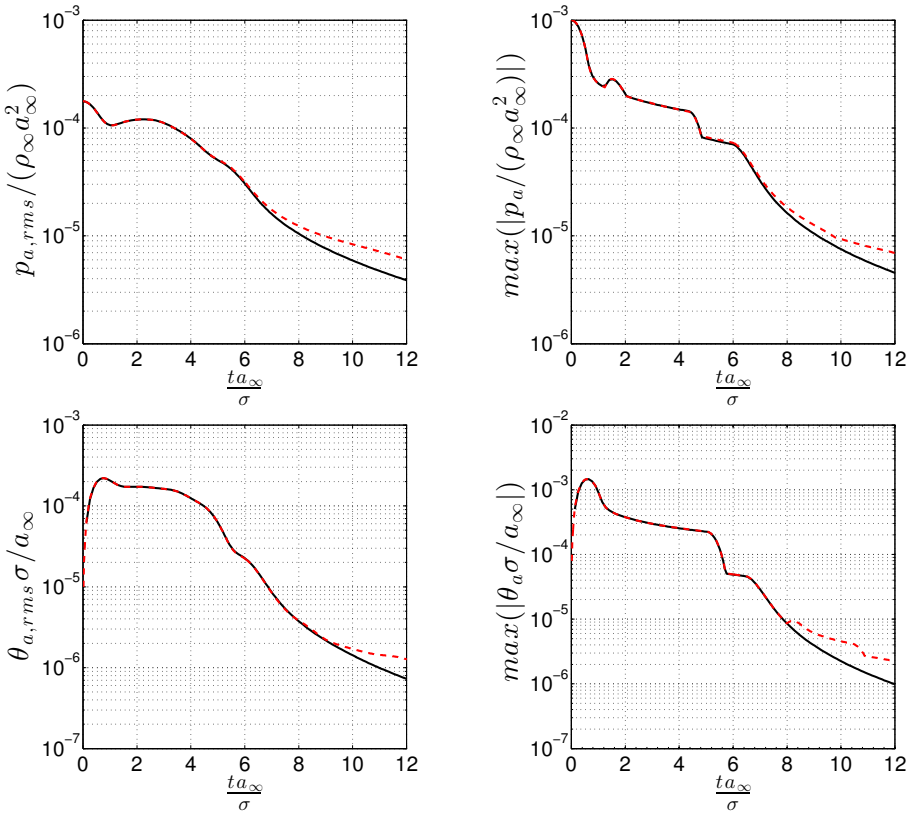


Figure 2.5: Decentered acoustic pulse: Temporal evolution of the rms. value (left) and maximum value (right) of the pressure (top) and dilatation (bottom) fluctuations respectively. Exact solution : solid line, VPM-C: dashed line.

filtering (n_f) frequencies will be made in the subsequent sections.

The particles are initially positioned on a uniform Cartesian grid and their initial surface thus corresponds to the grid cell size ($S_p(0) = h^2$). Following [50] and [106] and in order to reduce the magnitude of the acoustic transient at the onset of the simulation, the flow is initially specified as homentropic and solenoidal. It then follows, from Eq. (2.4), and neglecting the viscous term, that the initial enthalpy is obtained by solving the following Poisson problem:

$$\nabla^2 h = -\nabla\mathbf{u} : (\nabla\mathbf{u})^T . \tag{2.35}$$

The same methodology as for the Poisson problems ensued by the Helmholtz decomposition is implemented. The only difference lies in the evaluation of

the source term. Indeed, it is here evaluated on the whole computational square grid and not only in Ω. Spectral differentiation is then used in lieu of finite differences. The initial density is then computed using the isentropic relation:

$$\frac{\rho}{\rho_\infty} = \left(\frac{T}{T_\infty}\right)^{1/(\gamma-1)}. \tag{2.36}$$

The results will be compared to those obtained in the literature [106, 50, 164]. Mitchell et al. [106] used an Eulerian solver employing a sixth-order compact Padé scheme [94] and the fourth-order Runge-Kutta scheme. The computational domain covered the near-field and the far-field regions in order to allow the direct evaluation of the far-field sound. Indeed, one of the goals of this paper was to investigate the quantitative validity of existing acoustic analogies (including the Möhring analogy, see Section 2.3.2). A fourth-order accurate compact Padé filter [94] was also necessary, but its use was limited to the initial phases of the simulation. It allowed to suppress spurious oscillations of the same level as the physical pressure fluctuations. These oscillations resulted from unphysical reflections arising when the acoustic transient crossed a region of grid stretching that ensured a smooth transition from a fine uniform grid in the near field and to a coarser uniform grid in the far field. The zeroth-order boundary conditions described in [33] were used to allow outgoing waves to leave the domain. Eldredge et al. validated the DVP solver presented in Section 2.1 using the results in [106].

Zhang et al. [164] studied the sound generation process due to the interaction of two Taylor vortices and identified a classification of the interaction depending on the relative strengths of the vortices and on the geometric parameters. They also performed the simulation of Gaussian vortices as in [106, 50] and discussed the difference regarding the sound generation process between these two types of vortices. They used the fifth-order weighted essentially non-oscillatory (WENO) finite difference scheme of Jiang and Shu [82, 143] to discretize the nonlinear first order derivative terms while the viscous terms were discretized using a fourth-order finite difference scheme. The temporal discretization was carried out using the third-order total variation diminishing (TVD) Runge-Kutta method of Shu and Osher [144, 143].

Spatial and temporal discretization errors

The order of accuracy of the discretization techniques used in the VPM-C solver are verified using refinement studies on the early stages of this flow. Approximations of the discretization errors are evaluated by a pointwise com-

parison between the grid solutions obtained using a given resolution and using the finest resolution. The following error norms are used:

$$\begin{aligned} \epsilon_2 &\triangleq \|\epsilon\|_2 \triangleq \frac{h}{R}\left(\sum_{i,j}\left(f(\mathbf{x}_{i,j}) - f_{ref}(\mathbf{x}_{i,j})\right)^2\right)^{\frac{1}{2}} \\ \epsilon_\infty &\triangleq \|\epsilon\|_\infty \triangleq \max_{i,j}|f(\mathbf{x}_{i,j}) - f_{ref}(\mathbf{x}_{i,j})| \end{aligned}$$

with f a dimensionless grid field, f_{ref} the dimensionless reference solution obtained using the finest resolution and h the mesh spacing. The comparison is carried out in a narrowed circular region precluding the boundary condition to reduce the spatial and temporal orders of accuracy. The radius of the control region was chosen to be $0.75R_\Omega$.

The spatial refinement study is performed using $\frac{(\Delta x)_i}{(\Delta x)_0} = \frac{1}{2^i}$ with $i = 0, 1, 2, 3, 4$ and $\frac{(\Delta x)_0}{R} = 3.5 \cdot 10^{-2}$. The reference solution was computed using the finest resolution, $\frac{(\Delta x)_{ref}}{(\Delta x)_0} = \frac{(\Delta x)_4}{(\Delta x)_0}$. The same time step $\frac{\Delta t a_\infty}{R} = 2.0 \cdot 10^{-5}$ is used for all cases and the error norms are computed after 10 time steps. The particles were remeshed at each time step and filtering was not used. The grids were constructed such that a direct correspondence exists between the points of the finest grid and those of the coarser grids.

The temporal refinement study is performed using $\frac{(\Delta t)_i}{(\Delta t)_0} = \frac{1}{2^i}$ with $i = 0, 1, 2, 3$ and $\frac{(\Delta t)_0 a_\infty}{R} = 1.024 \cdot 10^{-2}$. The same spatial resolution is used in each case: the number of particles placed across the vortex core diameter $2r_0$ is $N_c = 13$. The remeshing frequencies were chosen such that $\frac{(n_r)_i}{(n_r)_0} = \frac{(\Delta t)_0}{(\Delta t)_i}$, with $(n_r)_0 = 1$ and filtering was not applied. The reference solution was computed using $\frac{(\Delta t)_{ref} a_\infty}{R} = 1.0 \cdot 10^{-5}$, or $\frac{(\Delta t)_{ref}}{(\Delta t)_0} = \frac{1}{2^{10}}$. The error norms were computed after $T = 10\ (\Delta t)_0$. As shown in Figs. 2.6 and 2.7, a fourth-order accuracy appears to be achieved in both refinement studies that solely focus on the verification of the order of accuracy of the discretization techniques. It is, however, unlikely that the method is globally fourth-order accurate. Indeed, in [42], Cottet et al. showed that, besides the interpolation property and the conservation of moments up to a given order p, the regularity of a redistribution kernel is also of importance as regards the order of accuracy of a remeshed Particle Method. In their terminology, the classical M_4' scheme corresponds to the $\Lambda_{2,1}$ scheme, the subscripts indicating that the kernel preserves the moments up to $p = 2$ and is of class C^1. The M_6^* kernel used in the present work corresponds to the $\Lambda_{4,2}$ kernel and should thus lead to a solver having an overall second order of accuracy, due to the limited regularity of

the remeshing kernel. In order to obtain an overall fourth-order accuracy, the numerical analysis performed in [42] suggests using the $\Lambda_{4,4}$ remeshing kernel derived in the same paper. As explained in Section 2.2.4, we, however, elected to use the M_6^* scheme as it was shown in [156] that an incompressible VPM solver using this scheme was in good quantitative agreement with a pseudo-spectral solver on benchmark cases at high Reynolds numbers.

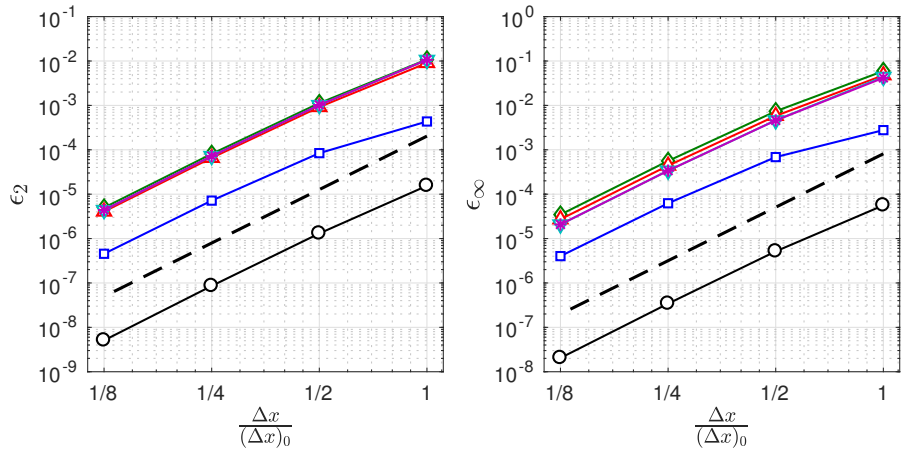


Figure 2.6: Spatial discretization error: (a) L_2-error norm and (b) L_∞-error norm for the following quantities: vorticity ($\circ$), dilatation ($\square$), density ($\diamond$), pressure ($\triangle$), x- ($\triangledown$) and y-components ($*$) of the velocity field. The thick dashed line shows a fourth-order slope.

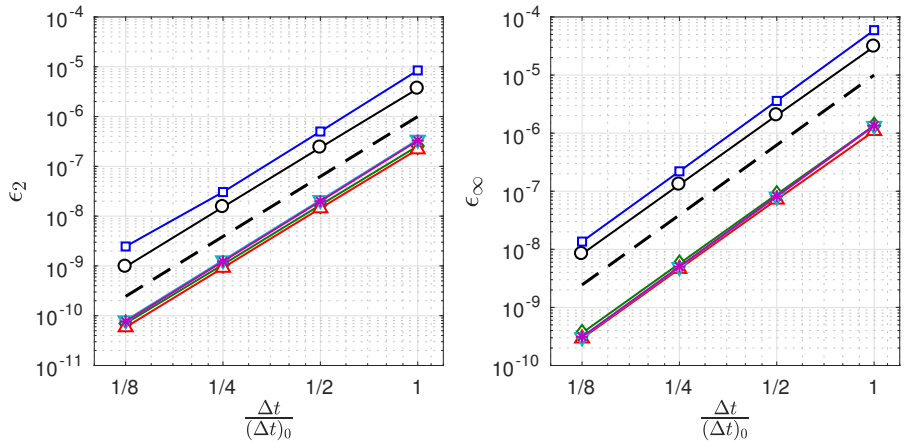


Figure 2.7: Temporal discretization error: (a) L_2-error norm and (b) L_∞-error norm for the following quantities: vorticity ($\circ$), dilatation ($\square$), density ($\diamond$), pressure ($\triangle$), x- ($\triangledown$) and y-components ($*$) of the velocity field. The thick dashed line shows a fourth-order slope.

Convergence study to compare to reference results

A convergence study is carried out using the same indicator as in [50] and [164]: the dilatation field is evaluated at $(x, y) = (0, 1.2R)$ as a function of time, for different resolutions. As in [50], the extent of the domain is here narrowed to $R_\Omega = 2.5R$ and the number of particles placed across the vortex core diameter $2r_0$ are $N_c = 13 \text{ and } 40$. The time step for the first resolution is $\Delta t\, a_\infty/R = 9.0 \cdot 10^{-3}$ whereas that for the finest resolution is $\Delta t\, a_\infty/R = 4.5 \cdot 10^{-3}$. The particles are remeshed every $t_r a_\infty/R = 18 \cdot 10^{-3}$ and no filtering was applied. Figure 2.8 shows the results obtained using the VPM-C solver and reproduces the results obtained by Eldredge et al. [50] and Zhang et al. [164]. Although the results are sensitive to the initialization and simulation parameters, they are also quantitatively in accordance with the reference literature. In fact, we match the phase of the least dispersive method (i.e. the DVP method of Eldredge et al.) and the amplitude of the least dissipative method (i.e. the method used by Zhang et al.); this hints at interesting numerical qualities for our method. Moreover, there is no significant difference between the solutions obtained with the two different resolutions ($N_c = 13$ and $N_c = 40$) when using the VPM-C solver, contrary to the results obtained using the DVP solver of Eldredge et al. The VPM-C solver thus requires fewer points per vortex core diameter to be spatially converged. We then elect to use $N_c = 13$ for the subsequent simulations. The same resolution was used in [50].

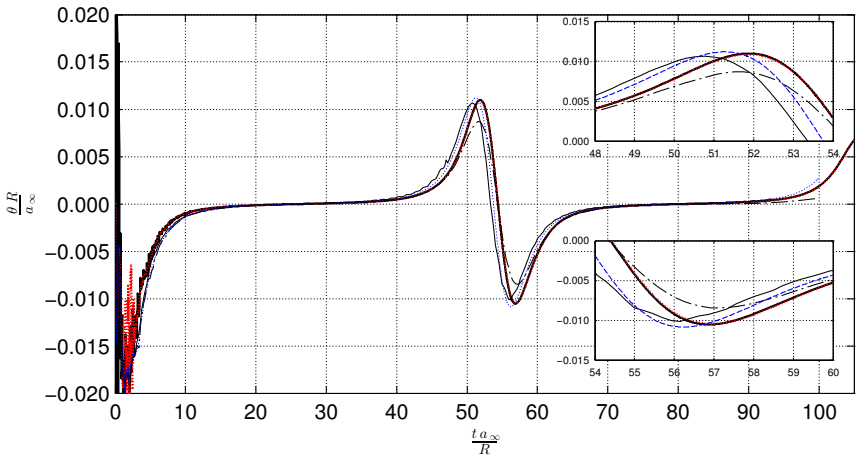


Figure 2.8: Convergence study: dilatation monitored at $(x, y) = (0, 1.2R)$. The results using the VPM-C solver with $N_c = 13$ (thick solid black line) and $N_c = 40$ (thick dotted red line) are almost indistinguishable. The results from Eldredge et al. with $N_c = 13$ (thin solid black line) and $N_c = 40$ (thin dash-dotted black line) and from Zhang et al. (dashed blue line) are also reproduced from [50] and [164]. The inset in the upper right corner is a zoom on the maximum value achieved during the first rotation of the system. The lower right inset is a zoom on the minimum.

Long-time simulation

Previous tests have allowed to verify and validate the VPM-C solver against previous reference results, and to identify a resolution leading to a converged solution. The complete merging process of two co-rotating vortex is now simulated using the VPM-C solver. The results will be compared to those obtained by Eldredge et al. [50] with the DVP solver and by Zhang et al. [164] with their WENO solver.

The radius of the circular computational domain is such that $R_\Omega = 4R$ and the mesh spacing h is chosen such that the number of particles in the core diameter is $N_c = \frac{2r_0}{h} = 13$ as in [50] ; this resolution was found to be sufficient in Section 2.3.2. A time step of $\Delta t\, a_\infty/R = 1.0 \cdot 10^{-2}$ is used. The particles are remeshed and filtered on the Cartesian grid every $n_r = n_f = 2$ time steps. Note that the filtering operation is not necessary to guarantee the stability of the VPM-C method, but it proved to be useful to provide smooth dilatation fields.

Figures 2.9 and 2.10 show the vorticity and dilatation contours at different times up to the axisymmetrisation of the merged vortices. The results exhibit the same vortical and dilatation structures as in [106] and [50]. The first panel shows the situation shortly after the first half-revolution. At that time, the initial acoustic transient has already been released because of the radiation boundary condition. The dilatation field associated to a single vortex exhibits the classical quadrupole configuration. The merging of the vortices occurs after a couple of revolutions.

Figure 2.11 represents the half-separation distance between the vortices – it is denoted as R_{hs}. The center of each vortex is here located using the extrema of the vorticity field. Figure 2.11 confirms that the merging occurs around $t\,a_\infty/R = 500$ as in [50] and [164]. The merging occurred after fewer revolutions in Mitchell et al. [106]. However, as explained in [50], the merging time is very sensitive to the initial condition and to the numerics. The earlier merging observed in [106] is most likely due to numerical dissipation that is higher than in the present VPM-C and DVP [50] results. Of course, the VPM-C and the DVP methods also include some numerical diffusion, due to the remeshing and to the added filtering. We recall that, for VPM-C, this filtering is an operation separate from remeshing and, being explicit its effect can be more controlled.

Evaluation of the far-field pressure fluctuations using the Möhring analogy As demonstrated in [106], the analogy proposed by Möhring [108] may be used to accurately predict the far-field pressure fluctuations in the case of acoustically compact and compact vortical flows at low Mach numbers. It requires the third order time derivative of the second-order moments of the vorticity field, $Q_1 \triangleq 2\int\int x\,y\,\omega\,dxdy$ and $Q_2 \triangleq \int\int \left(y^2 - x^2\right)\omega\,dxdy$. The analogy reads as follows in 2-D

$$p_a(x,y,t) = \frac{\rho_\infty}{8\pi a_\infty^2}\int_0^\infty \left[\dddot{Q}_1(\tau)\cos(2\vartheta) + \dddot{Q}_2(\tau)\sin(2\vartheta)\right] d\xi \tag{2.37}$$

where $\tau = t - (r/a_\infty)\cosh(\xi)$ and (r,ϑ) the polar coordinates. The second-order moments of the vorticity Q_1 and Q_2 are approximated using the results obtained with the VPM-C solver. A quadrature is performed on the particles located in the circular domain Ω of radius R_Ω, $Q_1 \simeq 2\sum_p x_p y_p\,\Gamma_p$ and $Q_2 \simeq \sum_p (y_p^2 - x_p^2)\,\Gamma_p$. The methodology is here assessed by comparing the far-field pressure fluctuations obtained using the Möhring analogy with the here obtained Q_1 and Q_2 against the results from Eldredge et al. [50] using the same analogy. In their paper, Eldredge et al. validated their results against the

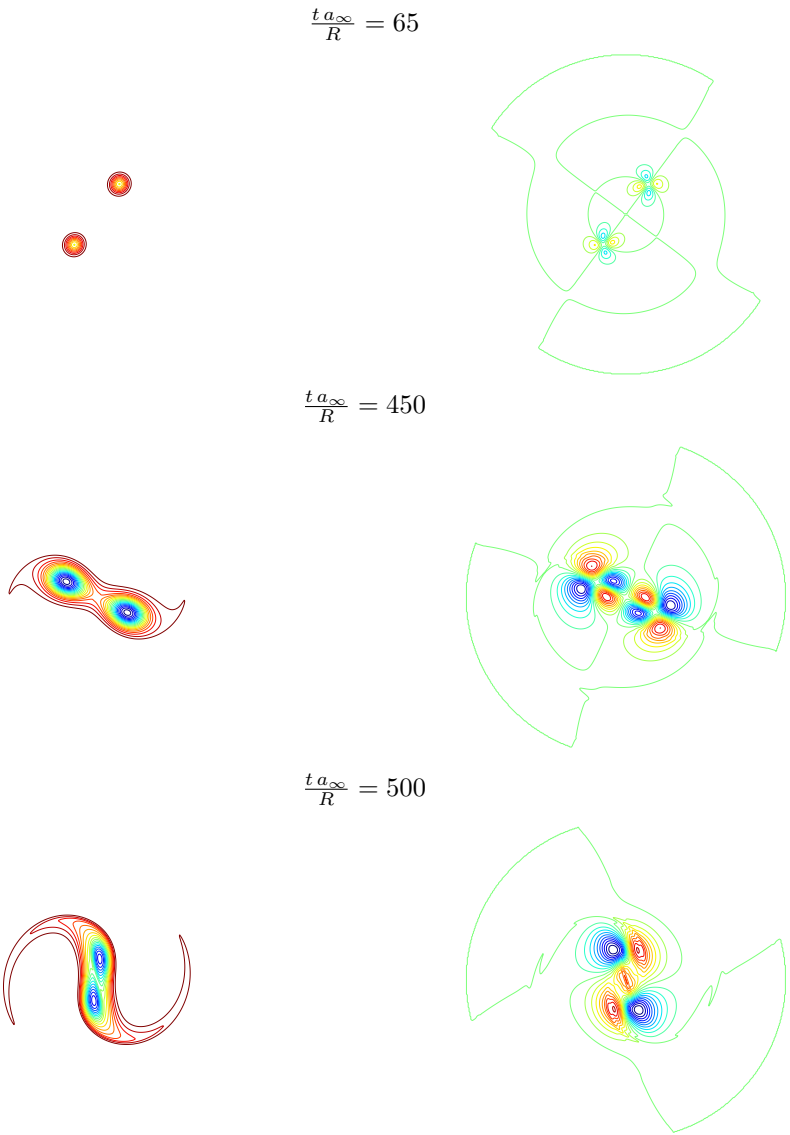


Figure 2.9: Co-rotating vortex pair: Vorticity $\omega R/U_0$ (left) and dilatation $\theta R/U_0$ contours (right) at $t\,a_\infty/R = 65, 450, 500, 550, 600$ and 700. The vorticity contours are in the range [-23, -1] in increments of $\Delta(\omega R/U_0) = 1$ until $t\,a_\infty/R = 65$ and in $[-2, -0.05]$ with $\Delta(\omega R/U_0) = 0.103$ for the subsequent times, as in [50]. Dilatation contours are in the range $[-0.06, 0.06]$ in increments of 0.006 until $ta_\infty/R = 65$ and in $[-0.005, 0.005]$ in increments of 0.0005.

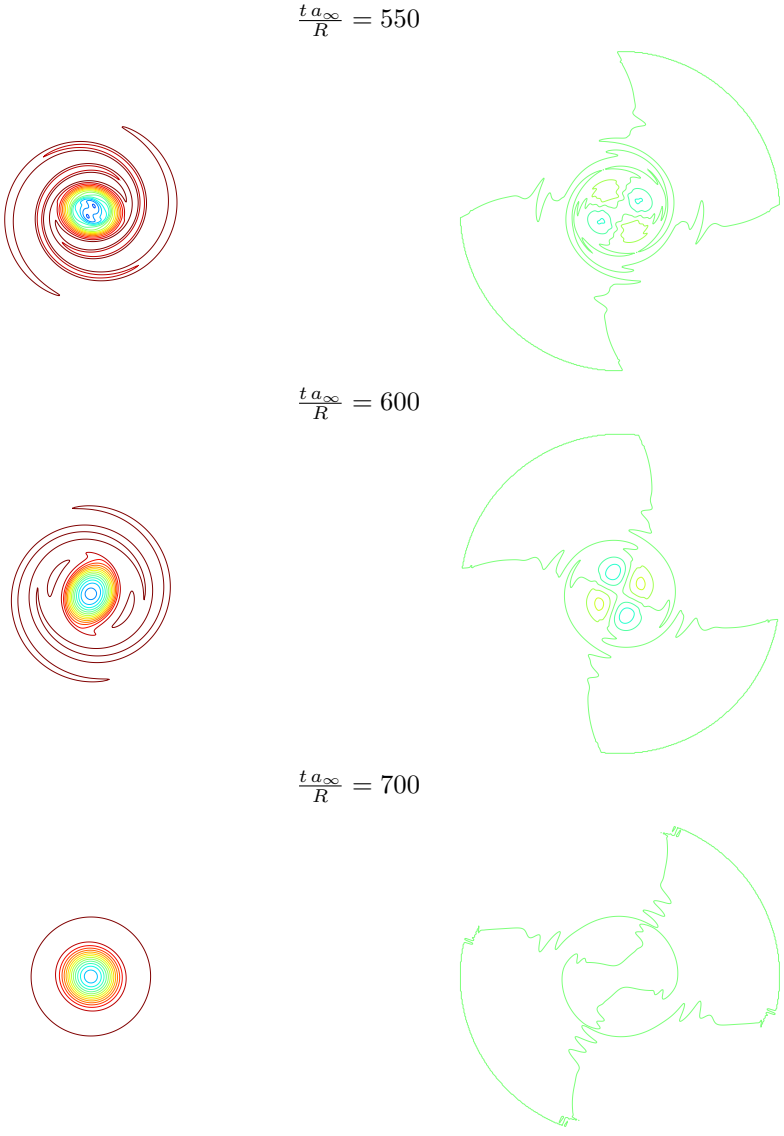


Figure 2.10: Co-rotating vortex pair: Continued

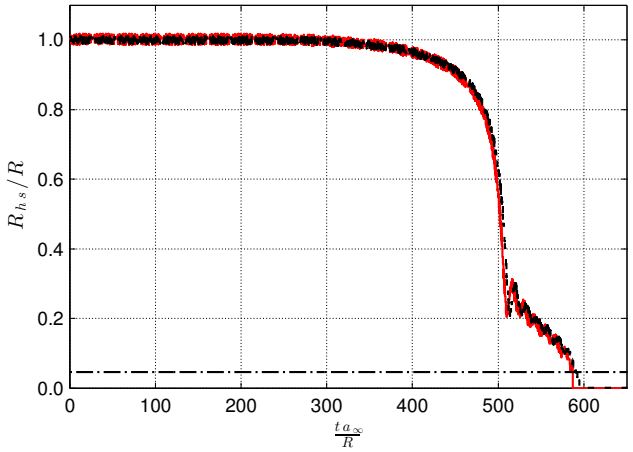
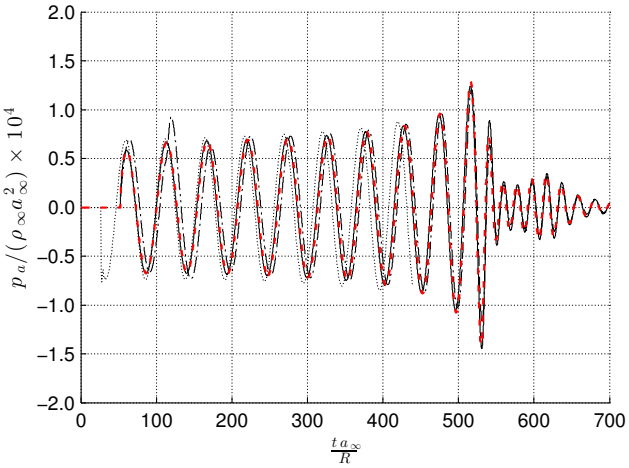
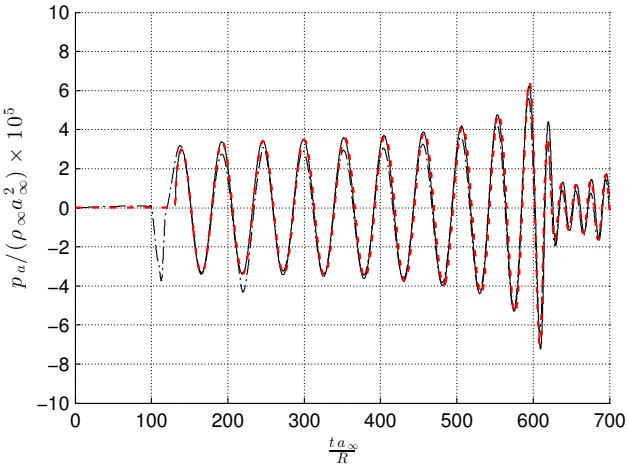


Figure 2.11: Co-rotating vortex pair: Temporal evolution of the half-separation distance using the VPM-C solver (solid red line). The dash-dotted line indicates the threshold ($2h$) below which the distance can not be measured. The results from [164] are reproduced (dashed black line).

results of Mitchell et al. [106] and also showed that they were in good accordance with the results obtained using a Kirchoff surface. The results are also compared to the results of [164] obtained using the WENO solver only, on the one hand, (thus using an extended computational domain to perform the DNS of the far-field acoustics), and using the WENO solver with the Möhring analogy, on the other hand. Figure 2.12 shows the far-field pressure fluctuations predicted at $(x, y) = (0, \frac{1}{2}\lambda)$ and $(x, y) = (0, 2\lambda)$, with λ the fundamental wavelength of the sound, which is here equal to $\frac{\lambda}{R} = 52.5$ (hence indeed the present problem is acoustically compact). The Möhring analogy, Eq. (2.37), was computed numerically following the procedure introduced in [48]. Following Mitchell et al. [106], the upper bound of the integral was chosen as $cosh^{-1}\left(\frac{(t-t_0)a_\infty}{r}\right)$ with $t_0 = \tau/4$, $\frac{\tau a_\infty}{R} = 105$ being the period of rotation based on the initial velocity field. This choice of t_0 allowed the initial acoustic transient to leave the near-field region, and even to reach the nearest pressure probe. Here again, the results obtained with the VPM-C method agree quantitatively with the results from the literature, in terms of both magnitude and phase. They are found to be in excellent agreement throughout the complete merging process with those obtained using the WENO solver together with the Möhring analogy.



(a) Pressure fluctuations measured at $(x, y) = (0, \frac{1}{2}\lambda)$.



(b) Pressure fluctuations measured at $(x, y) = (0, 2\lambda)$.

Figure 2.12: Co-rotating vortex pair: Pressure fluctuations measured at $(x, y) = (0, \frac{1}{2}\lambda)$ (a) and at $(x, y) = (0, 2\lambda)$ (b). VPM-C solver and Möhring analogy (thick dashed red line), DVP solver and Möhring analogy [50] (Fig. (a) only, dotted line), WENO solver and Möhring analogy (solid line), WENO solver (DNS) [164] (dash-dotted line).

2.4 Conclusions

In this chapter, we introduced a Vortex Particle-Mesh method solving the full compressible Navier-Stokes equations in 2-D. The solver is supplemented with a radiation boundary condition and allows the simulation of compressible 2-D flows in an unbounded domain. It is capable of simulating viscous compressible flows in a subsonic regime. A shock capturing methodology would be required to extend the range of application of the solver to the transonic regime. This Eulerian-Lagrangian solver follows the evolution of integrated quantities over a material volume: vorticity, dilatation, density and internal energy fields. The velocity field is calculated using the Helmholtz decomposition, which requires the solution of two Poisson equations. The spatial and temporal orders of convergence were verified. The boundary condition, which is directly derived from the work of Tam and Webb [150], was validated against the analytical solution of the acoustic pulse problem. The method was then applied to an aeroacoustic problem; the study of the sound generation process of two co-rotating Gaussian vortices. This test case demonstrated the capability of the solver to accurately capture the flow dynamics, also during the merging process, as well as the sound generation. The results were validated against existing reference results found in the literature obtained using either grid-based [106, 164] or remeshed particle-based [50] high-order solvers and were found to be in excellent agreement.

When compared to the remeshed Lagrangian solver of Eldredge et al. [50], the adjunction of an Eulerian grid is beneficial. The underlying grid indeed allows the use of efficient grid-based Poisson solvers. Efficiency analysis performed by Cottet and Poncet [41] and by Cocle et al. [31] illustrate the speed-up provided by a VPM method in comparison to a grid-free method. The grid-based data structures provide data-alignment and efficient vectorizing loops, which directly translates into a computational efficiency higher than for the FMM method alone. We here elected to use a Fourier-based Poisson solver. Given the level of optimization in modern implementation of FFTs, this is even more efficient than the method used in the VPM solver in [41] and the method used in the combined VPM-PFM solver in [31] when applied on a single domain. The underlying grid also allows to efficiently evaluate the spatial differential operators on the grid, whereas the Lagrangian solver in [50] relied on PSE-type integral operators. As the latter requires to perform a nearest neighbours finding, an expensive operation especially when the remeshing operation is not performed at each time step, it is thus anticipated that the hybrid scheme should prove more computationally efficient. The willingness shown

by Eldredge et al. to achieve a high-order accuracy is here maintained, as least as far as the spatial and temporal discretization techniques are concerned (yet, with the caution that the remeshing kernel is in principle second order accurate). High-order methods are indeed more efficient than low order methods in the context of computational aeroacoustics [35]. Some attention was also given to the degree of isotropy of the numerical schemes. Finally, a separate implementation of the remeshing and the filtering operators allows to regain some flexibility on the application frequency of the filtering operation. In that respect, our implementation departs from the combined remeshing-filtering operation implemented in [50]. In our implementation, the filtering operation can thus be performed more sparingly. It can even be deactivated if it turns out to be superfluous, as in the test case of the radiation of an acoustic pulse. It also allows the use of recent high-order interpolation schemes – we here elected to use the M_6^* scheme derived in [11] – and of efficient high-order discrete filters obtained by iteration of a simple compact filter [80]. Moreover, this separate implementation also allows to gain a better control on the numerical diffusion induced by the filtering operation.

Radiation boundary conditions are required for the present simulations. Those are typically based on the asymptotic solution of the linearized Euler equations. They belong to the larger group of conventionally called Linearized BCs – together with the Nonreflecting BCs and the Perfectly Matched Layer [35]. The enforcement of radiation boundary conditions only requires a boundary region that is a few particles wide to be compatible with the M2P operation. It is expected that a buffer region or a damping function (as in Papadakis and Voutsinas [118, 117] for the more general case of flow past an airfoil) would have required a larger damping zone to achieve a similar performance. Nonetheless, in order to be able to treat more general problems, such as the flow past bodies, future work will be devoted to the inclusion of inflow and outflow conditions in the formulation.

At this stage, the case for a vortex method for compressible flows is a bit weaker than with a vortex method for incompressible flows when they are compared to state-of-the-art Eulerian methods. First, the benefits of a compact computational domain due to the compactness of the vorticity field are here lost: the particles here follow the evolution of non-compact fields on top of the compact vorticity field. Then, the necessary truncation of the dilatation field is responsible for a (controlled) loss of accuracy in the computation of the velocity field as it is the source term of the Poisson equation 2.9. Therefore, the level

of truncation is a parameter that needs to be carefully chosen. As suggested in [50], a sensitivity analysis may be used to verify that the results remain essentially unchanged when using a larger computational domain. Finally, the relaxed stability constraints due to the Lagrangian treatment of the advection in vortex methods for incompressible flows are here retightened: the advection is still carried out in a Lagrangian fashion but the acoustic components of the flow also have to be resolved. A CFL-like constraint is thus restored. It would thus be interesting to compare the robustness and efficiency of the VPM method for compressible flows with state-of-the-art Eulerian methods for a given level of accuracy in order to further shed light on that matter.

Each and every tool (e.g. the radiation condition, the unbounded Poisson solver) used in the present 2-D compressible flow VPM solver may be extended to the 3-D case. Features that are inherent to the 3-D equations or the numerical method will then need to be handled: the zero-divergence problem of the vorticity field as well as the evaluation of the stretching term. Those can be treated as efficiently and accurately as in the incompressible flow VPM solver using well-known techniques.

CHAPTER 3

A VPM SOLVER WITH A BRINKMAN PENALIZATION FOR COMPRESSIBLE FLOWS

In this chapter, the VPM solver for compressible flows presented in Chapter 2 is complemented with a methodology allowing to take an obstacle into account. This addition may prove necessary in the coupling algorithm presented in Chapter 4. Indeed, in order to provide good quality boundary conditions to the near-body Eulerian solver, the solution of the Lagrangian solver is required to remain of acceptable quality near the outer boundary of the Eulerian domain between two correction steps. A crude evaluation of the RHS terms of Eqs. (2.2)-(2.5) near the obstacle would lead to unphysical gradients and could rapidly impair the quality of the boundary conditions provided to the Eulerian solver, especially if the Eulerian domain is narrow. The most critical aspect of these unphysical gradients at the wall is their translation into unphysical values for the dilatation and/or the vorticity fields that would then generate erroneous boundary conditions at the outer boundary of the Eulerian domain, as an elliptic solver is used to compute the velocity field in the VPM *solver.*

3.1 Introduction

The most challenging aspect in a VP or VPM method still remains to deal with solid obstacles. All the techniques developed in the context of vortex methods for arbitrarily shaped geometry rely on a non-conformal approach; the mesh

is allowed to intersect the boundaries of the obstacle(s). Contrary to the conformal or body-fitted approach, the potentially strenuous and expensive process of mesh generation is thus avoided; yet the mesh needs to be finer. Early developments relied on vorticity flux and panel methods [88, 128, 129]. Other approaches rely on an immersed boundary approach, a comprehensive review of which can be found in the paper of Mittal and Iaccarino [107]. Belonging to the "continuous forcing approach" subclass in [107], the volume penalization method originally proposed by Arquis and Caltagirone [6] and later mathematically supported by Angot [4] and Angot et al. [5] was also adapted to vortex methods for incompressible flows. It was used in numerous works (for example, [36, 135, 132, 56, 67, 105, 59]). Most recent approaches [103, 104, 60] concentrate on the Immersed Interface Method (IIM) initially introduced by Leveque and Li [95]. This method relies on the modification of the finite difference stencils, the interpolation schemes and the elliptic and parabolic solvers near the solid boundaries in order to take these into account in a manner consistent across all components of the method.

We here decided to rely on the more mature and simpler penalization method. Still, the topic constitutes an active research area as this adaptation in a vorticity-velocity formulation comes with its own set of challenges [132, 67, 59]. Moreover, on top of these challenges, the penalization method also exhibits an important shortcoming: the regularization (of the singular vortex sheet in particular) caused by the discretization of the continuous volume forcing term smears the solution near the interface. As explained in [103] and references therein, this can eventually lead to a loss of accuracy and could render the method inappropriate to study high Reynolds number flows as the boundary layer will not be captured accurately. However, this should not be seen as a definitive hindrance to the avowed aim of this work to solve high Reynolds number (and moderate Mach number) flows; an Eulerian solver using a body-fitted approach will indeed be used in the coupling methodology to accurately capture the near-body region and the boundary layer in particular. This thus reduces the role of the penalization method; it is here dedicated to helping the VPM solver provide proper boundary conditions to the Eulerian solver.

In [99], Liu and Vasilyev proposed an extension of the Brinkman penalization method to compressible flows. Their approach is based on a mathematical model for compressible flows through porous media. Contrary to previous contributions, the continuity equation is also modified in order to recover the mass conservation equation for porous media inside the obstacle.

Liu and Vasilyev showed that this allowed to avoid non-physical wave transmission inside the obstacle. Error estimates were obtained through an asymptotic analysis for the case of the wave propagation in a fluid-porous medium. They also were numerically verified.

Boiron et al. [15] developed another penalization method combined with a shock-capturing technique to treat higher Mach number shocked flows. In their paper, the penalization terms are expressed in terms of the conservative variables $(\rho, \rho\mathbf{u}, \rho E)$. An asymptotic analysis allowed them to show that the proper Dirichlet boundary conditions were indeed imposed and that the density remains constant inside the obstacle. A constant density boundary condition inside the obstacle is in contradiction with what was claimed by Liu and Vasilyev in [99]. In [15], the results were compared to results obtained with Ansys Fluent™ and, as acknowledged by the authors, the accuracy of the resulting solver was not considered.

Piquet et al. [127] also implemented a Brinkman penalization method using the conservative variables in the context of high Mach number flows. Contrary to Boiron et al. [15], they retained the penalization term in the mass conservation of Liu and Vasilyev [99]. Through a comparison with the analytical solution of a shock wave reflection from a solid wall, they showed that the mass conservation equation should indeed be penalized.

While the three previous methods only implement Dirichlet boundary conditions, Brown et al. [18] proposed a penalization technique differing from the classical approach called a Characteristic-Based Volume Penalization (CBVP) method. This method is also able to impose Neumann and Robin boundary conditions.

In this chapter, the VPM solver for compressible flows presented in Chapter 2 will be fitted with the Brinkman penalization method of Liu and Vasilyev [99]. It will also rely on some aspects developed by Piquet et al. [127]. The developed methodology will also rely on existing techniques for VPM methods for incompressible flows. Indeed, the developments will require to formulate the penalization sub-problem(s) in terms of the variables used in the VPM solver. Two time stepping techniques will be considered to integrate the penalization terms: an explicit and an implicit time stepping. The methodology will be presented in Section 3.2 for a single obstacle but it could readily be applied to flows with multiple obstacles. Verification and valida-

tion numerical tests are then carried out on classical 2-D benchmark problems in Section 3.3.

3.2 Methodology

We here briefly recall the main aspects of the combination of remeshed vortex methods with the Brinkman penalization technique for incompressible flows. We then apply a similar methodology to fit the VPM solver of Chapter 2 with the Brinkman penalization for compressible flows developed by Liu and Vasilyev [99].

3.2.1 Vortex Particle-Mesh method and Brinkman penalization for incompressible flows

The Brinkman penalization technique [5, 83] allows to impose the velocity at the surface of a body (either a no-slip condition or a prescribed motion) by adding a forcing term in the momentum equation for incompressible flows

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u} - \lambda \chi_b (\mathbf{u} - \mathbf{u}_b) \tag{3.1}$$

with λ the penalization factor. The characteristic function (or mask function), χ_b, is used to represent the volume Ω_b occupied by the obstacle

$$\chi_b(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_b \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The evolution equation for the vorticity including the additional penalization term is obtained by taking the curl of Eq. (3.1) and reads

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\nabla \mathbf{u}) \cdot \boldsymbol{\omega} + \nu \nabla^2 \boldsymbol{\omega} - \lambda \nabla \times \big(\chi_b (\mathbf{u} - \mathbf{u}_b)\big) \tag{3.3}$$

with ν the cinematic viscosity of the fluid, $\mathbf{u}_b$ the velocity inside the obstacle and λ the penalization parameter. VPM solvers used in conjunction with a penalization technique generally use a splitting approach for the time integration [56]; the position and the circulation of the particles are updated without actual knowledge of the obstacle and are subsequently corrected. Partly using the notations introduced in [59] and [56], the correction step reads

$$\boldsymbol{\omega}_\lambda^{n+1} = \boldsymbol{\omega}^{n+1} + \delta\boldsymbol{\omega} = \boldsymbol{\omega}^{n+1} + \mathcal{P}(\mathbf{u}_b, \mathbf{u}^{\mathbf{n+1}}) \tag{3.4}$$

with $\boldsymbol{\omega}_\lambda^{n+1}$ the penalized vorticity and $\mathcal{P}$ a correction function varying from one methodology to another.

As explained in [59], a widespread strategy [36, 135, 56, 105] consists in using an implicit Euler scheme to integrate the penalization sub-problem. The unconditional stability of the implicit Euler scheme then allows to employ an arbitrarily high penalization parameter, λ. The velocity is then penalized using

$$\mathbf{u}_\lambda^{n+1} = \frac{\left(\mathbf{u}^{n+1} + \lambda\Delta t\ \chi_b\ \mathbf{u}_b\right)}{(1 + \lambda\Delta t\ \chi_b)} \tag{3.5}$$

and one updates the vorticity using the penalized velocity,

$$\boldsymbol{\omega}_\lambda^{n+1} = \nabla \times \mathbf{u}_\lambda^{n+1} \tag{3.6}$$

corresponding to the correction function

$$\mathcal{P}(\mathbf{u}_b, \mathbf{u^{n+1}}) = \left(\nabla \times \mathbf{u}_\lambda^{n+1}\right) - \boldsymbol{\omega}^{n+1} \tag{3.7}$$

The unconditional stability of this approach clearly constitutes an advantage but the method is not without drawbacks as recalled in [67, 59]:

1. this procedure introduces substantial numerical diffusion due to the replacement of the vorticity field by the finite difference approximation of the implicitly penalized velocity field. This was originally claimed by Rasmussen et al. [132]. A procedure allowing to alleviate this problem while retaining the implicit integration will be presented hereunder.

2. the correction procedure will not ensure that the penalized velocity and vorticity field will still verify the Poisson equation that globally (or elliptically) links them [67]. To capture the correct vortex sheet of an impulsively started flow, this calls either for a prohibitively small time step or an iterative penalization technique as in [67, 59].

To circumvent the first drawback, Rasmussen et al. [132] formulated the problem as a localized correction rather than a global one by using an explicit penalization and $\lambda\Delta t = 1$. The corresponding correction function is then

$$\mathcal{P}(\mathbf{u}_b, \mathbf{u^{n+1}}) = \nabla \times \left(\lambda\Delta t\ \chi_b(\mathbf{u}_b - \mathbf{u}^n)\right). \tag{3.8}$$

Note that a localized correction may actually also be employed when using an implicit integration. Equation (3.6) may indeed be recast as

$$\begin{aligned}
\boldsymbol{\omega}_\lambda^{n+1} &= \nabla \times \mathbf{u}_\lambda^{n+1} + (\boldsymbol{\omega}^{n+1} - \nabla \times \mathbf{u}^{n+1}) && (3.9)\\
&= \boldsymbol{\omega}^{n+1} + \nabla \times \left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1}\right) && (3.10)\\
&= \boldsymbol{\omega}^{n+1} + \nabla \times \delta\mathbf{u} && (3.11)\\
&= \boldsymbol{\omega}^{n+1} + \delta\boldsymbol{\omega} && (3.12)
\end{aligned}$$

Although Equations (3.6) and (3.10) are mathematically equivalent, the numerical counterpart of Eq. (3.6) provides results of vastly inferior quality to those from the evaluation of Eq. (3.10). While it is not expressly stated in [135, 56, 105], the methodology working with the velocity difference (Eq. (3.10)) was most likely also implemented. The correction function for the implicit time stepping strategy, Eq. (3.7), is then rewritten as

$$\mathcal{P}(\mathbf{u}_b, \mathbf{u}^{\mathbf{n+1}}) = \nabla \times \left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1} \right) \tag{3.13}$$

To circumvent the second drawback, Hejlesen et al. [67] and, more recently Gillis et al. [59], extended the work of Rasmussen et al. [132]. They used an iterative approach allowing to find the correction $\delta\boldsymbol{\omega}$, or equivalently the vortex sheet such that $\mathbf{u}^{n+1} + \delta\mathbf{u} = \mathbf{u}_\lambda^{n+1} = \mathbf{u}_b$ inside the obstacle and such that the velocity and the vorticity fields are consistent.

The explicit approach comes with a stability constraint

$$0 \leq \lambda\Delta t \leq 2 \tag{3.14}$$

when using an explicit Euler time stepping technique. It even requires $0 \leq \lambda\Delta t \leq 1$ to exhibit a monotonic convergence, the value $\lambda\Delta t = 1$ being the best value as it provides the ideal value for the amplification factor, $\rho_{EE} = 1 - \lambda\Delta t = 0$. This choice corresponds to that of Rasmussen et al. [132]. On the contrary, the implicit scheme is unconditionally stable and monotonically convergent. Its amplification factor is given by $\rho_{EI} = \frac{1}{1+\lambda\Delta t}$.

3.2.2 Vortex Particle-Mesh method and Brinkman penalization for compressible flows

In [99], Liu and Vasilyev proposed an extension of the Brinkman penalization method to compressible flows. Their approach is based on a mathematical model for compressible flows through porous media. Contrary to previous contributions, the continuity equation for porous media is there also considered inside the obstacle. Liu and Vasilyev showed that this allowed to avoid non-physical reflections. The dimensionless penalized equations are

then given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = -\left(\left(\frac{1}{\phi} - 1\right)\chi_b\right)\nabla \cdot (\rho \mathbf{u}) \tag{3.15}$$

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \frac{1}{Re_a}\nabla \cdot \underline{\underline{\boldsymbol{\tau}}} - \frac{\chi_b}{\eta}\left(\mathbf{u} - \mathbf{u_b}\right) \tag{3.16}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\rho E) + \nabla \cdot \big((\rho E + p)\mathbf{u}\big) = \quad & \frac{1}{Re_a}\nabla \cdot (\mathbf{u} \cdot \underline{\underline{\boldsymbol{\tau}}}) \\ & + \frac{1}{Re_a Pr(\gamma - 1)}\nabla \cdot (\mu \nabla T) \\ & - \frac{\chi_b}{\eta_T}(T - T_b)\end{aligned} \tag{3.17}$$

with ϕ the porosity, $\eta = \alpha\phi$ and $\eta_T = \alpha_T\phi$ the dimensionless viscous and thermal permeabilities, respectively. The non-dimensionalization used in [99] is identical to the acoustic non-dimensionalization introduced in Chapter 2 except for the temperature. In [99], the temperature is non-dimensionalized by a reference temperature T_0. It is thus related to the dimensionless temperature $\widetilde{T} = (c_p T)/c_0^2$ used in the present contribution by $\widetilde{\widetilde{T}} = T/T_0 = (\gamma - 1)\tilde{T}$. Dirichlet boundary conditions are enforced on the velocity and the temperature

$$\mathbf{u} = \mathbf{u}_b \quad \text{on} \quad \partial\Omega_b \tag{3.18}$$

$$T = T_b \quad \text{on} \quad \partial\Omega_b \tag{3.19}$$

where $\mathbf{u}_b$ and T_b are the velocity and the temperature of the obstacle, respectively.

In their paper, Liu and Vasilyev showed that the porosity, ϕ, should be taken sufficiently small in order to minimize nonphysical wave transmissions inside the obstacle. An asymptotic analysis for the case of acoustic wave propagation in the small amplitude limit was also performed. It allowed them to establish error bounds for the general subsonic case and to provide guidelines for the choice of the three penalization parameters (the porosity, ϕ, and the permeabilities, η and η_T):

1. An intermediate region inside the porous media, which is here referred to as the "transition" region[1], was necessary to consider to match the solutions in the fluid and porous media regions in the asymptotic analysis. The amplitude and phase errors of the method were shown to depend on the level of resolution in that transition region:

[1] In [99], the authors refers to this region as a "boundary layer".

- the errors are of order $\mathcal{O}\left(\alpha^{1/2}\phi\right)$ when the transition region is resolved,
- the errors are of order $\mathcal{O}\left((\frac{\eta}{\eta_T})^{1/4}\phi^{3/4}\right)$ when the transition region is unresolved.

As claimed in [99], these rates can be viewed as error bounds thus requiring small values either for $(\alpha^{1/2}\phi)$ or $(\frac{\eta}{\eta_T})^{1/4}\phi^{3/4}$ depending on the mesh resolution. The porosity is in both cases the parameter allowing to control the error in the most effective way. Moreover, it is also claimed that it is unnecessary to resolve the transition region within the porous medium in order to get accurate results.

2. The magnitude of the permeabilities, $\eta = \alpha\phi$ and $\eta_T = \alpha_T\phi$, should be chosen sufficiently small in order to properly enforce the boundary conditions. It indeed ensures that the damping inside the porous medium is significantly faster than the time scales associated with the external flow.

They validated the method on a series of 1-D and 2-D benchmark problems and showed that the results were in excellent agreement with analytical solutions.

The porting of this penalization method into a VPM solver for compressible flows will now be presented. It will require to formulate the penalization sub-problem(s) in terms of the variables used in the VPM solver, similarly to what was presented in Section 3.2.1 for a VPM solver for incompressible flow.

Using an explicit integration scheme

Following the same methodology as in Chapter 2, the penalized equations for the vorticity and the dilatation field are obtained by taking the curl and the divergence of the momentum equation, respectively. Equations (2.2)-(2.5) are supplemented by the penalization terms and non-dimensionalized using the non-dimensionalization defined in Chapter 2 to read[2]

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = \mathcal{L}_\rho(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) - \left(\left(\frac{1}{\phi} - 1 \right) \chi_b \right) \nabla \cdot (\rho \mathbf{u}) \tag{3.20}$$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \cdot (\boldsymbol{\omega} \mathbf{u}) = \mathcal{L}_{\boldsymbol{\omega}}(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) - \frac{1}{\eta} \nabla \times \left(\chi_b \left(\frac{\mathbf{u} - \mathbf{u}_b}{\rho} \right) \right) \tag{3.21}$$

$$\frac{\partial \theta}{\partial t} + \nabla \cdot (\theta \mathbf{u}) = \mathcal{L}_\theta(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) - \frac{1}{\eta} \nabla \cdot \left(\chi_b \left(\frac{\mathbf{u} - \mathbf{u}_b}{\rho} \right) \right) \tag{3.22}$$

$$\frac{\partial}{\partial t} (\rho T) + \nabla \cdot (\rho T \mathbf{u}) = \mathcal{L}_{\rho T}(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) - \gamma (\gamma - 1) \frac{\chi_b}{\eta_T} (T - T_b) \tag{3.23}$$

where $\mathcal{L}_\rho$, $\mathcal{L}_{\boldsymbol{\omega}}$, $\mathcal{L}_\theta$ and $\mathcal{L}_{\rho T}$ are respectively the dimensionless right-hand side terms of the non-penalized conservation equations for the vorticity, dilatation and pressure. For the sake of clarity and completeness, they are here recalled

$$\mathcal{L}_\rho(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) = 0 \tag{3.24}$$

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\omega}}(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) = (\nabla \mathbf{u}) \cdot \boldsymbol{\omega} + \frac{(\gamma - 1)}{\gamma} \frac{\nabla \rho \times \left(\nabla (\rho T) \right)}{\rho^2} \\ + \frac{1}{Re_a} \nabla \times \left(\frac{1}{\rho} \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \right) \end{aligned} \tag{3.25}$$

$$\begin{aligned} \mathcal{L}_\theta(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) = \left(\theta^2 - \nabla \mathbf{u} : (\nabla \mathbf{u})^T \right) - \frac{(\gamma - 1)}{\gamma} \nabla \cdot \left(\frac{\nabla (\rho T)}{\rho} \right) \\ + \frac{1}{Re_a} \nabla \cdot \left(\frac{1}{\rho} \nabla \cdot \underline{\underline{\boldsymbol{\tau}}} \right) \end{aligned} \tag{3.26}$$

$$\mathcal{L}_{\rho T}(\mathbf{F}, \mathbf{u}, \mathbf{x}, t) = -a^2 \rho \theta + \frac{\gamma}{Re_a} \phi + \gamma \frac{1}{Re_a Pr} \nabla^2 T \tag{3.27}$$

In this section, we wish to retain the strategy of a localized correction proposed in Rasmussen et al. [132]. Using the same notations as in Section 3.2.1, a vector correction function $\delta \mathbf{F} = \boldsymbol{\mathcal{P}}(\mathbf{F}, \mathbf{u}, \mathbf{u}_b, T_b)$ similar to Eq. (3.8) is here

[2]One could argue that the '-1' facing $\frac{1}{\phi}$ in the continuity equation should disappear when using a Lagrangian solver. It is, however, not really important as $\frac{1}{\phi} \gg 1$. It may even prove useful to progressively activate the penalization term in the continuity equation using a ramp starting from $\phi = 1$ to the desired value.

obtained[3]

$$\boldsymbol{\mathcal{P}} = \begin{pmatrix} \mathcal{P}_\rho \\ \mathcal{P}_{\boldsymbol{\omega}} \\ \mathcal{P}_\theta \\ \mathcal{P}_{\rho T} \end{pmatrix} (\mathbf{F}, \mathbf{u}, \mathbf{u}_b, T_b) = \begin{pmatrix} -\left(\left(\frac{1}{\phi} - 1\right)\chi_b\right)\nabla\cdot(\rho\mathbf{u}) \\ \frac{1}{\eta}\nabla\times\left(\chi_b\left(\frac{\mathbf{u}_b-\mathbf{u}}{\rho}\right)\right) \\ \frac{1}{\eta}\nabla\cdot\left(\chi_b\left(\frac{\mathbf{u}_b-\mathbf{u}}{\rho}\right)\right) \\ \gamma\left(\gamma-1\right)\frac{\chi_b}{\eta_T}\left(T_b - T\right) \end{pmatrix} \tag{3.28}$$

The characteristic function (3.2) is here not mollified. As explained by Hejlesen et al. [67], the high order Poisson solver from [68] also implemented in this solver ensures the smoothness of the velocity field ($\mathbf{u} \in \mathcal{C}^2$). As a consequence, Hejlesen et al. stated that the explicit mollification required in previous studies [135, 132, 56] is not necessary. The RHS terms are evaluated on the grid using finite differences of the same order for all terms. The equations are advanced in time using the same temporal scheme for all RHS terms. As explained in Section 3.2.1, the explicit integration of the penalization terms limits the time step for a given penalization intensity. As we here use a RK4C temporal scheme to integrate the penalization terms as well, the stability constraint is here slightly relaxed. However, the complete alleviation of this constraint calls for an implicit integration of the penalization terms.

Using an implicit integration scheme

Contrary to the previous section, a genuine splitting approach is here implemented. The unpenalized equations are first advanced in time using a RK4C time stepping scheme. The velocity field, $\mathbf{u}^{n+1}$, as well as the grid fields, $\mathbf{F}^{n+1}$, associated to the strengths of the particles are then corrected on the Lagrangian mesh. Without lack of generality, we here suppose that the particles are remeshed before the penalization step[4] The procedure is here described with the support of Algorithm 1.

The penalization is first applied to the momentum equation

$$\frac{\partial(\rho\mathbf{u})}{\partial t} = -\lambda\chi_b\left(\rho\mathbf{u} - (\rho\mathbf{u})_b\right), \tag{3.29}$$

which is integrated using the implicit Euler scheme and the boundary condition on the velocity, $\mathbf{u}_b = 0$ (see Eq. (Alg. 1.1) in Algorithm 1). The mass

[3]Note that it is here provided as a real-valued function while Eq. (3.8) is evaluated at discrete times. Equation (3.28) will indeed be treated using the same integration scheme as the one used for the RHS terms of the Navier-Stokes equations. At this stage, it is thus not yet evaluated at discrete times.

[4]Should the remeshing period be different from the unity, the penalization terms evaluated on the grid would be projected on the particles using a M2P operation.

Algorithm 1: Implicit penalization

Given: $\mathbf{u}^{n+1},\ \mathbf{F}^{n+1},\ \mathbf{u}_b = 0,\ T_b$
Compute:

$$(\rho\mathbf{u})_\lambda^{n+1} = \frac{(\rho\mathbf{u})^{n+1}}{(1+\lambda\Delta t\chi_b)} \tag{Alg. 1.1}$$

$$\rho_\lambda^{n+1} = \rho^{n+1} - \Delta t\left(\left(\frac{1}{\phi}-1\right)\chi_b\right)\nabla\cdot\left((\rho\mathbf{u})_\lambda^{n+1}\right) \tag{Alg. 1.2}$$

$$\mathbf{u}_\lambda^{n+1} = \frac{(\rho\mathbf{u})_\lambda^{n+1}}{\rho_\lambda^{n+1}} \tag{Alg. 1.3}$$

$$\boldsymbol{\omega}_\lambda^{n+1} = \boldsymbol{\omega}^{n+1} + \nabla\times\left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1}\right) \tag{Alg. 1.4}$$

$$\theta_\lambda^{n+1} = \theta^{n+1} + \nabla\cdot\left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1}\right) \tag{Alg. 1.5}$$

$$(\rho T)_\lambda^{n+1} = \rho_\lambda^{n+1}\left(\frac{T^{n+1} + \lambda_T\Delta t\ \chi_b\ T_b}{1+\lambda_T\Delta t\ \chi_b}\right) \tag{Alg. 1.6}$$

conservation equation is in turn penalized using the penalized momentum, $(\rho\mathbf{u})_\lambda^{n+1}$, using Eq. (Alg. 1.2). A similar methodology was employed by Piquet et al. in [127]. The penalized velocity field is then evaluated using Eq. (Alg. 1.3). The vorticity and dilatation fields can now be computed using

$$\boldsymbol{\omega}_\lambda^{n+1} = \nabla\times\mathbf{u}_\lambda^{n+1}, \tag{3.30}$$

$$\theta_\lambda^{n+1} = \nabla\cdot\mathbf{u}_\lambda^{n+1}. \tag{3.31}$$

However, the remarks that were made in Section 3.2.1 regarding the replacement of the vorticity field by its FD approximation naturally also applies here, both for the vorticity and the dilatation fields. Equations (3.30) and (3.31) are then replaced by

$$\boldsymbol{\omega}_\lambda^{n+1} = \boldsymbol{\omega}^{n+1} + \nabla\times\left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1}\right), \tag{3.32}$$

$$\theta_\lambda^{n+1} = \theta^{n+1} + \nabla\cdot\left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1}\right). \tag{3.33}$$

Finally, a penalization of the temperature is performed to impose the isothermal boundary condition,

$$\frac{\partial T}{\partial t} = -\lambda_T\chi_b\left(T - T_b\right) \tag{3.34}$$

where λ_T is the penalization coefficient for the temperature equation. In practice, λ and λ_T are chosen to be equal. Equation (3.34) is also integrated implicitly and used to correct the internal energy (divided by the specific heat at

constant volume c_v) (Eq. (Alg. 1.6)).

Using the same notations as in Eq. (3.4) and Eq. (3.10), we here have a vector correction function $\delta\mathbf{F} = \boldsymbol{\mathcal{P}}(\mathbf{F}^{n+1}, \mathbf{u}^{n+1}, \mathbf{u}_b, T_b)$ evaluated on the grid that reads

$$\begin{aligned}\boldsymbol{\mathcal{P}} &= \begin{pmatrix} \mathcal{P}_\rho \\ \mathcal{P}_{\boldsymbol{\omega}} \\ \mathcal{P}_\theta \\ \mathcal{P}_{\rho T} \end{pmatrix} (\mathbf{F}^{n+1}, \mathbf{u}^{n+1}, \mathbf{u}_b, T_b) \\ &= \begin{pmatrix} -\Delta t \left(\left(\frac{1}{\phi} - 1 \right) \chi_b \right) \nabla \cdot \left((\rho\mathbf{u})_\lambda^{n+1} \right) \\ \nabla \times \left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1} \right) \\ \nabla \cdot \left(\mathbf{u}_\lambda^{n+1} - \mathbf{u}^{n+1} \right) \\ \rho_\lambda^{n+1} \left(\frac{T^{n+1} + \lambda_T \Delta t \ \chi_b \ T_b}{1 + \lambda_T \Delta t \ \chi_b} \right) - (\rho T)^{n+1} \end{pmatrix} \end{aligned} \tag{3.35}$$

together with

$$\mathcal{P}_{\rho\mathbf{u}} = \frac{(\rho\mathbf{u})^{n+1}}{(1 + \lambda\Delta t \chi_b)} - (\rho\mathbf{u})^{n+1} \tag{3.36}$$

3.3 Numerical tests

In this section, the VPM solver for compressible flows with a Brinkman penalization technique will be validated on benchmark test cases. Both the explicit and the implicit strategies for the penalization terms will be used. The first test case is the benchmark problem of an incompressible flow past a cylinder at low Reynolds number. The second test involves a compressible flow: the 2-D acoustic scattering of a cylinder by a single localized acoustic pulse. Analytical solutions exist for both tests; even though only for short times for the first one. The results will be compared to those analytical solutions and from results obtained in the literature.

3.3.1 Impulsively started and incompressible flow past a cylinder at low Reynolds number (Re = 550)

For this first test case, the compressible flow solver presented in Section 3.2.2 is here reduced to solve the penalized Navier-Stokes equations for incompressible flows; the temporal loop and the sequence of operations are left unchanged but the RHS terms are reduced to those of an incompressible flow.

This incompressible test case is performed to validate the algorithms but more importantly to reach full mastery on the penalization method with both time-stepping strategies. We consider the same configuration as in [59]: a cylinder of diameter D in a computational domain of size $[5D; 5D]$. The center of the cylinder is located at $[5D/4;\ 5D/2]$. The variables were made dimensionless using the diameter of the cylinder and the freestream velocity, U_∞. The Reynolds number is taken as $\mathrm{Re} = \frac{DU_\infty}{\nu} = 550$ while the drag coefficient is defined as

$$C_D = \frac{\mathbf{F} \cdot \mathbf{e}_x}{\frac{1}{2}\rho U_\infty^2 D}. \tag{3.37}$$

The drag coefficient will be compared to the analytical solution valid for short times and to reference results from the literature [87, 128, 59]. The short time analytical expression of the drag coefficient obtained in [10] reads

$$C_D = 4\left(\frac{\pi}{\mathrm{Re}\, t}\right)^{1/2} + 2\pi\left(9 - \frac{15}{\sqrt{\pi}}\right)\frac{1}{\mathrm{Re}} \tag{3.38}$$

and was seen as a good solution until $t \leq 0.25$ for a similar Reynolds number ($Re = 500$) by Bar-Lev and Yang [10].

The grid size h is such that $\frac{h}{5D} = \frac{1}{1024}$ as in [59]. The dimensionless viscous permeability $(\eta\rho)$ is equal to the dimensionless penalization factor $1/\lambda$ in this incompressible test case. It was chosen such that $(\eta\rho) = \frac{1}{\lambda} = 1.0 \cdot 10^{-4}$. A fixed time step was chosen such that $\lambda\Delta t = 2.5$ when using the explicit time stepping strategy (with the RK4C scheme). Using this penalization parameter, the time step for the explicit time stepping technique is here strongly constrained by the penalization. Indeed, it leads to an a priori diffusion number $r = \frac{\nu\Delta t}{h^2} = 1.91{\cdot}10^{-2}$ and to an a priori CFL number $\mathrm{CFL} = \frac{U_\infty \Delta t}{h} = 5.12{\cdot}10^{-2}$. The time step used in this validation test case is one to two orders of magnitude lower than the time step used in the reference results from the literature used hereunder [87, 128, 59]. When using the implicit strategy, the dimensionless penalization parameter is chosen such that $\lambda\Delta t = 1.0 \cdot 10^5$ with $\Delta t = 1 \cdot 10^{-3}$. The particles are remeshed using the M_6^* scheme at each time step but are not filtered. The Poisson solver with unbounded boundary conditions introduced in Chapter 2 is used. As in [102, 105, 59], the force exerted on the cylinder is computed using the approach from Noca et al. [110, 111]. This force could also be measured by integrating the penalization term in the momentum equation. This second technique has the advantage to also be valid in a compressible regime contrarily to the approach derived by Noca et al. It leads, however, to noisy results when using large time steps as observed

in [105].

Figure 3.1 compares the results of the presented methodology against the analytical solution of Eq. (3.38) and with reference results from the literature. The results obtained with this non-iterated method are in satisfactory agreement with the analytical solution at the onset of the flow. However, the time step used with the explicit time stepping strategy is here tiny; the method should thus not be deemed better than the results from the literature. For $t \geq 0.2$, the results depart from the analytical solution valid for short times and are found to be in excellent agreement with the numerical results from the literature as shown on the right-hand side of Figure 3.1 and in Figure 3.2.

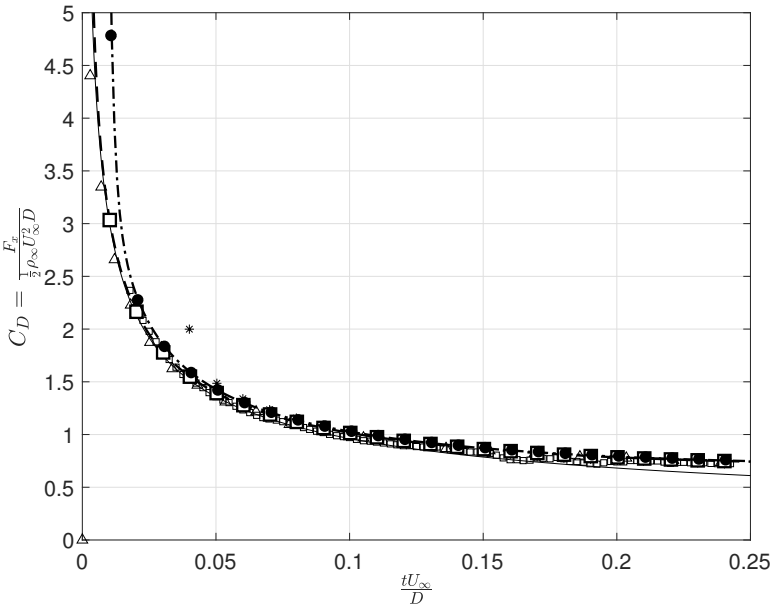


Figure 3.1: Incompressible flow past a cylinder at $\mathrm{Re} = 550$: Analytical solution: thin solid line, Koumoutsakos and Leonard [87] (star), Ploumhans and Winckelmans [128] (square), Gillis et al. [59] (triangle), present method with explicit penalization (dashed line with squares), present method with implicit penalization (dashed-dotted line with black circles).

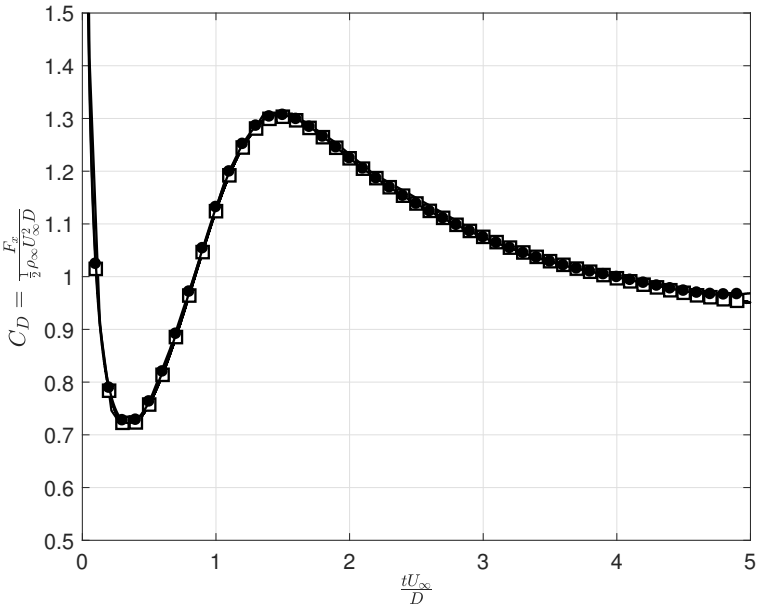


Figure 3.2: Incompressible flow past a cylinder at $\mathrm{Re} = 550$: Ploumhans and Winckelmans [128] (solid line), present method with explicit penalization (dashed line with squares), present method with implicit penalization (dashed-dotted line with black circles).

3.3.2 2-D cylindrical acoustic benchmark problem

The second test case is the classical 2-D acoustic scattering of a cylinder generated by a single localized acoustic pulse. This benchmark problem was introduced in the Second Computational Aeroacoustics workshop on Benchmark Problems (Category 1, problem 2) [149]. An analytical solution exists when solving the Euler equations. It was also one of the verification test cases considered by Liu and Vasilyev [99]. As in their paper, we here solve the full Navier-Stokes equations with the same acoustic Reynolds number $Re_a = \frac{\rho_\infty a_\infty D}{\mu} = 5.0 \cdot 10^5$, with D the diameter of the cylinder located at the origin of the computational domain. The same acoustic nondimensionalization as in Chapter 2 is here used with the diameter D as the reference length. The initial condition is similar to the acoustic test case presented in Section 2.3.1; the fluid is initially at rest, and the pressure and density fields

exhibit a Gaussian excess zone. It is here recalled for convenience:

$$\frac{u(0)}{a_\infty} = \frac{v(0)}{a_\infty} = 0, \tag{3.39}$$

$$\frac{p_a(0)}{\rho_\infty a_\infty^2} = \frac{\rho_a(0)}{\rho_\infty} = \varepsilon \exp\left(-\frac{r^2}{\sigma^2}\right) = \varepsilon \exp\left(-\ln 2\,\frac{r^2}{b^2}\right) \tag{3.40}$$

with $p_a = p - p_\infty$ the pressure disturbance, $\rho_a = \rho - \rho_\infty$ the density disturbance, $r = \left((x - x_c)^2 + (y - y_c)^2\right)^{1/2}$ the radial distance in polar coordinates, and b the half-width of the Gaussian. The pulse is located at the center of computational domain ($[x_c, y_c] = [4, 0]$) and is such that $b/D = 0.2$ and $\varepsilon = 1 \cdot 10^{-3}$. The computational domain as well as the observations points are depicted in Figure 3.3. The same resolution as in the Section 2.3.1 is used: 40 particles lay across the half-width diameter 2b. This will allow to isolate the error due to the boundary treatment from the truncation error in the domain.

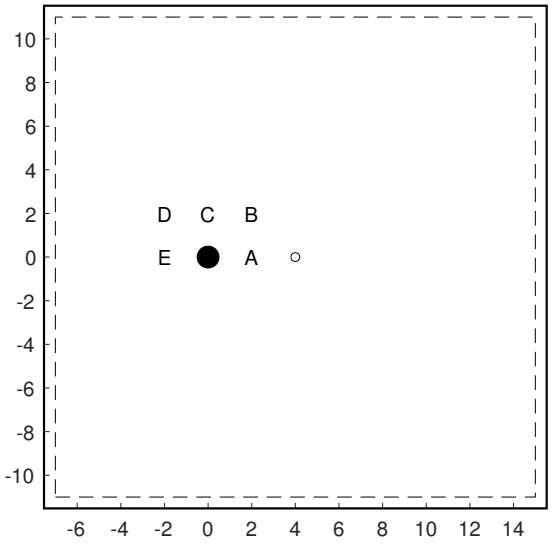


Figure 3.3: Acoustic scattering of a cylinder: sketch of the computational domain and observation points lettered from A to E. The cylinder and the initial location of the acoustic pulse are depicted as the large filled and small open black circles, respectively. The dashed line marks the beginning of the zone where the radiation condition is applied.

We here consider a squared computational domain $\Omega = [-7.52, 15.52] \times [-7.52, 15.52]$. This domain has a noticeably greater extent than the rectangular domain considered in [99], which was defined as $\Omega = [-5, 9] \times [-5, 5]$. This is a direct consequence of the use of the Helmholtz decomposition to calculate the velocity field and from the radiating nature of the dilatation field.

As already discussed in Section 2.2.3, regardless of the quality of the radiation boundary condition, the size of the domain itself has to be chosen such that the truncation of the dilatation field has a negligible or reasonable effect on the velocity field. By choosing such an extent for the domain, the dilatation field will not be truncated in the course of this simulation. However, the domain where the NS equations are solved is still surrounded by a band of 10 mesh points used to impose the radiation condition. Its sole purpose is to evacuate possible spurious numerical oscillations. The purpose of the remaining points in the boundary zone is to have a total number of points leading to optimal conditions for the Poisson solver. The bottleneck of VPM methods is indeed the resolution of the Poisson equation(s). The documentation of the FFTW library used in this implementation states that the standard FFTW distribution reaches optimal efficiency when using arrays whose size can be factored into small primes (2, 3, 5 and 7). In this work, additional grid points are added when necessary so that the total number of points in each direction is such that $N = 2^a \times 3^b \times 5^c$.

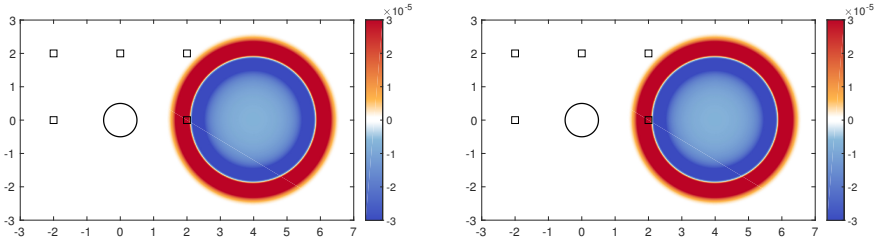
When using the explicit time-stepping technique, the penalization parameters were chosen to be identical to those used by Liu and Vasilyev in [99] for comparison purposes; the porosity, the dimensionless viscous and thermal permeabilities are such that $\phi = 2 \cdot 10^{-2}$, $\eta = \alpha\phi = 1 \cdot 10^{-3}$ and $\eta_T = \alpha_T\phi = 1 \cdot 10^{-3}$. As explained in [99], the thickness of the internal transition region can be estimated as $\mathcal{O}((\alpha\eta_T)^{1/2})) = 7.1 \cdot 10^{-3}$. The internal transition region is thus not resolved in this test case as $h/D = 1.0 \cdot 10^{-2}$. The corresponding error estimate is thus $\mathcal{O}\left((\frac{\eta}{\eta_T})^{1/4}\phi^{3/4}\right) = \mathcal{O}(5.3 \cdot 10^{-2})$. When using the implicit penalization, the porosity ϕ was left unchanged and the penalization parameters λ and λ_T were chosen such that $\frac{\lambda D}{a_\infty} = \frac{\lambda_T D}{a_\infty} = 1 \cdot 10^{-8}$. A fixed time step $\frac{\Delta t a_\infty}{D} = 2.5 \cdot 10^{-3}$ was used in both cases. The particles were remeshed and filtered at each time step ($n_r = n_f = 1$) although no comprehensive study on the impact of these parameters was performed.

Figure 3.4 shows close-up snapshots of the pressure field for both time stepping strategies. The acoustic pulse first propagates in the domain (Fig. 3.4 (a)) and hits the cylinder at $ta_\infty/D \simeq 3.0$. The first wave is then split by the cylinder while a secondary wave is reflected by the cylinder (Fig. 3.4 (b)). The secondary wave propagates towards the boundaries while wrapping around the cylinder. (Fig. 3.4 (c)). A tertiary wave is formed when the two parts of the separated first wave merge on the left of the cylinder (Fig. 3.4 (c)). This third wave then propagates towards the boundaries (Fig. 3.4 (d)). Results of

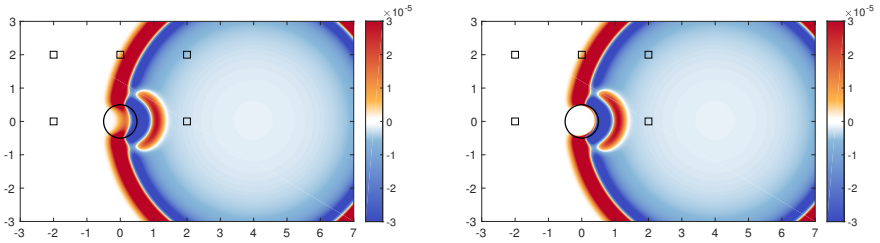
similar quality were obtained with both integration techniques. Note, however, that the pressure field obtained using the explicit integration exhibits a higher value inside the obstacle at $\frac{t\,a_\infty}{D} = 4$. This can be explained by the lower penalization parameter used to enforce the temperature inside the obstacle when using the explicit integration.

Figures 3.5 to 3.7 compare the pressure perturbations measured at five different locations (see Fig. 3.3) with the analytical solution. Insets of the pressure field at relevant times are also provided. The first wave is unsurprisingly accurately captured at all locations; at this point, it is indeed essentially the same test case as in Section 2.3.1. More interestingly, the accuracy with which the second and third waves are captured rests on the quality of the penalization method. The results are found to be in excellent quantitative agreement with the analytical solution.

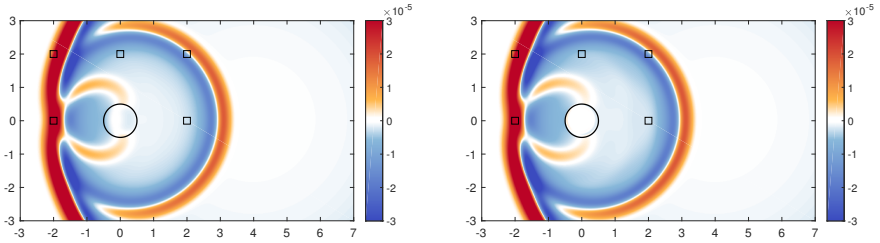
As explained and numerically illustrated in [99], and recalled here above, the quality of the solution strongly depends on the porosity, ϕ. It was also confirmed by Piquet et al. [127]; a parameter study was performed on the problem of the shock wave reflection from a solid wall, even going so far as to deactivate the penalization of the continuity equation (by taking $\phi = 1$). We here perform a similar test; a unity porosity is chosen. The dimensionless penalization parameters were chosen such that $\frac{\lambda D}{a_\infty} = 1.0 \cdot 10^{10}$ and $\frac{\lambda_T D}{a_\infty} = 8.0 \cdot 10^4$ in order to leave the value of the error estimate of the previous simulation unchanged. The other parameters of the simulation were not modified and the implicit strategy was used. As expected, the larger wave transmission inside the obstacle translates into higher amplitude and transmission error. This is best observed in the time history of the pressure perturbation at the location of the sensors C and D (see Fig. 3.8).



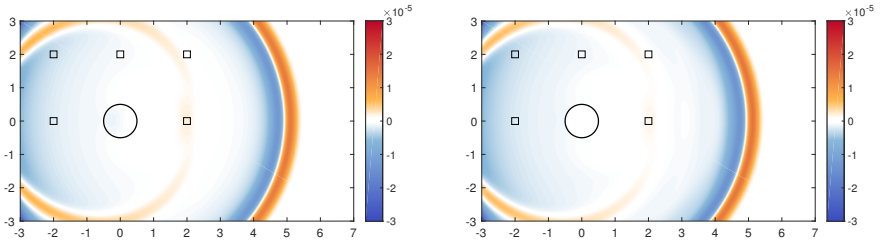
(a) $\frac{t\,a_\infty}{D} = 2$



(b) $\frac{t\,a_\infty}{D} = 4$

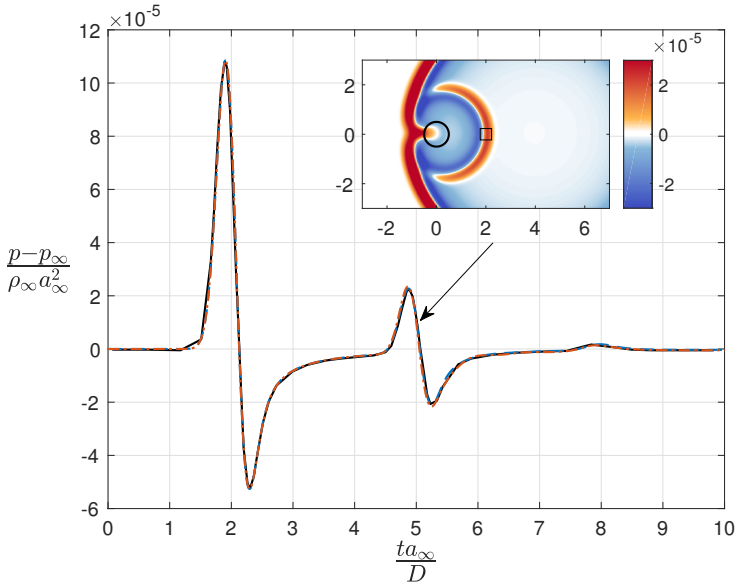


(c) $\frac{t\,a_\infty}{D} = 6$

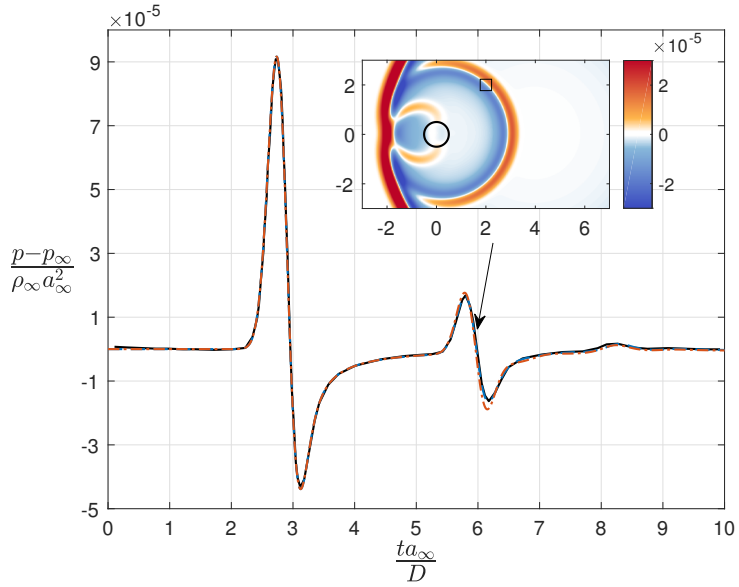


(d) $\frac{t\,a_\infty}{D} = 8$

Figure 3.4: Acoustic scattering of a cylinder: Pressure perturbations in a close-up view around the cylinder (bold solid line) at four times using the explicit (left) and the implicit technique (right). The five observation points are indicated with squares. Note that the interior of the cylinder is not hidden in order to see the penalized pressure field inside the cylinder.

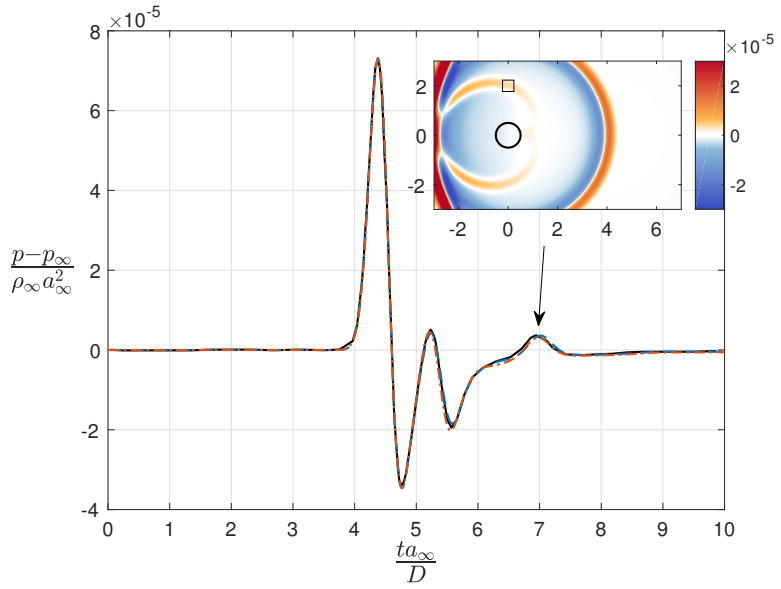


(a) Sensor A at (2,0), inset at time $\frac{t\,a_\infty}{D} = 5$

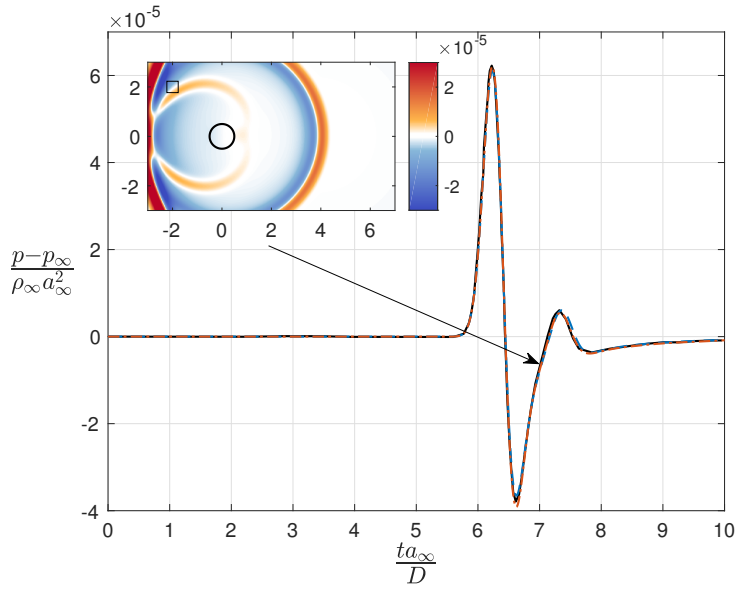


(b) Sensor B at (2,2), inset at time $\frac{t\,a_\infty}{D} = 6$

Figure 3.5: Acoustic scattering of a cylinder: time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five locations indicated with square in the inset. Analytical solution (solid black line), explicit penalization (blue dashed line), implicit penalization (red dashed-dotted line). The solutions shown in the insets were obtained using the explicit strategy.

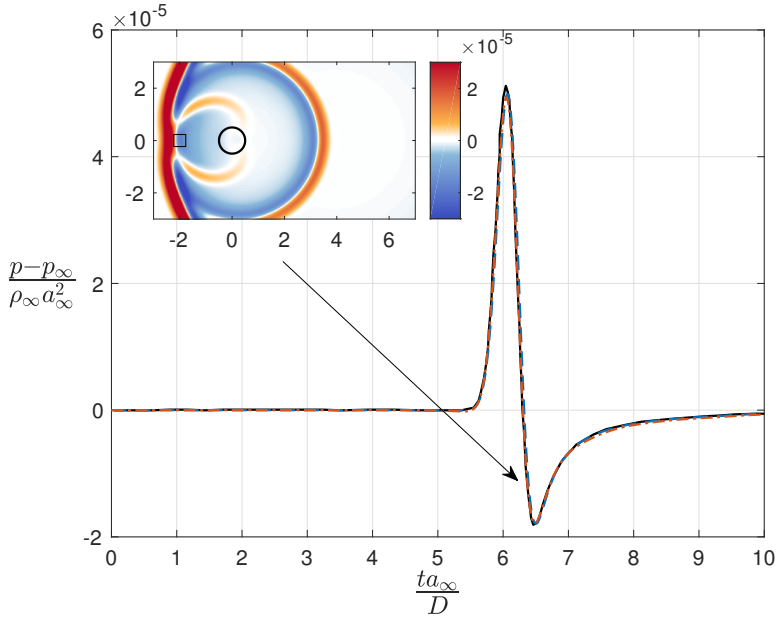


(a) Sensor C at (0,2), inset at time $\frac{t\,a_\infty}{D} = 7$



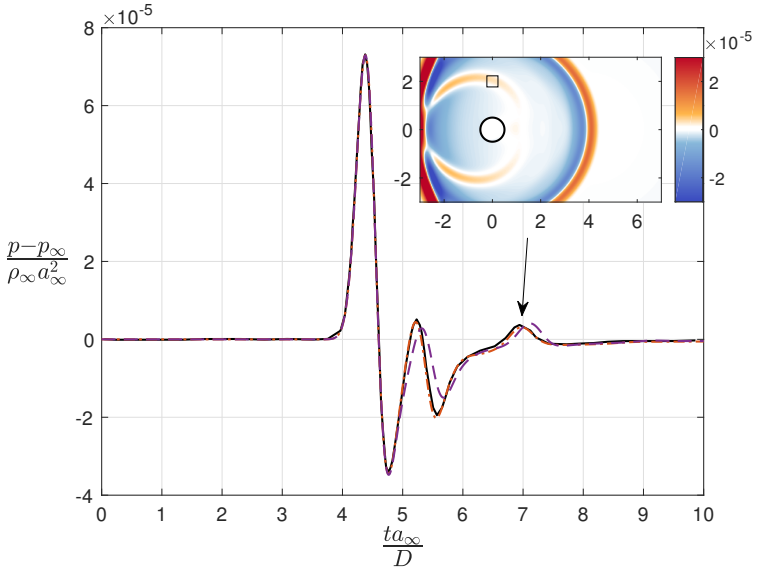
(b) Sensor D at (-2,2), inset at time $\frac{t\,a_\infty}{D} = 7$

Figure 3.6: Acoustic scattering of a cylinder (continued): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five locations indicated with square in the inset. Analytical solution (solid black line), explicit penalization (blue dashed line), implicit penalization (red dashed-dotted line). The solutions shown in the insets were obtained using the explicit strategy.

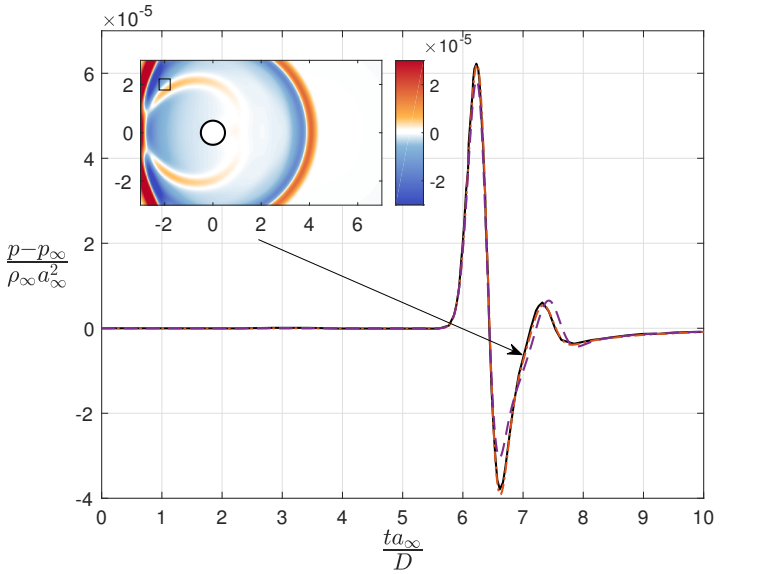


(a) Sensor E at (-2,0), inset at time $\frac{t\,a_\infty}{D} = 6.4$

Figure 3.7: Acoustic scattering of a cylinder (continued): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five locations indicated with square in the inset. Analytical solution (solid black line), explicit penalization (blue dashed line), implicit penalization (red dashed-dotted line). The solutions shown in the insets were obtained using the explicit strategy.



(a) Sensor C at (0,2), inset at time $\frac{t\,a_\infty}{D} = 7$



(b) Sensor D at (-2,2), inset at time $\frac{t\,a_\infty}{D} = 7$

Figure 3.8: Acoustic scattering of a cylinder: time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at the location of the sensors C and D. Analytical solution (solid black line), implicit penalization with $\phi = 2.0 \cdot 10^{-2}$, $\lambda D/a_\infty = \lambda_T D/a_\infty = 1.0 \cdot 10^8$ (dashed-dotted red line), implicit penalization with $\phi = 1.0$, $\lambda D/a_\infty = 1.0 \cdot 10^{10}$ and $\lambda_T D/a_\infty = 8.0 \cdot 10^4$ (dashed purple line). The solutions shown in the insets were obtained from the simulation using $\phi = 2.0 \cdot 10^{-2}$.

3.4 Conclusions

In this chapter, the Brinkman penalization method for compressible flows from Liu and Vasilyev [99] was adapted to be used in the VPM for compressible flows presented in Chapter 2. It imposes Dirichlet boundary conditions on the velocity and the temperature fields at the surface of one or multiple obstacles. This methodology allows to take obstacles into account while retaining the advantages associated with the use of an underlying Cartesian grid: efficient grid-based Poisson solvers, data alignment and vectorizing loops, and efficient evaluation of the spatial differential operators. The straightforward implementation of the penalization method clearly constitutes an advantage. Although it is not as accurate as other existing methodologies (like an Immersed Interface method for example), especially at high Reynolds number, it allows to capture the overall effect of obstacles.

Anyhow, the VPM is not intended to accurately capture the near-body complex dynamics in the coupling methodology presented in Chapter 4. Indeed, as already briefly explained in Chapter 1, the near-body region will be resolved using an Eulerian body-fitted solver. The solution computed by the latter will then be used to correct the solution of the VPM solver in that region, thus compensating the lack of accuracy of the VPM solver and of the Brinkman penalization method. The main purpose of the addition of the Brinkman penalization method to the VPM solver is thus to provide proper boundary conditions to the Eulerian near-body solver.

The method was implemented using two different strategies; an explicit and an implicit integration of the penalization terms. The first methodology relies on the simple addition of the penalization terms to the RHS terms of the Navier-Stokes equations. The same temporal scheme is then applied to all terms. The second methodology, on the contrary, relies on a time-splitting strategy. At each time step, the Navier-Stokes equations are solved omitting the presence of the obstacles. The penalization terms are then integrated using an implicit Euler scheme. The implicit strategy was reported in the literature [132] to provoke a substantial diffusion error due to the replacement of the vorticity field by the finite difference evaluation of the implicitly penalized velocity field. However, this shortcoming may be significantly attenuated by recasting the penalization terms. At a given time step, the vorticity field is then corrected by the curl of the difference between the penalized and the unpenalized velocity fields. The resulting correction is thus only applied in a narrow region covering the obstacles (and a few exterior points whose num-

ber depends on the FD scheme used to evaluate the curl). The same remark holds for the dilatation field when considering a compressible flow solver.

Both strategies were tested on benchmark problems for which analytical and / or reference results exists in the literature. The solver was first tested on the test case of an incompressible flow past a cylinder at $Re = 550$. The acoustic scattering of sound by a cylinder was then studied using the full compressible solver. In both cases, the results were found to be in excellent quantitative agreement with the available analytical solutions and with reference results from the literature.

CHAPTER 4

COUPLING A VPM SOLVER AND AN EULERIAN DG SOLVER FOR COMPRESSIBLE FLOWS

In this chapter, the methodology allowing to use the VPM *method for compressible flows of Chapter 2 in conjunction with a Discontinuous Galerkin solver will be presented. To that end, the VPM solver will also be fitted with the penalization of Chapter 3 when solid obstacles are present.*

4.1 Introduction

The idea of coupling a VP or VPM method with an Eulerian solver for the near-body region is not new. Several implementations already exist for incompressible flows both in 2-D and in 3-D, either using a vorticity-velocity formulation, a vorticity-stream function formulation or a velocity-pressure formulation in the Eulerian solver; a non-exhaustive list includes [37, 65, 113, 44, 100, 102], also see the reference book of Cottet and Koumoutsakos [40] and references therein. While the equations to be solved are different in nature, the developments that will be introduced in this chapter share some common grounds with the methodologies developed for incompressible flows.

The completely overlapping domain decomposition technique illustrated in Fig. 1.5 was first introduced by Daeninck [44] and latter applied in different contributions [147, 100, 102]. The use of the Lagrangian solver in the whole computational domain differs from the partially overlapping domain decomposition technique used in previous works [37, 40, 113]. This obviates the need for an iterative procedure, namely a Schwarz alternating method, to compute the boundary conditions of the Eulerian solver.

Using the completely overlapping domain decomposition technique of Daeninck [44] implies a correction procedure of the Lagrangian solution in the Eulerian subdomain; the Lagrangian solver is purposely under-resolved in that region. The whole coupling strategy relies on the consistent representation of the solution in both solvers in the Eulerian subdomain. To do so, the correction of the Lagrangian solution should be conservative. This is the major challenge of the coupling methodology. To the author's knowledge, no conservative correction procedure has been reported in the existing coupling methodologies. Several contributions correct the particle strengths using a simple interpolation from the Eulerian mesh [44, 147]. In [102], the particle strengths, $\alpha_p = \boldsymbol{\omega}_p V_p$, are corrected using a linear reconstruction of the vorticity field of the Finite Volume (FV) Eulerian solver and a Gauss quadrature over the particles. As stated in [102], this correction procedure is not fully conservative but provides smooth fields across the boundary of the Eulerian solver. An interesting alternative was proposed by Marichal [102]: a finite volume cell could be assimilated to a particle, and its integrals could be redistributed on the nearest VPM particle. As stated in [102], this alternative would be conservative but would lead to a less smooth Lagrangian grid representation. Most of these contributions [147, 102] report that the correction should be performed in a subset of the Eulerian subdomain excluding a small region close to the outer boundaries of the Eulerian solver. This small buffer region is generally a few particles wide and aims at avoiding inaccuracies near the outer boundary. In [147], a small region close the solid wall is also excluded to avoid spurious oscillations near the wall. It is, however, the author's belief that this region should not be excluded; it is precisely where the under-resolved Lagrangian solution should be corrected. In [44, 147, 102], the Lagrangian solver is complemented with a panel solver to compensate its under-resolution at the wall and to eventually provide proper boundary conditions to the Eulerian solver.

Some efforts have already been made to couple a near-body Eulerian solver for compressible flows with a Lagrangian solver either for incompressible flows [147] or for compressible flows [114, 118, 116]. They rely on a completely overlapping domain decomposition technique. In [147], Stock et al. developed a 3-D coupling methodology combining an Eulerian solver for compressible flows and a VP method for inviscid incompressible flows without remeshing. The Eulerian solver is a fully compressible flow solver using a structured overset grid systems for resolution adaptation. The VP solver uses a Fast-Multipole tree code solver to compute the velocity field and the anisotropic subgrid-scale model from Cottet et al. [38, 39]. A panel solver was also used to account for solid obstacles. As already mentioned above, the correction procedure was performed using a simple interpolation from the Eulerian mesh in a subset excluding a band close to the outer boundary of the Eulerian solver but also excluding a region close to the wall, where the vorticity field is important. Besides lacking good conservation property, the correction also misses a significant part of the available information in the Eulerian solver. Even using a panel method, it is doubtful that proper boundary conditions for the Eulerian solver will be computed. In [147], the results were shown to be in good qualitative agreement on different 3-D cases including a 4-bladed advancing rotor, although no quantitive assessment was performed. The use of an incompressible flow solver in the region close to the rotor blades is also questionable.

Following the work of Oxley [114], Papadakis and Voutsinas [118] developed a MPI-enabled 2-D hybrid compressible flow solver. It was later extended to perform 3-D aeroelastic simulations of the inviscid compressible flow past a simplified helicopter rotor by coupling the hybrid solver to a multibody structural solver [116]. This hybrid solver combines a low-order VPM solver for compressible flows with a near-body Finite Volume (FV) code solving the compressible Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations. The low-order VPM methodology was already presented in Chapter 2 and raises some concerns regarding the conservation properties of the VPM solver itself; a renormalization of the physical quantities, which are governed by conservation equations, is indeed performed. Moreover, in the 3-D version [116], there is no mention of the enforcement of the zero-divergence constraint of the vorticity field, although its necessity is acknowledged in [118]. In both [118] and [116], the coupling methodology is also rather poorly documented. The correction procedure is performed by evaluating $(\rho, \boldsymbol{\omega}, \theta, p)$ at the grid cell centers, using a Green-Gauss evaluation to compute the dilatation and the vorticity at these locations. These quantities are then interpolated on

an intermediate set of particles defined using iso-parametric finite element approximations; 4 new particles are defined per Eulerian cell in 2-D and 8 in 3-D. This information is then projected on the Lagrangian grid and transformed to remeshed particles upon convergence of an iterative procedure between the Eulerian and the Lagrangian solvers (yet there is no clear information regarding what is iterated). Although not acknowledged in [118, 116], this correction procedure is not conservative. Without necessarily being detrimental, solid boundaries are not taken into account in the Lagrangian solver. In [118], verification test cases were performed on compressible flows past airfoils at high Reynolds number (up to $12 \cdot 10^6$) and low Mach number (up to 0.26). The results seem however questionable; the physics of the flow appears to be in contradiction with the claimed Reynolds numbers. The wakes of these airfoils are, in the best-case scenario, highly under-resolved.

The coupling methodology presented in this Chapter will rely on existing techniques for incompressible flows when possible. The completely overlapping domain decomposition technique of Daeninck [44] and the idea of a conservative quadrature on the Eulerian elements suggested by Marichal [102] will be pursued in this Chapter: it will be exploited for the correction of the thermodynamic quantities and will indeed require a non-conservative renormalization of the particles. The circulation and the dilatation of the particles will be computed using contour integrals on the Lagrangian grid cells. The intended goal is to rely on a high order Eulerian solver. As stated in [71], it is now broadly accepted that high precision simulations may be achieved using lower computational resources when using a high order solver rather than with a lower order solver. The 2^{nd} order FV solver used by Papadakis and Voutsinas [118, 116] belongs to the second class. The scale-resolving capability of the DG solver is also an appealing capability for the applications introduced in Chapter 1. The URANS Eulerian solver of Stock et al. [147] does not exhibit such capability. In that regard, without being a detrimental aspect, their method is doubly hybrid as two different turbulence modeling are used in the coupled solver.

The chapter is organized as follows. The first section is dedicated to a succinct presentation of the Discontinuous Galerkin solver. A closer attention will be devoted to the elements of the solver involved in the coupling methodology. The following sections describe the key elements of the coupling approach: the imposition of the boundary conditions on the outer boundary of

the Eulerian solver on the one hand, and the correction of the solution of the Lagrangian solver in the Eulerian subdomain on the other hand.

4.2 Discontinuous Galerkin solver

The near-body Eulerian solver is developed at Cenaero, a research center based in Gosselies, Belgium. It implements a Discontinuous Galerkin Method (DGM), whose detailed description can be found in the PhD dissertation of Koen Hillewaert [71]. Other pertinent and interesting information about the DG solver used in Argo are found in [23, 139, 52].

The Discontinuous Galerkin Method is a Galerkin finite element method[1]. Non-overlapping elements e are the building blocks of the mesh $\mathcal{E}$ that approximates and discretizes the computational domain $\Omega \simeq \mathcal{E} = \cup e$. Contrarily to the cartesian grid used in the VPM solver which is designed to be used in conjunction with finite differences, the elements used in the DGM solver could in theory be of any type. In practice, however, the elements in Argo are triangles and quadrilaterals in 2-D and tetrahedra, prisms, hexahedra and pyramids in 3-D. Moreover, conformal elements are used; i.e. the boundary face of an element may only connect to a single other element of the mesh [71]. An example of non-conformal mesh is represented in Fig. 4.1. As in

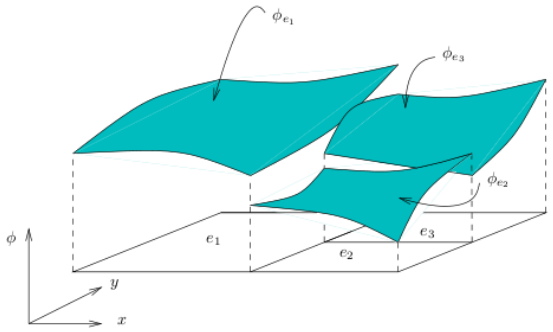


Figure 4.1: Example of a discontinuous interpolation on a non-conformal mesh using quadrilateral elements. Reproduced from [71], which was itself inspired from [134].

finite element methods, these elements are the support for the definition of the functions belonging to the *trial space*, $\mathcal{V}$; a finite-dimensional function space in which the solution state vector $\tilde{\mathbf{U}}$ is sought for. Each component of the exact

[1] It thus chooses that the spaces spanned by the basis functions ψ_i and the test functions ϕ_i are the same [70]

solution vector $\tilde{\mathbf{U}} = [\tilde{u}_0, \tilde{u}_1, ..., \tilde{u}_{N_v}]$ is thus approximated by a combination of linearly independent vector functions ϕ_{im}, called *shape functions*, forming a basis of the trial space, $\mathcal{V}$. On each element, the approximation of this function will thus be written as

$$u_m = \sum_{i=0}^{N} u_{im}\phi_{im}, \qquad m = 1, 2, ..., N_v \tag{4.1}$$

with N_v the number of variables and $N + 1$ the number of degrees of freedom per element. The expansion coefficients u_{im} are the unknowns of the problem. In Argo, the trial space $\mathcal{V}$ is composed of functions v that are Lagrange polynomials of interpolation order p defined on each element of the mesh (i.e. $N = p$ in Eq. (4.1) for one-dimensional problems). Contrarily to classical Finite Element Methods (FEM), a DG method authorizes discontinuities across the elements faces f by duplicating the variables at the element boundaries. The polynomials are defined in parametric coordinates ξ in a parent or reference element for each element of the same type and interpolation order (the combination of these two being designated as a category). A mapping then allows to convert parametric coordinates (ξ_1, ξ_2, ξ_3) into the physical coordinates (x_1, x_2, x_3), or conversely, as illustrated in Fig. 4.2 for a curved triangle.

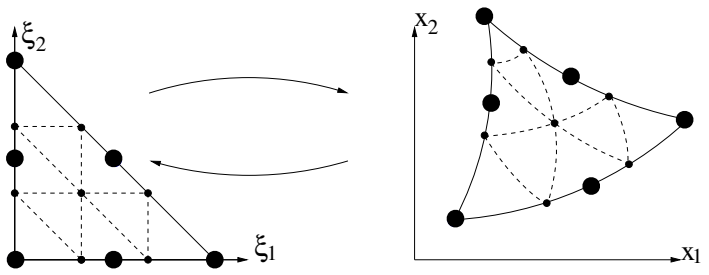


Figure 4.2: Third order mapping (small control points) of a triangle with second order interpolation (large control points). Reproduced from [71].

The Navier-Stokes equations (1.1) can be rewritten in the following general form

$$\mathcal{L}_m(\tilde{\mathbf{U}}) = 0 \quad \text{with } m = 1, 2, ..., N_v \tag{4.2}$$

When evaluated using the approximate solution $\mathbf{U}$, this form coincides with the residual $\mathcal{R}_m(\mathbf{U}) = \mathcal{L}_m(\mathbf{U})$. As for any Galerkin formulation, the expansion weights u_{im} in Eq. (4.1) are found by requiring that the residual is orthogonal to all functions in the trial space $\mathcal{V}$. This statement is referred to as the

Galerkin variational formulation

$$\int_\Omega v_m \mathcal{L}_m(\mathbf{U}) = 0, \quad \forall \mathbf{v} \in \mathcal{V} \tag{4.3}$$

However, as the functions ϕ_i form a basis of the trial space, it is sufficient to require the residual to vanish for all $\phi_i \in \mathcal{V}$ on each element e of the mesh. The local definition of the functions in the trial space $\mathcal{V}$ as well as the duplicate unknowns at the elements boundary allows to have a local statement in order to compute the expansion weights. The locality of the DG method constitutes one of its appealing aspects.

At this point, the linear diffusive flux is defined using the Jacobian $\mathbf{D}$ of the diffusive fluxes with respect to the solution gradient

$$\boldsymbol{\mathcal{D}}_m^k \simeq D_{mn}^{kl} \frac{\partial u_n}{\partial x^l} \tag{4.4}$$

using the fourth-order tensor D_{mn}^{kl} for the variable m in the direction k and Einstein summation notation over l and n. Applying the Galerkin variational formulation Eq. (4.3) to the Navier-Stokes equations Eq. (1.1), and using integration by parts to separate volume and interface contributions, the variational formulation can be rewritten as

$$\int_\Omega v_m \mathcal{L}_m(\mathbf{U}) = 0, \quad \forall \mathbf{v} \in \mathcal{V} \tag{4.5}$$

$$\begin{aligned}
&= \underbrace{\sum_e \int_e v_m \frac{\partial u_m}{\partial t} dV}_{T_v} \\
&\quad \underbrace{- \sum_e \int_e \frac{\partial v_m}{\partial x^k} \boldsymbol{\mathcal{C}}_m^k(\mathbf{U}) dV}_{C_v} + \underbrace{\sum_f \int_f [[v_m]]^k n^k \boldsymbol{\mathcal{H}}_m(\mathbf{U}^+, \mathbf{U}^-, \mathbf{n}) dS}_{C_i} \\
&\quad \underbrace{- \sum_e \int_e \frac{\partial v_m}{\partial x^k} \boldsymbol{\mathcal{D}}_m^k(\mathbf{U}) dV}_{D_v} + \underbrace{\sum_f \int_f \left\langle D_{mn}^{kl} \frac{\partial u_n}{\partial x^l} \right\rangle [[v_m]]^k dS}_{D_i} \\
&\quad \underbrace{+ \sum_f \int_f \left\langle D_{nm}^{kl} \frac{\partial v_m}{\partial x^l} \right\rangle [[u_n]]^k dS}_{D_t} + \underbrace{\sigma \sum_f \int_f [[v_m]]^k [[u_m]]^k dS}_{D_p} \\
&\quad \underbrace{+ \sum_e \int_e v_m \boldsymbol{\mathcal{S}}_m(\mathbf{U}, \nabla \mathbf{U}) dV}_{S_v}
\end{aligned} \tag{4.6}$$

where the Symmetric Interior Penalty (SIP) method has been used to treat the diffusive terms. The terms D_t, D_i and D_p arise from the SIP method. The

volume terms in Eq. (4.6) (T_v, C_v, D_v, S_v) are similar to the Galerkin variational formulation in a classical Finite Element Method. The first term T_v represents the temporal evolution in the volume while the other three, C_v, D_v and S_v, represent the volume contributions of the convective, diffusive fluxes and source terms, respectively. The other terms couple the current element to its neighbors. The interface terms C_i and D_i are specific to the Discontinuous Galerkin method and result from the possibility for the approximate solution to be discontinuous at the elements boundaries (in a classical FEM method, these terms vanish). The following average and jump operators are defined in order to write the interface terms concisely,

$$[[a]] = a^+\mathbf{n}^+ + a^-\mathbf{n}^-, \tag{4.7}$$

$$[[\mathbf{a}]] = \mathbf{a}^+ \cdot \mathbf{n}^+ + \mathbf{a}^- \cdot \mathbf{n}^-, \tag{4.8}$$

$$\langle a \rangle = \frac{(a^+ + a^-)}{2}. \tag{4.9}$$

As detailed in [71] and shown at Fig. 4.3, the $+$ and $-$ signs are used to distinguish the values of the discontinuous quantities on either side of an interface f and depend on the element from which the interface is approached; a $+$ sign is used to refer to the element e sharing the same exterior normal as the oriented face f while a $-$ sign is used otherwise,

$$\mathbf{n}_e^f = \begin{cases} \mathbf{n}^+ & \text{if} \quad \mathbf{n}_e^f = \mathbf{n}_f, \\ \mathbf{n}^- & \text{if} \quad \mathbf{n}_e^f = -\mathbf{n}_f \end{cases} \tag{4.10}$$

where $\mathbf{n}_e^f$ is the exterior normal of the element e on the face f and $\mathbf{n}_f$ is the normal of the oriented face f. A limit value a defined on the element of exterior normal $\mathbf{n}^+$ will then be referred to as the *internal state* or *inside state*, a^+, whereas the neighbor element that is contiguous through the face f and of exterior normal $\mathbf{n}^-$ carry the *external state* or *outside state*, a^-. We note that a more explicit but less concise notation would, however, be

$$[[a\mathbf{n}]] = a^+\mathbf{n}^+ + a^-\mathbf{n}^-, \tag{4.11}$$

$$[[\mathbf{a} \cdot \mathbf{n}]] = \mathbf{a}^+ \cdot \mathbf{n}^+ + \mathbf{a}^- \cdot \mathbf{n}^-, \tag{4.12}$$

$$\langle a \rangle = \frac{(a^+ + a^-)}{2}. \tag{4.13}$$

We nevertheless elected to use the notations of Equations (4.7)-(4.9) to remain consistent with the work of [71, 23, 139, 52].

As explained in [71], a DG method can be interpreted as solving small FE problems on each element using Dirichlet-type boundary conditions to connect an element to its neighbors. For the convective interface term C_i, as in a

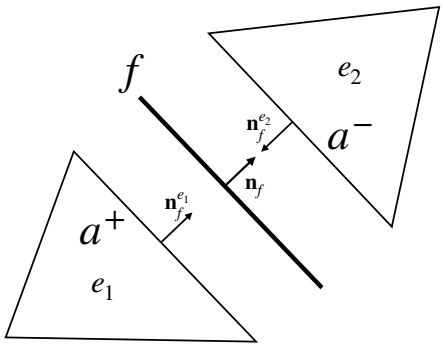


Figure 4.3: Sign convention: split-view of two adjacent elements (solid line) and their interface (bold solid line). As the lower element shares the same unit normal vector as the oriented interface f, the value of the discontinuous quantity a on its side of the interface is denoted as a^+. Inspired from [71].

Finite Volume Method (FVM), these internal boundary conditions are treated using a Riemann solver $\mathcal{H}$ (either exact or approximate) as the interface numerical flux, thus implementing characteristic boundary conditions. This ensures that the correct number of boundary conditions are employed for the convective terms. Several Riemann solvers are available in the module for compressible flows in Argo. Unless otherwise stated, the Roe approximate upwind Riemann solver will be used in this work. For the diffusive terms, the Symmetric Interior Penalty (SIP) method is used. The IP method, one variant of which being the SIP method, was chosen for its compacity in Argo as it indeed only uses the direct neighbors of a given element to evaluate the residual. The IP method generalizes a boundary penalty method to weakly enforce Dirichlet boundary conditions at the interfaces. This is again in agreement with the interpretation of a DGM as elementwise FEM problems with Dirichlet boundary conditions to interconnect them. The Dirichlet boundary condition is weakly enforced in the IP problem; the jump across the interface is penalized through the penalization parameter σ in the variational interface term D_p while the term D_i needs to be added for consistency. The stability of the SIP method is controlled by the stabilization parameter σ for which sharp bounds can be found in [71] for hybrid meshes.

As explained in [139], Dirichlet, Neumann and Robin boundary conditions are weakly enforced in a similar fashion at physical boundaries. For Dirichlet boundary conditions, a fictitious external state is provided and used:

- as an input for the Riemann solver in the convective interface flux term C_i,

- to compute the jump of the solution vector in the symmetrization term D_t and in the penalty term D_p.

The diffusive flux in the diffusive interface term D_i for the corresponding variable is also set to zero. As an example, a freestream boundary condition is enforced by specifying the exterior state for the velocity $\mathbf{u}^-$, the pressure p^- and the temperature T^-. The mean diffusive flux in D_i is, furthermore, set to zero. In this work, the Eulerian solver will be used in combination with the VPM solver fitted with the penalization method presented in Chapter 3, which enforces an isothermal wall boundary condition and a no-slip condition. As explained in [139], the same Dirichlet boundary condition is enforced in Argo by requiring that $\mathbf{u}^- = \mathbf{u}_b = 0$, $T^- = T_b$ and by taking the other variables as $\mathbf{U}^+ = \mathbf{U}^-$.

Neumann boundary conditions for the diffusive terms are enforced by first imposing identical inner and outer states at the interface, which cancels the jumps in the interface terms D_t and D_p. The average diffusive flux in the diffusive interface flux term D_i is further imposed to comply with the Neumann boundary condition. Should we have the possibility to impose an adiabatic wall boundary condition in the VPM solver, the same boundary conditions would be enforced in Argo by imposing $\mathbf{u}^- = \mathbf{u}_b = 0$, by taking the other variables as $\mathbf{U}^+ = \mathbf{U}^-$, and by modifying the mean diffusive flux in the term D_i such that the adiabatic wall boundary condition

$$\boldsymbol{\mathcal{D}}_{\rho E} \cdot \mathbf{n} = \mathbf{q} \cdot \mathbf{n} = 0. \tag{4.14}$$

is respected.

The integrals involved in the weak formulation (Eq. (4.6)) are then easily integrated using a numerical quadrature in the parametric space

$$\int_e a dV = \int_e a(\xi)|\mathbf{J}_e| d\xi \simeq \sum_{q=1}^{N_\alpha} w_q a(\alpha_q)|\mathbf{J}_e|_{\alpha_q} \tag{4.15}$$

where $\xi = [\xi_1, \xi_2, \xi_3]$ are the parametric coordinates and α_q, w_q and $|\mathbf{J}_e|_{\alpha_q}$ are the coordinates, the weight and the value of the determinant of the mapping Jacobian at the q-th integration point, respectively. The integrals on the boundaries of the element use their face counterparts, β_q, v_q and $|\mathbf{J}_f|_{\beta_q}$.

Another important operation is the interpolation of the solution variables onto the quadrature points. This operation is referred to as the *collocation*. Fol-

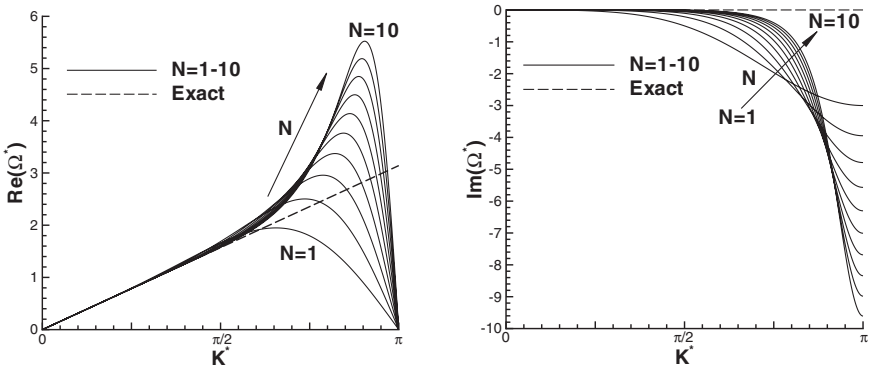
lowing [71], it can be expressed as the following matrix-matrix multiplication

$$\mathbf{U}^*_{qm} = (u_m)_{\alpha_q} = \sum_{i=1}^{N_\phi} (\phi_i)_{\alpha_q} \mathbf{U_{im}} = \mathfrak{C}^*_{qi} \mathbf{U}_{im} \tag{4.16}$$

where $\mathfrak{C}^*$ is called the collocation matrix for the elements of a given category. For further details regarding the data structures in Argo, refer to [71].

Out of the different explicit and implicit time-stepping techniques implemented in Argo, the temporal integration will here be carried out using the second-order implicit scheme Backward Differentiation Formula BDF2 in combination with a Newton-Raphson algorithm. The linear system resolutions that are involved in this iterative algorithm will here be solved using the matrix-free Generalized Minimum Residual method (GMRES). The details regarding the implementation of this matrix-free algorithm as well as the available preconditioners can be found in [71] and references therein.

The Discontinuous Galerkin method exhibits a high order accuracy on an unstructured mesh; the L_2 norm of its discretization error is $\mathcal{O}(h^{p+1})$, p being the interpolation order. Besides its capability to adapt the mesh to reflect the local requirements of the flow physics (*h-adaptivity*), it also enables the use of varying interpolation order p throughout the mesh (*p-adaptivity*). The combination of these two capabilities is known as *hp-adaptivity*. The hp-adaptivity allows to tailor the discretization in order to capture the flow physics both accurately and efficiently. It can also prove useful from a numerical and practical standpoint; for instance, remaining flow structures approaching the numerical boundaries can be damped by adequately decreasing the hp-refinement level in order to avoid spurious reflections. As already mentioned, the locality of the DG method also constitutes one of its appealing aspects; the method indeed lends itself to an efficient parallelization. The DG method is locally conservative and it, furthermore, exhibits attractive dispersion and dissipation properties, as shown in Figure 4.4. It has indeed a very low dispersion below $K^* = \pi/2$. The typical overshoot for large wavenumber values increases with the polynomial degree and is attributable to the one-sided approximations of the spatial derivatives. This is thus not specific to a DG discretization and was compared in [55] to the dispersion properties of the one-sided 6th-order finite difference scheme. When comparing Fig. 4.4(a) and Fig. 4.4(b) it can, however, be observed that the dispersed modes are also dissipated. As pointed out in [55], this confers some form of robustness to the method in case of under-resolution. A DG method is thus a prime candidate as near-body Eu-



(a) Dispersion properties (b) Dissipation properties

Figure 4.4: Dispersion (left) and dissipation (right) properties of the DG method for the 1-D periodic linear convection problem using polynomials of order $N = p$ ranging from 1 to 10 and a purely upwind flux. For comparison purpose, the eigenvalues and the wavenumbers, $K = kh$, have been normalized by the number of grid points of each grid cell, i.e. $\Omega^* = \Omega/(N+1)$ and $K^* = K/(N+1)$. Figures taken from Gassner and Kopriva [55].

lerian solver as the intended goal of the coupling methodology is convection-dominated problems; two low-dispersive methodologies will then indeed be used. Starting from the methodology of Marichal [103], a similar analysis is performed for the VPM solver in Appendix C.

4.3 Coupling boundary conditions for the Eulerian solver

The VPM solver is responsible for providing the boundary conditions that are necessary to evaluate the Galerkin variational formulation 4.6 on the elements on the outer boundary of the Eulerian domain. When studying flows past solid obstacles with the coupling methodology, which uses a completely overlapping decomposition technique, the VPM solver remains fitted with the penalization of Chapter 3, even though it is the role of the near-body Eulerian solver to properly account for these obstacles. Indeed, as illustrated in Fig. 1.6, the VPM solver might be used to perform several time steps between two corrections provided by the Eulerian solver. While being underresolved in the near-body region, the Lagrangian solution should thus remain of acceptable quality to provide proper boundary conditions to the Eulerian solver. Without any body treatment in the VPM solver, the application of finite difference operators close to the wall to evaluate the RHS of the Navier-Stokes equations would lead to unphysical gradients, which would result in the propagation of

spurious oscillations in the Lagrangian solution. The most significant errors would appear in the numerically evaluated gradients of the thermodynamic quantities and, in particular, the gradients of the density and the pressure; in this work, the temperature, the dilatation and the vorticity field are all subject to a Dirichlet boundary condition, which could be naively imposed inside the obstacle when evaluating derivatives at the wall. It is all the more important to avoid unphysical variations in the kinematics quantities, ω and θ, at the wall; they are indeed elliptically involved in the computation of the velocity field (see Eq. (2.8) and Eq. (2.9)) and thus also elliptically involved in the computation of the coupling boundary conditions provided to the Eulerian solver. As a consequence, unphysical variations in the near-wall region would also alter the quality of the coupling boundary conditions.

The imposition of the boundary conditions in the coupling methodology is to a large extent inspired by an existing capability of the DG solver; the non-matching boundary conditions, which allow the solver to deal with multiple computational domains. This authorizes the use of non-conformal and, more importantly, of sliding interfaces between the corresponding meshes. Typical usages of this capability include the simulation of turbomachinery flows when a rotor-stator interface has to be taken into account. Considering a segment of a single compressor or turbine stage, the physical domain that includes a rotating part (the rotor) and a stationary part (the stator) is decomposed into two computational domains. Proper boundary conditions emulating the sliding rotor-stator interface then need to be enforced. To do so, the quantities that are necessary to evaluate the interface terms in the Galerkin variational formulation, Eq. (4.6), are exchanged between the computational domains; this requires to provide the state vector $\mathbf{U}$ as well as its gradient $\nabla\mathbf{U}$ at the quadrature points of the interfaces coinciding with the non-matching boundaries. The faces on the sliding boundaries are thus treated in a fashion similar to the interfaces located in the volume of the computational domain; the exterior states are indeed computed and communicated rather than artificially imposed. The complexity comes evidently from the non-conformal and potentially time-varying configuration at the non-matching boundary, in particular for the parallelization. Indeed, a given element on one side of a sliding interface will need to exchange information with varying elements in the course of the emulated movement; those elements potentially being on different partitions of the mesh. To a certain extent, the Eulerian and the Lagrangian domains in the coupling methodology are in a situation similar to the abovementioned non-matching but motionless boundary condition. Aside from the

fact that different solvers are considered on both sides of the non-matching boundaries, another difference arises: as we here use a completely overlapping domain decomposition technique, the Eulerian solver is not required to provide (non-matching) boundary conditions to the Lagrangian solver. The imposition of boundary conditions is here thus unilateral: we note that a coupling methodology using partially overlapping domains and a Schwarz alternating method would have closer ties; the boundary conditions on both domains would then need to be enforced.

The details of the implementation will now be presented. For the remainder of this document, the quantities defined in the Lagrangian solver will be denoted with the subscript L while the quantities of the Eulerian solver will be denoted with a subscript E when necessary.

As explained here above, the VPM solver has to provide the boundary conditions that are required to evaluate all the interface terms of the Galerkin variational formulation, Eq. (4.6). The complete exterior solution vector, $\mathbf{U}^-$, is then provided by the Lagrangian solver as the input for the computation of the convective interface term C_i and for the computation of the jumps in the diffusive interface terms D_t and D_p,

$$\mathbf{U}^- = \mathbf{U}_L = [\rho,\ \rho\mathbf{u},\ \rho E]_L^T \tag{4.17}$$

with $\mathbf{U}_L$ is the solution vector as computed by the Lagrangian solver. The coupling methodology greatly benefits from the Riemann solver of the convective interface term, C_i. Indeed, it inherently implements characteristic boundary conditions as explained in Section 4.2. The proper number of boundary conditions is thus imposed without requiring any additional treatment in the coupling methodology.

In order to evaluate the average of the diffusive flux, $\left\langle D_{mn}^{kl} \frac{\partial u_n}{\partial x^l} \right\rangle$, in the interface term D_i, the exterior state of the gradient of the solution vector is also provided by the Lagrangian solver

$$(\nabla \mathbf{U})^- = (\nabla \mathbf{U})_L \tag{4.18}$$

The solution vector $\mathbf{U}_L$ and its gradient $(\nabla \mathbf{U})_L$ have to be provided at the quadrature points of the element faces of the outer boundary, $\partial\Omega_E$. To do so, the coordinates of those quadrature points, which are denoted as $\mathbf{x}_q$, are provided to the Lagrangian solver at the initialization of the calculation. As the spatial configuration of the computational domains here does not change

in the course of the simulation, the coordinates are stored in the Lagrangian solver for further use. The boundary conditions are computed at the end of each time step of the Lagrangian solver following Algorithm 2. As the operation is here performed exclusively within the Lagrangian solver, the index L is dropped for convenience purpose. As stated in Eq. (Alg. 2.1), the Eulerian state vector $\mathbf{U}$ is first computed on the Lagrangian grid using the velocity field and the Lagrangian state vector, $\mathbf{F}$, which are available on the grid at the end of the time step, as a P2M operation has been performed. The gradient of the Eulerian state vector is then computed using finite differences (Eq. Alg. 2.2). Using a single M2P operation, they are then both projected on the quadrature points.

Algorithm 2: Computation of the BCs for the Eulerian solver

Given: $\mathbf{x}_q,\ \mathbf{u}_{i,j}^{n+1},\ \mathbf{F}_{i,j}^{n+1}$
Compute:

$$\mathbf{u}_{i,j}^{n+1}, \mathbf{F}_{i,j}^{n+1} \longrightarrow \mathbf{U}_{i,j}^{n+1} \qquad \text{(Alg. 2.1)}$$

$$\mathbf{U}_{i,j}^{n+1} \xrightarrow{\text{FDs}} (\nabla_h \mathbf{U})_{i,j}^{n+1} \qquad \text{(Alg. 2.2)}$$

$$\mathbf{U}_{i,j}^{n+1}, (\nabla_h \mathbf{U})_{i,j}^{n+1} \xrightarrow{\text{M2P}} \mathbf{U}_{q}^{n+1}, (\nabla_h \mathbf{U})_{q}^{n+1} \qquad \text{(Alg. 2.3)}$$

The projection (Eq. (Alg. 2.3)) could also be performed using a matrix-matrix multiplication if the coordinates of the quadrature points do not vary in time, as it is the case in this work. At each time step, the matrix Q containing the state vector of size N_v and its gradient for each quadrature point q, i.e. the result of Eq. (Alg. 2.3), would be computed as

$$Q(t^{n+1}) = W \cdot G(t^{n+1}) \tag{4.19}$$

where $W \in \mathbb{R}^{N_q \times N_p^{bc}}$, $G \in \mathbb{R}^{N_p^{bc} \times N_v^{bc}}$ and $Q \in \mathbb{R}^{N_q \times N_v^{bc}}$ with $N_v^{bc} = N_v + (N_v \times N_d)$, N_d being the number of dimensions. The pre-computed and very sparse matrix, W, contains the weights of a subset of N_p^{bc} Lagrangian grid points contributing to the boundary conditions at the N_q quadrature points using a given interpolation kernel. The matrix G contains the values of the N_v^{bc} grid fields of the boundary conditions (with $N_v^{bc} = N_v + (N_v \times N_d)$ where N_v is the size of the Eulerian state vector and N_d is the number of dimensions).

As explained in Chapter 1, the boundary conditions to be provided to the Eulerian solver are computed at the end of each time step of the Lagrangian

solver, i.e. right after the penalization step. In order to improve the computation of the boundary conditions, it could prove useful to solve an additional Poisson problem just after the penalization, to propagate the information contained in the vortex sheet into the volume. At this stage, the velocity field is indeed not fully consistent with the penalized vorticity and dilatation fields. This would actually call for an iterated penalization as in [67, 59]. However, as the influence of the vortex sheet decreases as $1/r$ in 2-D, this should not have a significant impact on the boundary conditions. However, we did not investigate the more consistent algorithm with iterated penalization.

Finally, a linear interpolation is used to compute the boundary conditions at the time required by the Eulerian solver. Indeed, the time at which the Eulerian solver requires the boundary conditions from the Lagrangian solver does not necessarily correspond to one of the time steps computed by the Lagrangian solver. The boundary conditions at time t_L^n and t_L^{n+1} are thus linearly interpolated at time $t_E \in \left[t_L^n, t_L^{n+1}\right]$.

A mesh size requirement for the Lagrangian grid results from the high-order capability of the Eulerian solver: both spatial discretizations should be able to represent the same largest wavenumber in regions where we want to preserve the same level of information, i.e. near the outer boundary of the Eulerian solver. In other words, they should have the same cutoff wavenumber, k_c. For the Lagrangian solver, it is such that:

$$k_c h = \pi \tag{4.20}$$

with h the mesh size. The basis functions used in Argo are Lagrange interpolants based on equispaced interpolation points [71]. Considering a 1-D problem, evenly spaced elements of size h_E and interpolation functions of order p, the cutoff wavenumber for such discretization is thus

$$k_c \frac{h_E}{p} = \pi \tag{4.21}$$

as $p+1$ interpolation points are used. As a consequence, the discretization in both solvers have the same cutoff wavenumber if

$$h_L = \frac{h_E}{p}. \tag{4.22}$$

In higher dimensions, an equivalent requirement could be

$$h_L \simeq \left(\frac{L_e^{\min}}{p}\right)_{\partial\Omega_E} \tag{4.23}$$

where $L_e^{\min}$ is a minimal characteristic length; it could for instance be chosen as the smallest interface length on the outer boundary of the Eulerian mesh. This notion of *effective resolution* is often used to allow proper comparison between different spatial discretizations. In [24], it was used in the context of a grid convergence study to compare the performance of Argo against the results obtained using a pseudo-spectral code on the benchmark problem of the three-dimensional Taylor-Green vortex. In [52], the division of each distance of the mesh by the interpolation order p was used to compare the refinement criteria in Argo with those used with finite volume and finite difference solvers.

4.4 Correction of the Lagrangian solution in the Eulerian subdomain

In the present completely overlapping domain decomposition technique, the strengths of the particles close to the body are corrected using the solution of the accurate near-body Eulerian solver. To be more specific, only the particles that are "sufficiently" remote from the outer boundary of the Eulerian solver will be corrected to avoid spurious interactions between the correction and the BCs; this internal subdomain is denoted as Ω_E^c and is such that $\Omega_E^c \subset \Omega_E$. This will be covered in detail in Section 4.4.3. Irrespective of the correction methodology, the goal is to compute corrected values for the strengths vector

$$\mathcal{F}_p^c = \mathbf{F}_p^c S_p = \Big[\rho, \boldsymbol{\omega}, \theta, \rho T\Big]_p^{c,T} S_p, \quad \forall p \in \Omega_E^c \subset \Omega_E \subset \Omega_L \tag{4.24}$$

using the information from the Eulerian solver. We here elected to use the conservative variables, $\mathbf{U}$, as the building blocks to compute the correction, $\mathcal{F}_p^c$. The quadrature rules will thus involve the variables used as the primary variables of the Eulerian solver. The correction procedure then requires to obtain the representation in the Lagrangian solver of the solution vector computed in the Eulerian solver:

$$\mathcal{U}_{E,p} = \mathbf{U}_{E,p} S_p = \Big[\rho, \rho\mathbf{u}, \rho E\Big]_{E,p}^{T} S_p, \quad \forall p \in \Omega_E \subset \Omega_L \tag{4.25}$$

At this point, a dichotomy of the envisioned correction procedures that will be presented can be introduced. The correction procedure may involve:

- a quadrature directly performed on the Lagrangian particles ($\equiv$ cells),
- a quadrature on the elements of the Eulerian solver and a projection of the resulting integrals at the location of the particles.

4.4.1 Computation of the correction using quadrature rules on the Lagrangian particles

The technique that is here presented belongs to the first class of correction mentioned above; a Gauss-Legendre quadrature is performed on the elements of the Lagrangian grid[2]. A parent or reference quadrilateral cell is here also used for the ease of implementation. At the initialization of the solvers, the coordinates of the quadrature points, $\mathbf{x}_q$, for all the particles in the Eulerian domain Ω_E are sent to the Eulerian solver. The Eulerian solver then identifies which element contains each Lagrangian quadrature point; the result is thus a list of Lagrangian quadrature points for each element, the quadrature points inside the obstacles being discarded. The parametric coordinates, γ_q, of these Lagrangian quadrature points of the current element are also stored. This preparation of the data will allow a more efficient computation of the correction; it indeed allows to loop over the Eulerian elements and compute the required quantity on each quadrature points located in this element rather than looping on the quadrature points and thus constantly switching from one element to another. This implementation is also inspired by the non-matching boundary conditions; it indeed requires to identify which element of the neighbor domain contains a given quadrature point of the first domain.

The correction procedure of Algorithm 3 is then applied at time $t = t_L = t_E$. It is also illustrated in Figure 4.5. The solution vector is first interpolated at the position of the Lagrangian quadrature points (Equation (Alg. 3.1) in Algorithm 3). This operation is similar to the collocation operation of Eq. (4.16). For each category of element, a matrix-matrix operation involving the collocation matrix for that category can be performed. For the sake of clarity, the algorithm is here presented for a mesh composed of elements of the same category (same type and same interpolation order). The superscript $\star$ has also been used to distinguish the collocation matrices of Eq. (4.16) and Eq. (Alg. 3.1). Here again, the collocation matrix $\mathfrak{C}^\star$ can be precomputed at the initialization stage as the coordinates of the Lagrangian quadrature points do not change in time[3]. The Gauss-Legendre quadrature is then performed on the Lagrangian grid using Eq. (4.15) and the trivial mapping between the cartesian cell of area $S = h^2$ in the real space and its counterpart in the parametric

[2] As explained later, a correction directly performed on the Lagrangian particles would be more expensive.

[3] Although envisioned as explained in the text, Equation (Alg. 3.1) was, however, not implemented as efficiently; the evaluation of the shape functions at the quadrature points was performed at each time step. The storage of the collocation matrix (or matrices in the general case) is, however, recommended.

space. If necessary, the corrected values computed on the Lagrangian grid are then projected onto the particles through a M2P operation (Equation (Alg. 3.3)). Another possibility would be to compute the quadrature directly on the particles. It would, however, require to go through the expensive pairing of the Lagrangian quadrature points with the elements in which they are located at each time step. The collocation matrix would also need to be recomputed at each time step.

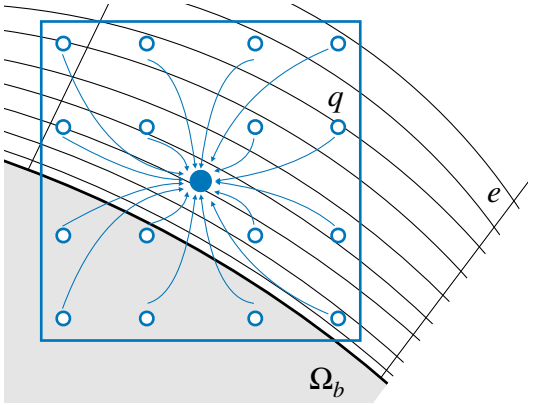


Figure 4.5: Gauss quadrature in an isolated VPM cell (blue) at the vicinity of an obstacle Ω_b. The solution vector is first interpolated on the quadrature points of the VPM cell (open blue circle) using the solution defined on the underlying Eulerian elements. The arrows depict the Gauss quadrature performed on the VPM cell.

The circulation and dilatation fields were, however, not corrected using the above methodology. By definition, these fields involve the first derivative of components of the solution vector. It would thus require to compute its gradients at the Lagrangian quadrature points. Following [71], the computation of the gradients requires to first interpolate the parametric gradients to the quadrature points

$$\mathfrak{g}^{\star u}_{qm} = \left(\frac{\partial u_m}{\partial \xi^u}\right)_{\gamma_q} = \sum_{i=1}^{N_\phi} \left(\frac{\partial \phi_i}{\partial \xi^u}\right)_{\gamma_q} \mathbf{U_{im}} = \mathfrak{G}^{\star u}_{qi} \mathbf{U}_{im} \tag{4.26}$$

where $\mathfrak{G}^{\star u}$ is the collocation matrix for the gradients of the solution vector and u is the index running on the directions in the parametric space. As Eq. (Alg. 3.1), its counterpart for the solution vector, this collocation is performed in the parametric space and can thus be performed using a matrix-matrix multiplication (per category of element). The computation of the Cartesian components

Algorithm 3: Correction step: quadrature on the Lagrangian grid

Given: $\gamma_q,\ \mathbf{U}_E(t)$
Compute:

1. Interpolation of the Eulerian state vector $U_E(t)$ to the Lagrangian quadrature points $\mathbf{x}_q$

$$\mathbf{U}^{\star}_{qm} = (u_m)_{\gamma_q} = \sum_{i=1}^{N_\phi} (\phi_i)_{\gamma_q} \mathbf{U}_{\mathbf{im}} = \mathfrak{C}^{\star}_{qi} \mathbf{U}_{im} \qquad \text{(Alg. 3.1)}$$

2. Evaluation of the volume integral of the Eulerian state vector on the Lagrangian cell

foreach *Lagrangian cell in Ω^c_E* **do**

$$\int_S u_m \, dS = \int_S u_m(\xi) |\mathbf{J}| \, d\xi \simeq \sum_{q=1}^{N_\alpha} w_q (u_m)_{\gamma_q} |\mathbf{J}| \qquad \text{(Alg. 3.2)}$$

end

3. If the particles were not remeshed, project the correction on the Lagrangian particles using a M2P operation

$$\mathbf{U}^c_{i,j} h^2 \quad \xrightarrow{\text{M2P}} \quad \mathcal{U}^c_p = \mathbf{U}^c_p S_p \qquad \text{(Alg. 3.3)}$$

of the gradients are then computed as

$$\mathbf{g}^{\star k}_{qm} = \sum_{u=1}^{d} \left(\frac{\partial \xi^u}{\partial x^k} \right) \mathfrak{g}^{\star u}_{qm} \qquad (4.27)$$

Besides the additional storage of the inverse of the mapping Jacobian (a 3x3 matrix per Lagrangian quadrature point in 3D), the computation of the Cartesian components of the gradients cannot be done as efficiently as the collocation operations (i.e. using a matrix-matrix operation); an additional loop on the dimensions is required.

The circulation and the dilatation fields were rather obtained using contour integrals ensued from Green's theorem,

$$\int_S \omega \, dS = \int_S (\nabla \times \mathbf{u}) \cdot \hat{\mathbf{e}}_z \, dS = \oint \mathbf{u} \cdot \mathbf{s} \, dl \qquad (4.28)$$

$$\int_S \theta \, dS = \int_S \nabla \cdot \mathbf{u} \, dS = \oint \mathbf{u} \cdot \mathbf{n} \, dl \qquad (4.29)$$

where $\mathbf{s}$ and $\mathbf{n}$ are the clockwise unit tangent and outward unit normal vectors, respectively. The implementation is essentially the same as for the volume integrals; the quadrature rule is the only difference. For the same reason as with the quadrature for the volume integrals, the numerical quadratures were also performed on the Lagrangian grid. The Lagrangian quadrature points were obtained by a recursive division of the faces of the Lagrangian cells as illustrated in Figure 4.6 (using $l_p = 3$ recursive subdivisions). The boundary condition associated with the velocity field was imposed for the values of the quadrature points that are within the body.

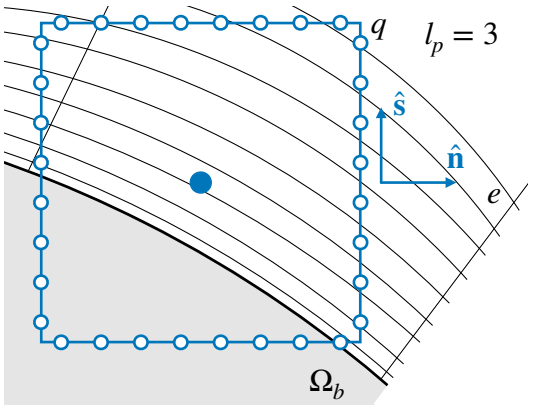


Figure 4.6: Contour integrals on an isolated VPM cell (blue) at the vicinity of an obstacle Ω_b. The solution vector, $\mathbf{U}$, is first interpolated on the quadrature points of the VPM cell (open blue circle) using the solution defined on the underlying Eulerian elements. The clockwise unit tangent vector $\mathbf{s}$ and the outward unit normal vector $\mathbf{n}$ of the right face are also represented.

As already stated in [102], a quadrature on the Lagrangian particles, either for volume integrals or contour integrals, is, however, not fully conservative and its accuracy might, furthermore, be problematic in the near-wall region for high-Reynolds number flows; as the velocity field and the vorticity field vary rapidly in the boundary layer and as the Lagrangian solver is there purposely under-resolved, the quadrature will most probably miss a part of the (integral) information contained in the boundary layer. In order to slightly improve the accuracy of the quadrature in the VPM cells in the near-wall region, one could resort to using a high order quadrature rule for the volume integrals in order to increase the number of quadrature points in a given VPM cell. In the same spirit for contour integrals, the number of subdivisions could be boosted. In the current implementation, a unique quadrature order and a

unique recursion level, l_p, are used for all the elements although those parameters could be easily tuned elementwise using the ratio between the smallest Eulerian element in the VPM cell and the Lagrangian mesh spacing. Increasing the number of quadrature points is, however, a band-aid solution and it motivates the computation of the correction using a quadrature on the Eulerian elements, which exhibits a suitable resolution in the near-wall region.

4.4.2 Computation of the correction using a quadrature on the Eulerian elements

As noted above, a natural alternative, which was already suggested in [102], consists in computing the integrals on the material volumes corresponding to the elements of the Eulerian solver. When associated to a *mean* position, these integrals may be seen as the strengths of "Eulerian particles". These strengths could then be redistributed on the Lagrangian particles using an appropriate kernel. This is depicted in Figure 4.7 for the same configuration as in Figure 4.5. As in the previous section, we will, however, here again refrain from computing the gradients of the solution vector to obtain the circulation and the dilatation. The methodology presented in this section can thus only be used to compute the Lagrangian representation of the Eulerian solution strengths vector, $\mathcal{U}_{E,p}$. Of course, such an approach is particularly fitting as the computation of integrals is in a sense a key operation in the DG solver. As a consequence, each and every tool is already at hand to efficiently evaluate the correction.

Difficulties, however, arise with the projection of the "Eulerian particles" onto the Lagrangian particles. Indeed, as already stated in Section 4.3, the elements of the Eulerian mesh in the vicinity of the outer boundary $\partial\Omega_E$ are bigger than their Lagrangian counterparts, which could be written as

$$h_L \simeq \frac{L_e^{\min}}{p} \tag{4.30}$$

This situation is schematically depicted in Figure 4.8 for a 1-D problem. The challenge is then to determine how to perform this "coarse-to-fine" redistribution of the Eulerian particles onto the Lagrangian particles. Despite being conservative, a simple nearest-neighbor projection may not be applied as it would indeed not provide a correction to each particle, thus leading to "holes" in the correction fields.

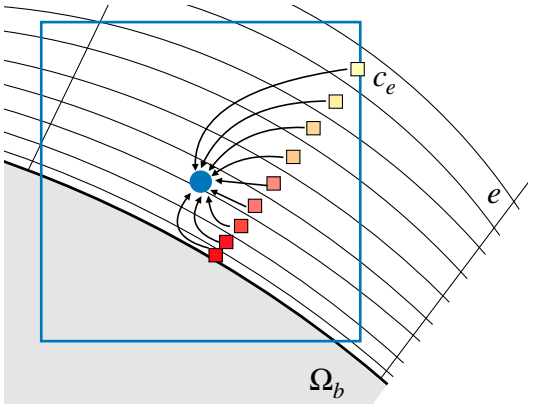


Figure 4.7: Gauss quadrature performed in the Eulerian elements. The integrals are first evaluated on the elements e of the Eulerian mesh and associated with the centroid of the element, denoted as c_e. The centroid of an element is colored by the value of the corresponding integral. The arrows depict the projection of the integrals on the nearest Lagrangian particle (blue dot).

A mixed situation is encountered near the wall of the obstacle for high Reynolds number flows; in that region, the Eulerian mesh is a boundary layer mesh with highly stretched elements while the Lagrangian mesh is purposely under-resolved (see Figure 4.7). The following relation could then possibly hold

$$L_e^{\min} << h_L \leq L_e^{\max} \tag{4.31}$$

where $L_e^{\min}$ and $L_e^{\max}$ are respectively the smaller and bigger characteristic lengths of the unstructured element e in the vicinity of the wall. $L_e^{\min}$ and $L_e^{\max}$ will typically reflect the size of the stretched element in the direction normal and tangential to the wall, respectively. This zone thus requires a "fine-to-coarse" projection; a simple projection on the nearest Lagrangian cell should suffice and be conservative. However, using this nearest-neighbor approach may also cause holes in the correction fields when using boundary layer meshes. It is illustrated at Figure 4.9 where the upper left Lagrangian cell (in red) won't receive any correction. Part of the correction may also be projected on particles inside the obstacle. This last aspect will be treated in Section 4.4.3 in conjunction with the penalization method introduced in Chapter 3.

Despite being of less concern than in the methodology using a quadrature on the Lagrangian particles, the near-wall region thus also requires to adapt the methodology when the largest characteristic length of the Eulerian ele-

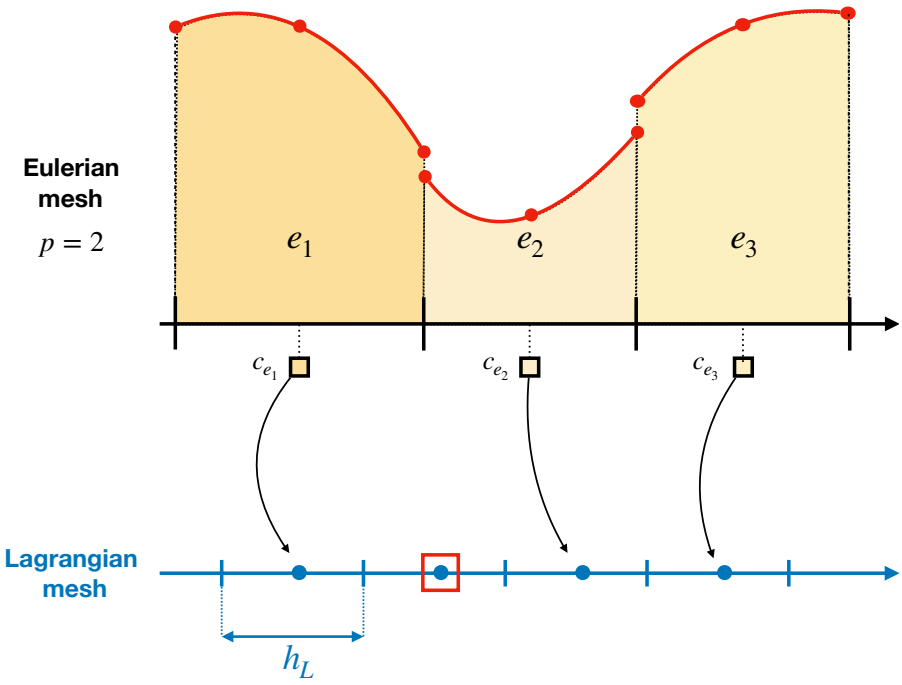


Figure 4.8: Tentative correction in the outer region of the Eulerian domain (i.e such $h_L < L_e^{\min}$) : elements ($p = 2$) of centroids c_e (open black square) and Lagrangian mesh (blue) with the remeshed particles (blue dots). The centroid of an Eulerian element is colored by its integral. The association of the integral of the element with its centroid is assimilated to an "Eulerian particle", which is projected onto the nearest Lagrangian particle. The second particle (in red) is not corrected by any Eulerian element.

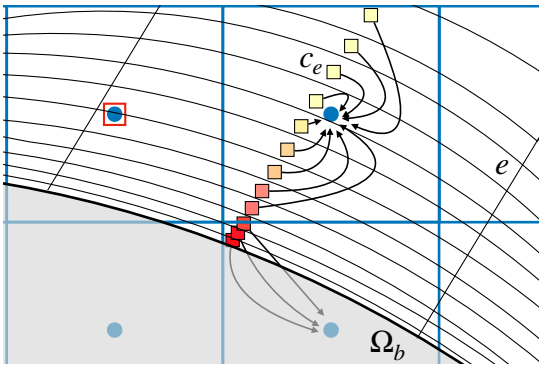


Figure 4.9: Gauss quadrature performed in the Eulerian elements. No correction will be provided to correct the upper left Lagrangian cell (blue dot). Moreover, the middle lower Lagrangian cell located inside the obstacle will receive a correction from the Eulerian solver.

ment is larger than the Lagrangian mesh spacing.

One of the possible approaches to deal with this challenge is to perform a subdivision of the Eulerian elements. Figure 4.10 illustrates this methodology on the same 1-D problem as in Fig. 4.8. The decomposition of elements into smaller entities aims to allow the computation of the integrals on the Eulerian (sub)elements and then to project those integrals on the nearest particles. As illustrated in Fig. 4.10, an element e of the Eulerian mesh is decomposed into subelements denoted as s_e, of centroid c and whose characteristic size is at most of the order of the grid spacing of the Lagrangian solver h_L,

$$L_{s_e}^{\max} \leq h_L \quad \forall s_e \in e \tag{4.32}$$

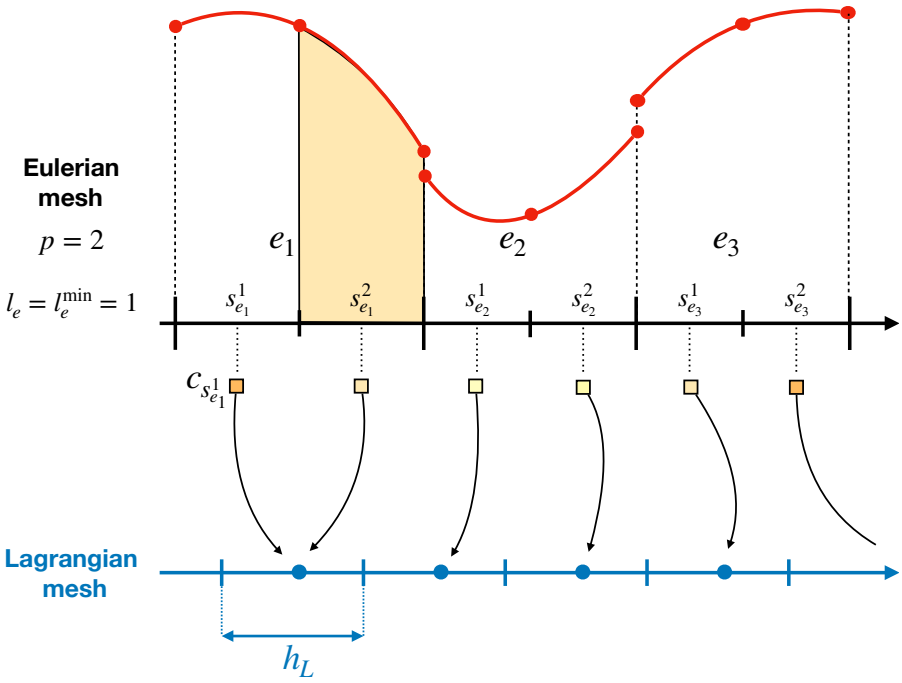


Figure 4.10: Correction in the outer region of the Eulerian domain (i.e such $h_L < L_e^{\min}$) using subdivided elements ($l_e = l_e^{\min} = 1$): Eulerian mesh using second order interpolation function (black) and Lagrangian mesh (blue) with the remeshed particles (blue dots). The centroids of the subelements (open black square) are colored by the integrals.

This is indeed the minimal requirement to allow a projection of the integrals on the subelements onto the nearest Lagrangian particles using a simple *nearest-neighbor* approach. A recursive element subdivision is here imple-

mented. For each element e, the minimal recursion level, denoted as $l_e^{\min}$ is chosen according to the requirement of Equation (4.32), i.e.

$$l_e^{\min} = \max\left(\log_2\left(\frac{L_e^{\max}}{h_L}\right), 0\right) \tag{4.33}$$

where $L_e^{\max}$ is here chosen as the diameter of the circumscribed circle if the element is a triangle or the length of the longest edge of the element otherwise. This quantity is obtained using a function implemented in the MAdlib library, the open source Mesh Adaptation Library used in Argo. However, as the recursively subdivided Eulerian mesh and the Lagrangian mesh do not coincide, some Lagrangian cells might still not receive any correction. The effective recursion level is thus chosen such that the longest dimension of the subelement is only a fraction $1/f$ of the Lagrangian mesh size

$$\frac{L_{s_e}^{max}}{h_L} = \frac{L_e^{\max}}{2^{l_e^{\min}}}\frac{1}{h_L} = \frac{1}{f}. \tag{4.34}$$

The effective recursion level is then chosen to be

$$l_e = \max\left(\log_2\left(\frac{L_e^{\max}}{h_L}\right), 0\right) + \log_2(f). \tag{4.35}$$

In this work, a rather conservative choice was made; the longest dimension of a subelement was chosen to be four times smaller than h_L, which is equivalent to

$$l_e = l_e^{\min} + 2. \tag{4.36}$$

This choice should, however, not be seen as definitive. In particular, the $\log_2(f)$ term has been placed outside the maximum function in Eq. (4.35) in order to force the decomposition of the near-wall elements. This conservative choice was, however, not tested against less expensive alternatives. Figure 4.11 shows the effective level of of recursion, l_e, computed using Eq. (4.36) on a typical mesh that would be used for the simulation of a flow past a NACA-0012 profile at moderate Reynolds number flows using interpolation functions of order $p = 4$.

The small relative size of the Eulerian subelements and the Lagrangian mesh spacing motivated the nearest-neighbor approach for the projection of the correction. Indeed, as the interpolation kernel operates at the Lagrangian mesh size level, a higher order and thus wider interpolation kernel (e.g. the M_4' scheme) would project part of the integral to a Lagrangian particle that is remote from the centroid of the subdivided Eulerian element, which would

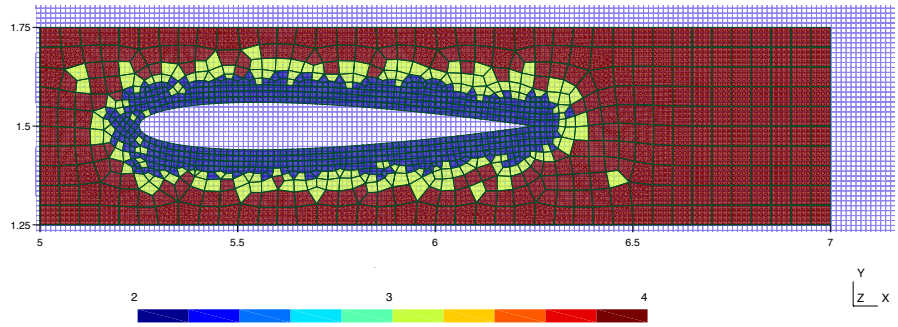


Figure 4.11: Effective recursion level l_e for each element of the Eulerian mesh. The elements are colored by their recursion level l_e. The truncated Lagrangian grid completely overlapping the Eulerian mesh is also represented (solid blue line).

impair the second order moment of the corrected quantity. Another point of attention is that, in the present implementation, the elements are decomposed into the same number of subelements in each direction. This "overkill approach" is, however, not necessary (and expensive) in the near-wall region when considering high-Reynolds number flows, for which Equation (4.31) holds. A further refinement of the methodology would thus need to better account for the aspect ratio of the elements.

The recursion level now being identified elementwise, the integral over a subelement, s_e, can be computed using the numerical quadrature of Equation (4.15)

$$\int_{s_e} a dV = \int_{s_e} a(\xi)|\mathbf{J}_{s_e}|d\xi \simeq \sum_{q=1}^{N_\alpha} w_q a(\gamma_q)|\mathbf{J}_{s_e}|_{\gamma_q} \tag{4.37}$$

where γ_q are the coordinates of the quadrature points of the subelements. Equation (4.37) requires to interpolate the solution from the interpolation points of the elements to the quadrature points of their subelements. As in the regular case on the elements (Eq. (4.16), this collocation can be performed efficiently for all the elements of the same category c and recursion level l by performing a matrix-matrix multiplication. It thus now involves the collocation matrix $\mathfrak{C}^{c,l}$, i.e. the collocation matrix for the element of category c and required recursion level l. Contrary to the implementation of the first approach presented in Section 4.4.1, these matrices are actually precomputed and stored at the initialization stage. Equation (4.37) also requires the weights as well as

the determinant of the Jacobian at the position of the quadrature points of the subelements. This information is also computed at the initialization stage and stored to evaluate Equation (4.37) at each time step. To be more explicit, a loop over the Eulerian elements is performed at the initialization stage in order to identify their required recursion level l_e. For each category of element in the mesh and for each required recursion level, the corresponding parent element is recursively subdivided. It is illustrated in Figure 4.13 for a single subdivision of a quadrilateral element, here using a first order mapping and a second order interpolation. The above mentioned information are then stored for the pair (c, l). The area as well as the centroid of the subelements are also computed using the numerical quadrature (Eq. (4.37)). The centroid is obtained using

$$\mathbf{c}_{s_e} = \frac{\int_{s_e} \mathbf{x}\, dS}{\int_{s_e} dS}. \tag{4.38}$$

In this work, the Lagrangian particles were remeshed before the correction step. Considering a single remeshed Lagrangian particle p, the correction for a field f resulting from the projection using a nearest-neighbor approach[4] may be written as

$$\sum_{s}^{n_s^p} \int_s f_{\mathrm{E}}\, dS \triangleq \bar{f}_{E,p} S_{E,p} \triangleq \alpha_{E,p} \tag{4.39}$$

where n_s^p is the number of subelements whose integrals contribute to the correction of the particle p, the surface of those subelements being computed as $S_{E,p} \triangleq \sum_s^{n_s^p} \int_s dS$. This is depicted in Figure 4.12. It is important to notice that $S_{E,p}$, the support of integration of the correction, Eq. (4.39), is different from the surface of the remeshed particle $S_p = h^2$; the surface $S_{E,p}$ should actually be seen as a correction for S_p. As a consequence and as the correction has been provided on the Lagrangian grid, the grid representation of the correction is given by

$$f_{E,i,j} = \frac{\bar{f}_{E,p} S_{E,p}}{S_{E,p}} = \bar{f}_{E,p}, \tag{4.40}$$

which was observed to provide a smooth representation of the Eulerian solution on the Lagrangian grid, even for Lagrangian grid cells intersected by the solid wall and for which $S_{E,p} < h^2$. As explained in [103], to obtain smooth fields is indeed important as it would otherwise cause spurious oscillations in the evaluation of the RHS of the conservation equation in the Lagrangian time step. We here also elected to renormalize the integral of the correction to

[4] One could also perform the projection onto the nearest particles using a matrix-matrix operation (if it is performed on the Lagrangian grid).

match the material area of the remeshed particles. The correction, Eq. (4.39), is thus multiplied by the ratio of the surfaces, which reads

$$\alpha^r_{E,p} = \left(\bar{f}_{E,p} S_{E,p}\right) \frac{h^2}{S_{E,p}} = \bar{f}_{E,p} h^2 \tag{4.41}$$

where $\alpha^r_{E,p}$ is the renormalized correction strength. Equation (4.41) indeed constitutes a renormalization from the Eulerian mesh to the Lagrangian mesh. Unfortunately, this choice amounts to turn a blind eye on the fact that the surfaces $S_{E,p}$ on which the corrections are computed in the Eulerian solver and h^2 are different and it thus prevents the correction procedure to be conservative,

$$\sum_p^{N_p^c} f_{E,p} h^2 \neq \sum_p^{N_p^c} f_{E,p} S_{E,p} \tag{4.42}$$

where N_p^c is the number of remeshed particles receiving a correction. Different schemes presented in Appendix E were tested in order to enforce the conservation. Unfortunately, no variant could enforce this global property without noticeably interfering with the local behavior of the corresponding field.

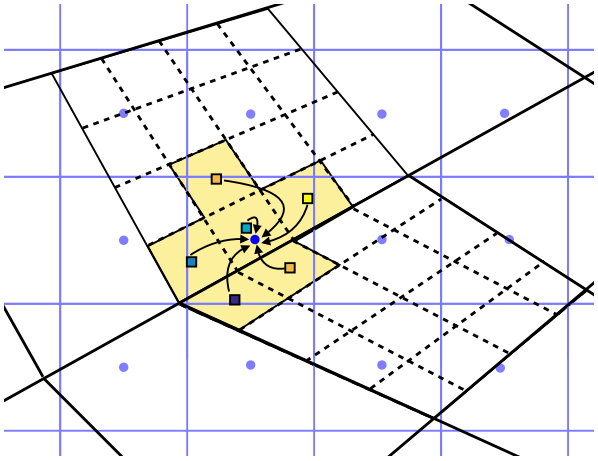


Figure 4.12: Correction procedure using elements subdivided $l = l^{\min} + 1 = 2$ times. A total of $n_s^p = 6$ subelements of different elements contribute to the correction of the remeshed particle p (blue dot). The area of the union of the subelements, $S_{E,p}$, is highlighted in yellow.

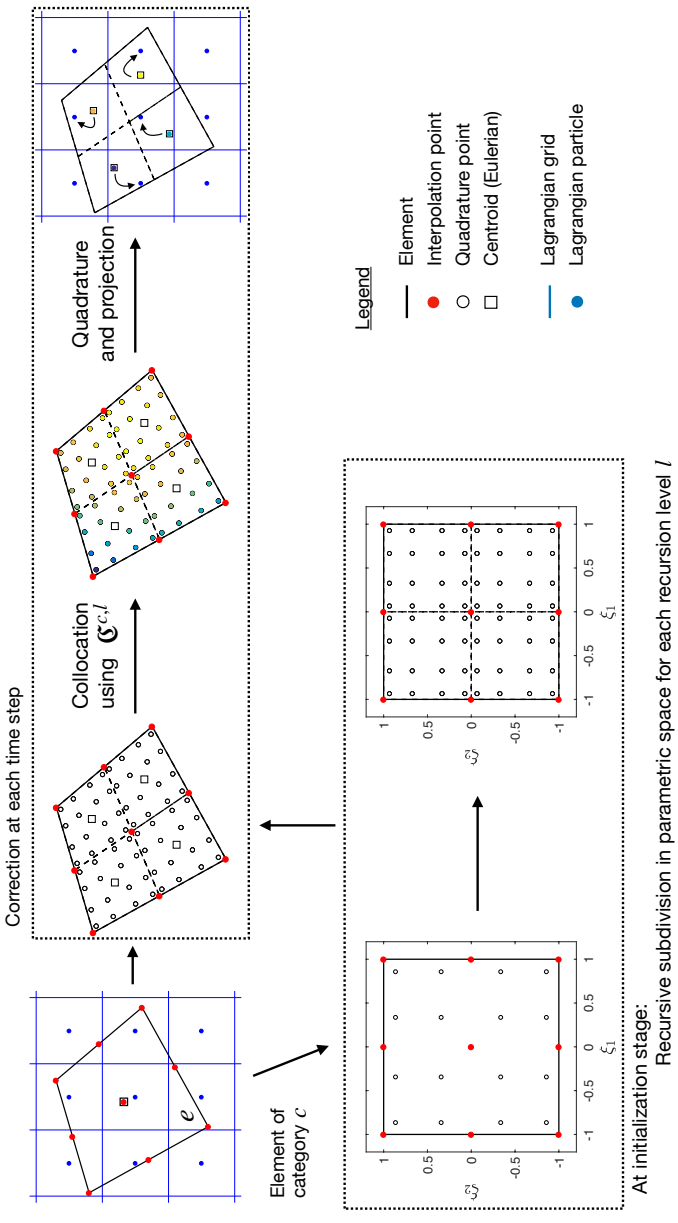


Figure 4.13: Correction procedure. At the initialization stage, the parent element of each category of element in the Eulerian mesh is recursively subdivided for each recursion level l that is required in the mesh. The following information is stored: the weights and the determinant of the Jabobian at the position of the quadrature points of the subelements, their area and centroids as well as the collocation matrix $\mathfrak{C}^{c,l}$. At each time step, efficient matrix-matrix multiplications are performed for the collocation. Finally, the numerical quadrature is performed using Eq. (4.37), the results of which is then projected on the Lagrangian grid using a nearest-point approach.

4.4.3 Blending of the correction and of the current Lagrangian solution

Different methodologies allowing to obtain the representation of the Eulerian solution on the Lagrangian particles, $\mathcal{U}_{E,p}$, were presented in Section 4.4 and can readily be translated to the thermodynamic variables used in the VPM formulation. Contour integrals were also used to obtain the corrections for the circulation and the dilatation fields. The methodology used to obtain the effective corrected strengths, $\mathcal{F}_p^c$, from these corrections will now be presented.

In theory, the solutions computed by both solvers should be identical at the outer boundary of the Eulerian domain; the correction step would have no effect and the boundary conditions for the Eulerian solver would be provided without additional error. In practice, however, a blending operation is carried out in order to ensure a smooth transition between the Eulerian solution and the Lagrangian solution. The discrepancies between the solutions would indeed result in spurious interactions between the imposition of the boundary conditions of the Eulerian solver and the correction step. The Lagrangian solution to which the solution is progressively relaxed in the correction region should evidently be of adequate quality. Regardless of the methodology employed to compute the corrections vector from the Eulerian solver, $\mathcal{F}_{E,p}$, it will be blended with the uncorrected Lagrangian solution, $\mathcal{F}_p$. This blending will vary in space by means of a smoothed Heaviside function $H(\xi)$,

$$\mathcal{F}_p^c = H(\xi_p)\,\mathcal{F}_{E,p} + (1 - H(\xi_p))\,\mathcal{F}_p \tag{4.43}$$

where $\xi_p = 2\frac{(x_p - x_c)}{l_b}$, the nondimensionalized signed distance to the middle of the blending transition region of length l_b and whose center is at $x = x_c$, with $\mathbf{x}_p$ the position of the particle p. The Heaviside function is defined as

$$H(\xi) = \begin{cases} 0 & \xi < -1 \\ \frac{1}{2}\left(1 + \xi + \frac{1}{\pi}\sin(\pi\xi)\right) & -1 \leq \xi \leq 1 \\ 1 & \xi > 1 \end{cases} \tag{4.44}$$

This function is symmetric with respect to $\xi = 0$ and its first and second derivatives are zero at $\xi = -1$ and $\xi = 1$.

At this point, it is important to note that the integral corrections have been computed in the whole Eulerian domain, Ω_E, without special treatment near the boundaries. In particular, the outer boundary of the Eulerian domain does not necessarily match the faces of Lagrangian elements. As a consequence,

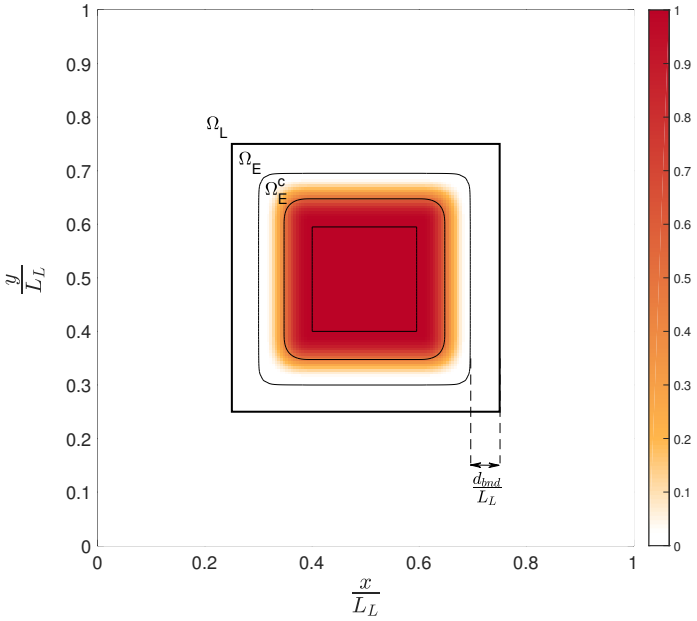


Figure 4.14: 2-D blending function using the smoothed Heaviside function Eq. (4.44): exterior boundaries of the Eulerian domain (bold solid line), contours of the blending function in the range [0, 1] in increment of 0.5 (thin solid lines). L_L is the length of the Lagrangian domain.

the corrections provided to the Lagrangian grid cells intersecting $\partial\Omega_E$ are incomplete and should be discarded. To do so, a supplementary buffer region where the Lagrangian solution is not corrected, and of width d_{bnd}, is placed ahead of the blending transition region inside the Eulerian subregion. In short, only the Lagrangian particles located inside the subregion $\Omega_E^c \in \Omega_E$ are corrected by the Eulerian solution through the blending function. In higher dimension, the blending function is obtained as the tensor product of the 1-D function in each direction, when a rectangular Eulerian subdomain is considered. This provides a smoother blending function in the corners of the transition region. An example of the corresponding 2-D blending function is depicted in Fig. 4.14. Note that the widths of the buffer and transition regions have been exaggerated for the sake of the clarity. Typical values used in this work are $d_{bnd}/h_L = 2$ and $l_b/h_L = 8$.

Another aspect of the blending of the two solutions is the role played by the penalization inside the obstacle in the Lagrangian solver. At this point, a correction for the Lagrangian solution has been provided outside the obstacle.

A correction for the penalized fields inside it should likewise be provided: numerical experiments on the flow past a cylinder showed the necessity to reenforce the kinematic conditions inside the obstacle at the correction step. The circulation and dilatation of the particles inside the obstacles are here reset to zero.

In Section 4.3, a remark regarding the consistency between the velocity and the vorticity and dilatation fields was made for the computation of the boundary conditions. A similar remark could be made here; the vorticity and the dilatation fields have been corrected by the Eulerian solver in the near-wall region and reenforced to $\boldsymbol{\omega} = 0$ and $\theta = 0$ inside the solid obstacle. A residual velocity due to the errors made in the computation of the correction will, however, exist inside the obstacle. As a consequence, an additional penalization step was performed before pursuing the time step in the Lagrangian solver in order to make the correction compliant with the Lagrangian framework. In a sense, this additional penalization step is similar to what is done in an iterated penalization [67, 59].

4.5 Additional features

In this section, additional features complementing the VPM solver for the simulation of the flow past a cylinder will be presented.

4.5.1 Sponge layer

The radiation boundary condition used in Chapter 2 was based on the assumption that only acoustic disturbances (i.e. small amplitude disturbances) reach the exterior boundaries of the Lagrangian domain. This radiation boundary condition would not be suited to simulations of flow past obstacles. Numerous techniques allowing disturbances of larger amplitude to leave the computational domain exist in the literature (see for example [35] and references therein). We here elected to use a simple sponge layer approach [78, 13] in addition to the radiation boundary condition. The sponge layer is located before the radiation boundary condition. The sponge layer approach is in many ways similar to a penalization method. Additional terms are also added to the Navier-Stokes equations to damp the solution to a prescribed reference solution. In the VPM method presented in Chapter 2, the equations are thus be modified to read,

$$\frac{\partial \mathbf{F}}{\partial t} + \nabla \cdot (\mathbf{F}\mathbf{u}) = \mathcal{L}(\mathbf{F}, \mathbf{x}, t) + \sigma(\mathbf{F}_{ref} - \mathbf{F}) \tag{4.45}$$

where $\mathbf{F} = [\rho, \omega, \theta, \rho T]^T$ is the solution vector of the 2-D VPM method and $\mathcal{L} = [\mathcal{L}_\rho, \mathcal{L}_\omega, \mathcal{L}_\theta, \mathcal{L}_{\rho T}]^T$ are the vector of the non-advective terms of the conservation equations. These additional terms can be treated in the exact same way as the penalization terms in Chapter 3. As explained in [13], the damping coefficient σ is generally a function of space only. The reference state may, on the contrary, be a function of both space and time. The damping terms are only active in a region near the artificial boundary, which is the domain of definition of the damping coefficient[5]. As its counterpart, λ, in a penalization method the damping coefficient σ is an inverse time scale. As initially stated by [13] and recalled in [101], the damping coefficient should (naturally) be the same for all conserved quantities to ensure stability.

In [101], Mani established guidelines to choose the parameters of a sponge layer whose profile is a power law,

$$\sigma(\mathbf{x}) = \sigma_0 \left(\frac{x}{l_{sp}} \right)^\beta \tag{4.46}$$

with l_{sp} the length of the sponge layer, σ_0 the magnitude of the sponge at the boundary, and β the exponent of the power-law. The strength of the sponge was defined with respect to the sound-sound reflectivity mechanism of a sponge layer. As stated in [101], it can be shown that the ratio of the reflected to incident amplitude of a sound wave reaching the sponge layer in the 1-D case is given by

$$\eta \triangleq \frac{L_{\text{reflected}}}{R_{\text{incident}}} = \eta_{\text{BC}} \exp\left(-\frac{2}{(1-M^2)} \int_0^{\tilde{l}_{sp}} \tilde{\sigma}(\tilde{x}) d\tilde{x} \right) \tag{4.47}$$

with η_{BC} is the "(sound-sound) reflection coefficient of the boundary coefficient at $x = l_{\text{sp}}$" and M the uniform freestream Mach number. Note that the quantities are here nondimensionalized using the freestream speed of sound, density and a length scale. The sponge strength was then defined as the opposite of the "target-damping", i.e. the target value in the decibel scale for the ratio $\frac{\eta}{\eta_{\text{BC}}}$ in Eq. (4.47) under 1-D condition

$$\eta_{\text{target}} \triangleq 20 \log \left[\exp\left(-\frac{2}{1-M^2} \int_0^{\tilde{l}_{sp}} \tilde{\sigma}(\tilde{x}) d\tilde{x} \right) \right] = 20 \log\left(\frac{\eta}{\eta_{\text{BC}}} \right) \tag{4.48}$$

$$= -20 \frac{2 \log e}{(1-M^2)} \int_0^{\tilde{l}_{sp}} \tilde{\sigma}(\tilde{x}) d\tilde{x} \tag{4.49}$$

[5]Similarly to what was done in Chapter 3 , the damping coefficient could be rewritten as $\sigma(\mathbf{x}) = |\sigma| \chi_{\text{sp}}(\mathbf{x})$, where the index "sp" would stand for "sponge".

which reduces to

$$\eta_{\text{target}} = -20\frac{2\log e}{(1-M^2)}\frac{\tilde{\sigma}_0\tilde{l}_{sp}}{(\beta+1)} \tag{4.50}$$

for the power law of Eq. 4.46. The parameters ensued from the guidelines provided in [101] will be specified for each test case in the remainder of the document.

4.5.2 Force computation

Two different techniques were considered to evaluate the force acting on a body. They both rely on the integral formulation of the momentum equation [112]. The first technique consists in evaluating the surface integral over the body of the stress tensor:

$$\boldsymbol{\mathcal{F}}_b = \oint_{S_b(t)} \underline{\underline{\boldsymbol{\sigma}}} \cdot \mathbf{n}\, dS \tag{4.51}$$

where $\mathbf{n}$ is the unit outward normal over the surface $S_b(t)$ of the body. This is generally the preferred methodology when using an Eulerian solver and a body-fitted mesh. The second technique employs a control volume that completely surrounds the body and requires to numerically evaluate the following equation [112]:

$$\boldsymbol{\mathcal{F}} = -\frac{d}{dt}\int_{V(t)} \rho\mathbf{u}\, dV - \oint_{S(t)} \mathbf{n}\cdot(\mathbf{u}-\mathbf{u}_s)\rho\mathbf{u}\, dS + \oint_{S^*(t)} \underline{\underline{\boldsymbol{\sigma}}} \cdot \mathbf{n}\, dS \tag{4.52}$$

where $S(t) = S_b(t) \cup S^*(t)$ is the surface of the control volume and $S^*(t)$ is its exterior surface. The unit outward normal over the surface is denoted as $\mathbf{n}$ while $\mathbf{u}_s$ is the velocity of the surface $S(t)$. For a fixed control volume ($\mathbf{u}_s = 0$ on $S(t)$) and with a no-slip boundary condition on the surface S_b, it reduces to:

$$\boldsymbol{\mathcal{F}} = -\frac{d}{dt}\int_{V} \rho\mathbf{u}\, dV - \oint_{S^*} \mathbf{n}\cdot\rho\mathbf{u}\mathbf{u}\, dS + \oint_{S^*} \underline{\underline{\boldsymbol{\sigma}}} \cdot \mathbf{n}\, dS \tag{4.53}$$

This second technique is used in the VPM solver. The volume and surface integrals in Eq. (4.53) are evaluated on the Lagrangian grid (using second order derivatives to evaluate the viscous stress tensor from the velocity field). The time derivative of the volume integral is then evaluated using a second order derivative able to handle a varying time step[6]. This control volume approach could also be employed using the solution of the Eulerian solver. However, the evaluation of the surface integral in Eq. (4.51) is generally more accurate as the mesh resolution at the wall is finer than in the volume. It is also easier to evaluate.

[6]This scheme was already used in the incompressible code to evaluate the force using the Noca's formulation.

4.6 Conclusions

In this chapter, the elements allowing to combine the VPM solver of Chapter 2 and Chapter 3 with a near-body Eulerian solver were presented.

The Eulerian solver responsible to capture the flow dynamics in the near-body region is Argo, a Discontinuous Galerkin solver developed at Cenaero (Gosselies, Belgium). This solver exhibits a high order accuracy on unstructured grid, with the additional capability of using curved elements. The locality of the method also allows an efficient parallelization. The DG method is furthermore locally conservative and has appealing dissipation and dispersion properties, a capability exhibited by the Lagrangian solver as well. All of these capabilities are looked upon very favorably in the context of a coupling methodology aiming at improving the quality of the solution near the obstacle at a reasonable computational cost, as discussed in the introduction. The existing coupling methodologies do not exhibit all of these benefits.

Referring to the notion of co-simulation introduced in Section 1.3.2, information are communicated between the solvers at times specified by the *orchestrator* following a *co-simulation scenario*. The scenario depends on the respective time steps used in the solvers as explained in Section 1.3.2. When combining a VPM solver with an Eulerian solver to capture the flow dynamics in the near-body region, these information are:

- the appropriate boundary conditions for the Eulerian solver; they are provided by the Lagrangian solver, which solves the Navier-Stokes equations in the whole domain while being purposely under-resolved in the near-body region. The formulation used in the DG solver requires to provide the conservative variables, $\mathbf{U} = [\rho, \rho\mathbf{u}, \rho E]$ as well as its gradient. These quantities are computed on the Lagrangian grid and then provided to the Eulerian solver using a high order interpolation kernel. The coupling methodology here greatly benefits from the use of a Riemann solver in the Eulerian solver. The Riemann solver indeed guarantees elementwise well-posed problems as it inherently prescribes the correct number of boundary conditions, as also stated in [23]. As a consequence, this part of the coupling algorithm is not completely agnostic about the Eulerian solver: it requires an implemented Riemann solver.

- the correction of the under-resolved Lagrangian solution in the near-body region as provided by the Eulerian solver. Two categories of correction were envisioned. The first family involves a quadrature on the

Lagrangian particles or on the underlying Lagrangian grid while the second family involves a quadrature on the Eulerian elements and a projection operation. Anticipating conservation problems for high Reynolds number flows due to the large size of the Lagrangian cells in the near-wall region, the second family was here favored. In this case, the first challenge is to account for the different possibilities of ratio between the size of the Eulerian elements and the VPM grid size that may appear throughout the Eulerian domain. To do so, an adaptive recursive subdivision of the Eulerian elements was implemented. The second challenge is the projection of this correction on the Lagrangian particles. A smooth representation of the correction on the Lagrangian grid is obtained by dividing the integral correction computed on the subelements by the corresponding area of these subelements. From the particles' point of view, this however amounts to a renormalization, which is a non-conservative operation. It will be demonstrated in Chapter 5 that this does not impair the quality of the solution. This methodology will be used to compute the correction of the thermodynamic quantities only. Indeed, the correction of the circulation and of the dilatation fields would require the (more) expensive computation of the derivatives of the solution vector in the Eulerian solver. As a consequence, we here elected to correct these fields using a correction belonging to the first category: contour integrals are computed on the VPM grid cells using an adaptable number of quadrature points.

Regardless of the methodology, the correction, which can be seen as the representation of the Eulerian solution on the Lagrangian grid, is blended with its counterpart from the Lagrangian solver in a narrow region close to the outer boundary. The purpose of this blending zone is to avoid any interaction between the boundary conditions imposed at the outer boundary and the correction step. A narrow region where the Lagrangian solution is less or even not corrected by the Eulerian solver thus exists inside the Eulerian subdomain ; this region may be referred to as the *trust region* as the Lagrangian solution is here preserved despite the existence of an Eulerian correction. As already stated in the Introduction of this chapter, there is no reason to "trust" the Lagrangian solver in the near-wall region; the additional *trust region* used in Stock et al. [147] was dismissed.

When compared to the existing correction procedures, a close attention was paid to this important step. The adaptive recursion performed on the Eulerian elements allows the conservative computation of the integrals to be used with a projection scheme throughout the computational domain. The method allowing to obtain a smooth representation on the Lagrangian grid was clarified in this thesis. However, using the same methodology for the particles amounts to perform a non-conservative renormalization (here only for the thermodynamic quantities). This renormalization as well as the (more) expensive evaluation of the velocity gradients in the Eulerian motivated the use of a different methodology to correct the kinematic quantities, the circulation and the dilatation: contour integrals on the Lagrangian cells were performed using an adaptable number of quadrature points. The coupled solver will here benefit from the adaptation and the adaptability of the two different techniques. Such capability was not reported in previous contributions.

The computation of the information to be exchanged as well as the orchestrator were here tailored for Argo. However, each and every technique presented in this Chapter could be readily adapted to use another Eulerian solver for compressible flows having a Riemann solver to treat the convective term of the conservation equations.

Although a high-order solver is used in the near-body region, with all the benefits it implies as mentioned above, it is important to stress that the coupling methodology itself does not exhibit a global spatial or temporal high-order capability. In particular, the current implementation of the coupling boundary conditions leads to a methodology that is first-order accurate in time.

The hybrid DG-VPM solver is restricted to cases where the flow is subsonic at the outer boundary of the Eulerian solver. Indeed, the VPM solver is not fitted with a shock-capturing methodology. The hybrid solver should however be able to treat transonic flows as long as the transonic region is entirely contained in the Eulerian subregion. The case of a supersonic flow everywhere in the computational domain is not of interest in this work. In Chapter 5, the coupling methodology will be assessed on a series of test cases of two-dimensional viscous compressible flows in the subsonic regime.

CHAPTER 5

VALIDATION OF THE COUPLING METHODOLOGY

In this Chapter, the coupling methodology will be applied to a series of test cases. In the following, co-simulations in which the Lagrangian solution is not corrected but is used to provide the boundary conditions to the Eulerian solver will be referred to as a "one-way coupling". The VPM *solver will then also be said to run in a "standalone" mode. On the contrary, when the full coupling methodology is employed, the co-simulation will be referred to as a "two-way coupling" and the solver itself as the "DG-VPM solver".*

5.1 Introduction

For all simulations, the second order 3BDF implicit scheme will be used in the Eulerian solver. The fourth-order Runge-Kutta scheme used in the VPM solver of Chapter 2 and 3 is here also used, even though a second order scheme, i.e. the same order as in the Eulerian solver, would probably suffice. The time step will be chosen such as to satisfy the stability criteria of the VPM solver and to achieve an acceptable accuracy. In that regard, the relaxed stability constraints of VPM solvers for incompressible flows are retightened for the VPM solver of this work. Although the advection is still carried out in a Lagrangian

fashion, a CFL-like constraint is, however, applicable due to the acoustics. The same time step was used in both solvers. The co-simulation scenario is thus the one depicted in Fig. 1.7. The boundary conditions are thus only computed at the end of the VPM time step. The quadrature on the Eulerian elements of Section 4.4.2 will be used to correct the thermodynamic quantities while contour integrals on the VPM cells will be used for the correction of the vorticity and the dilatation as described in Section 4.4.1.

The Paraview-based visualization of the high-order polynomial solutions of the Eulerian solver is here allowed by a plug-in developed at Cenaero [133]. The plug-in has been integrated into Paraview and is available online[1]. This allowed to visualize the solution of both solvers using the same tool. In the Eulerian region, the relevant solution is the solution computed by the Eulerian solver; the Lagrangian solver is purposely under-resolved there. Unless otherwise stated, the Eulerian solution will thus be shown there.

The chapter is organized as follows. The simulation of a co-rotating vortex pair that was performed in Chapter 2 using the VPM solver alone will here be performed using the DG-VPM solver. The results will be compared to reference results from the literature in Section 5.2. The second test case, in Section 5.3, will involve a solid obstacle; the benchmark test case of the scattering of sound by a cylinder of Chapter 3 is here once more considered. The extent of the Eulerian subdomain will be varied, and the results will be compared to the analytical solution. Next, the flow past a cylinder will be simulated using the DG-VPM solver in Section 5.4 and Section 5.5. Reference solutions using the DG solver only will also be obtained in both Sections. In Section 5.4, the Reynolds and freestream Mach number will be such that $Re = 100$ and $M = 0.1$, respectively. The results will be compared to the reference results of Canuto and Taira [20]. In Section 5.5, the results obtained at early times ($(t\, U_\infty)/D \leq 5.0$) will be compared to reference results. As such results were not provided in [20] for the flow at $Re = 100$, the Reynolds number will be increased to $Re = 550$, which is an extensively documented test case. The results will be compared to reference results using a state-of-the-art VPM solver for incompressible flows [58]. In light of these simulations, general considerations regarding the coupling methodology will be given in Section 5.6.

[1] https://github.com/Kitware/ParaView/tree/master/Plugins/GmshReader)

5.2 Co-rotating vortex pair

The simulation of the co-rotating vortex pair performed using the VPM solver alone in Chapter 2 is here performed again using the Eulerian solver to capture the flow dynamics in the center region of the domain. The situation is depicted in Figure 5.1. Three distinct simulations were considered:

1. a one-way coupling co-simulation; the VPM solver is capable of accurately simulating the flow physics in a standalone mode as demonstrated in Chapter 2. The Eulerian mesh is a Cartesian grid for comparison purposes with the next case. This simulation will be referred to as the 1W-s case, where "s" stands for the structured grid of the Eulerian solver.

2. a two-way coupling using the same Cartesian grid in the Eulerian solver; this simulation will thus be referred to as the 2W-s case. As explained in Section 4.4.2, the nearest-neighbor projection of the correction computed on the Eulerian elements requires a non-conservative renormalization. In order to eliminate this error, the Cartesian grid in the Eulerian solver is constructed so that the recursively subdivided mesh coincides with the Lagrangian grid.

3. the most general situation is the two-way coupling using an unstructured grid for the Eulerian solver; the simulation will be referred to as the 2W-u case, where "u" stands for unstructured. The number of degrees of freedom between 2W-s and 2W-u will, however, not be preserved.

In this acoustic test case, the quantities were made dimensionless using the same acoustic scaling as in Chapter 2. The goal of these simulations is to obtain the temporal evolution of the dilatation field at $(x, y) = (0, 1.2R)$, with R the initial half-separation distance between the vortices, and to compare the results with reference results from the literature. The Lagrangian and Eulerian computational domains were chosen as square domains of length $L_L = 9.0R$ and $L_E = L_L/2$, respectively. Their centers coincide with the center of the vortical system. The number of grid points in the Lagrangian solver was chosen as $N_x = N_y = 384$, thus leading to $N_c = 2\,r_0/R = 12.8$ Lagrangian grid points in the vortex core diameter. A similar resolution was shown to provide accurate results in Chapter 2. The radiation boundary condition, Equation (2.21), was imposed in an annular region surrounding the interior domain. For the 1W-s and 2W-s cases, the Cartesian grid spacing in the Eulerian solver, h_E, was chosen such that $h_E = ph_L$, with p the polynomial order and h_L the Lagrangian grid spacing; coincident meshes for the correction step are thus

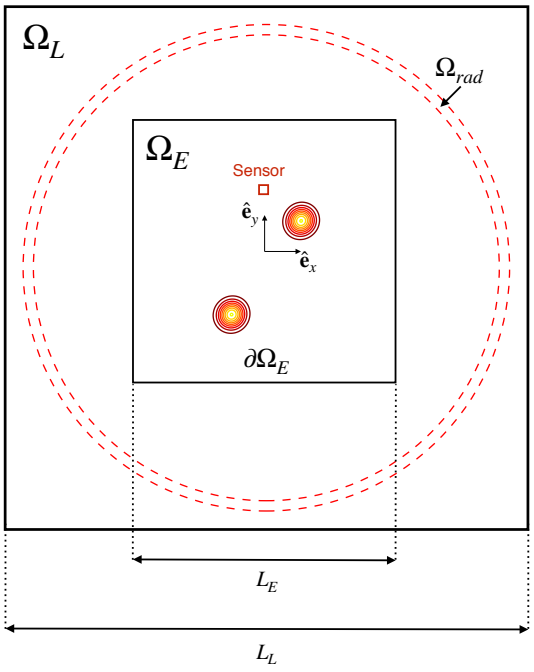


Figure 5.1: Co-rotating vortex pair: numerical setup. The numerical setup for the Lagrangian solver is the same as in Chapter 2. The dilatation field will be monitored at $(x, y) = (0, 1.2R)$ in the Eulerian solver and in the Lagrangian solver as indicated by the red square. A vorticity field obtained using the VPM solver alone is here shown for illustration purpose.

obtained when recursively subdividing the Eulerian element using the minimal recursion level $l_e = l_e^{min}$ provided by Eq. (4.33). For the 2W-u case, the unstructured mesh was constituted of 2997 quadrangles. As stated before, the number of dofs is thus not preserved; the unstructured mesh is not a perturbation of the cartesian grid used in the 1W-s and 2W-s cases. In all cases, the mesh requirement of Eq. (4.23) was respected and fourth order shape functions were used. The recursion level of the Eulerian elements were chosen as $l_e = l_e^{min} = 2$ and $l_e = 3$ for the 2W-s and the 2W-u cases, respectively. In both cases, the VPM cells were recursively subdivided twice to compute the contour integrals. The initialization presented in Section 2.3.2 allows to reduce the acoustic transient at the onset of the simulation and requires to solve Poisson problems (for the velocity field and the enthalpy field). The initialization of the Eulerian solver was here performed using the Lagrangian solver solution; the initialization was computed in the Lagrangian solver and then interpolated on the Eulerian mesh. A finer Lagrangian grid was used to en-

sure a good quality interpolation.

The results of the 1W-s case are given in Fig. 5.2. The dilatation is monitored both in the Eulerian solver and in the Lagrangian solver. Of course, the solution of the Lagrangian solver is similar to that obtained in Chapter 2 (see Figure 2.8). The solution obtained in the Eulerian solver is in excellent quantitative agreement with the WENO solver of Zhang et al. [164]. It is, furthermore, in better agreement than the Lagrangian solution.

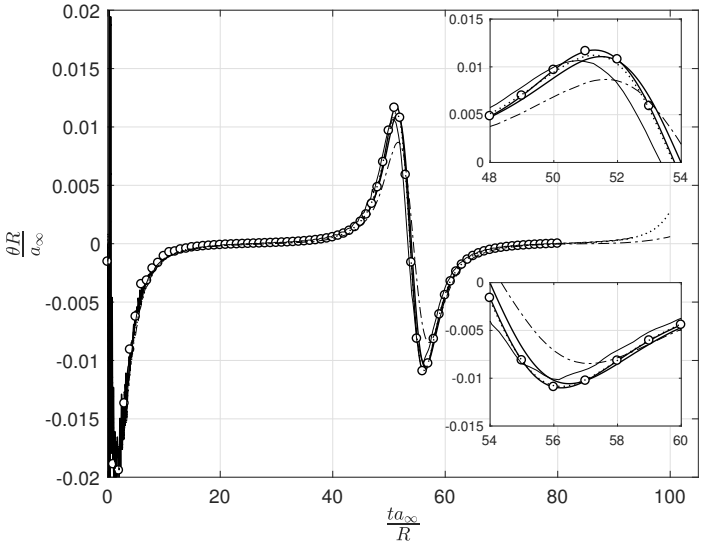


Figure 5.2: Co-rotating vortex pair (1W-s): dilatation monitored at $(x, y) = (0, 1.2R)$ in the Lagrangian solver (—) and in the Eulerian solver (-o-). The results from Eldredge et al. with $N_c = 13$ (—) and $N_c = 40$ (- -) and from Zhang et al. (·····) are also reproduced from [50] and [164]. The inset in the upper right corner is a zoom on the maximum value achieved during the first rotation of the vortical system. The lower right inset is a zoom on the minimum.

The results obtained in the 2W-s are compared with the results of the 1W-s case and with the reference results in Fig. 5.3, which is a zoom around the time corresponding to a half-rotation of the vortex pair. The values obtained at the extrema are not impacted by the correction of the Lagrangian solution whereas a slight modification of the monitored dilatation can be observed between these moments; the Eulerian solution exhibits an offset with respect to the results of Zhang et al. and from those of the one-way coupling. Although these changes are only slight, they illustrate the feedback that can ex-

ist between the correction of the Lagrangian solution and the imposition of the BCs for the Eulerian solver. Figure 5.4 shows the results obtained using the full coupling methodology with an unstructured grid. Despite the needed non-conservative renormalization of the correction provided to the Lagrangian solver, the results are also found to be in excellent agreement with the results from the literature.

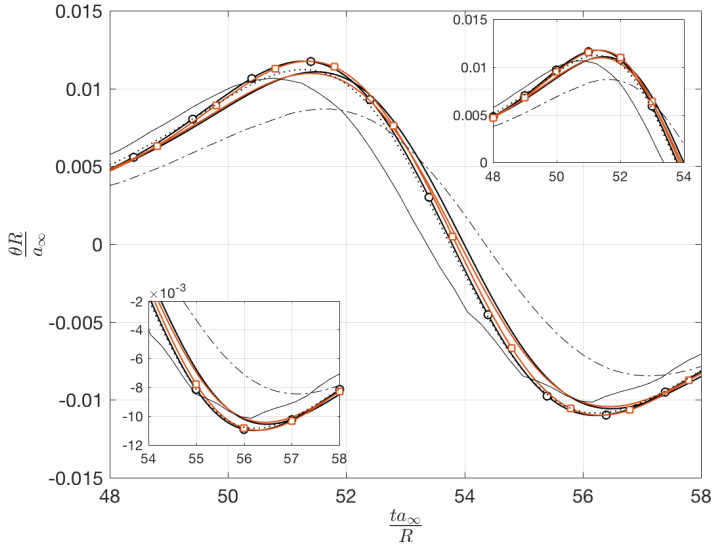


Figure 5.3: Co-rotating vortex pair (1W-s and 2W-s cases): dilatation monitored at $(x, y) = (0, 1.2R)$ in the Lagrangian solver (1W-s :—, 2W-s: —) and in the Eulerian solver (1W-s: —o—, 2W-s: —□—). The results from Eldredge et al. [50] with $N_c = 13$ (—) and $N_c = 40$ (- - -) and from Zhang et al. (·····) [164] are also reproduced. The inset in the upper right corner is a zoom on the maximum value achieved during the first rotation of the vortical system. The lower left inset is a zoom on the minimum.

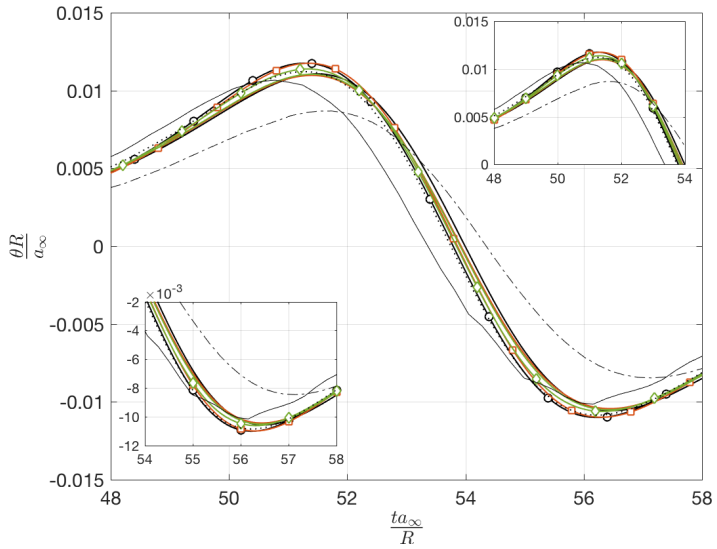


Figure 5.4: Co-rotating vortex pair (1W-s, 2W-s and 2W-u cases): dilatation monitored at $(x, y) = (0, 1.2R)$ in the Lagrangian solver (1W-s :—, 2W-s: —, 2W-u :—) and in the Eulerian solver (1W-s: —○—, 2W-s: —□—, 2W-u: —◇— . The results from Eldredge et al. with $N_c = 13$ (—) and $N_c = 40$ (- · -) and from Zhang et al. (·····) are also reproduced from [50] and [164] . The inset in the upper right corner is a zoom on the maximum value achieved during the first rotation of the vortical system. The lower left inset is a zoom on the minimum.

5.3 2-D cylindrical acoustic benchmark problem

The first simulation involving a solid obstacle is the benchmark test case of the scattering of sound of a cylinder, which was already considered in Chapter 3. The same computational domain was here used; the remark regarding the necessity to account for the whole dilatation field to obtain accurate results still holds. The penalization will here be treated using the explicit integration scheme. All the parameters are identical to those provided in Chapter 3 except for the grid resolution; the resolution of the Lagrangian grid was here halved to show the improvements offered by the use of the near-body Eulerian solver. As depicted in Figure 5.5, two different Eulerian subdomains will be considered:

1. the Eulerian subdomain surrounds the observation points (see Fig. 5.5(a)). It will thus be possible to monitor and compare the time history of the acoustic fluctuations at these locations in both solvers. The one-way and two-way coupling co-simulations using this configuration will be

referred to as 1W-w and 2W-w, with w standing for "wide". The results are provided in Section 5.3.1.

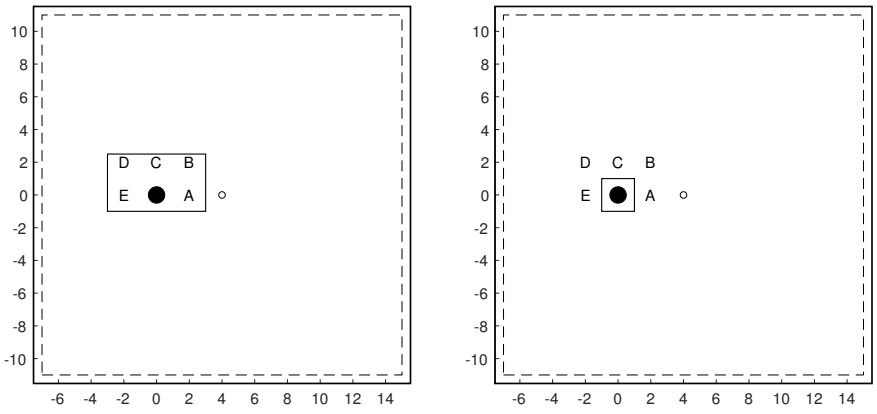
2. the second co-simulation moves closer to the intended use of the coupling methodology; a narrow Eulerian subdomain surrounds the cylinder (see Fig. 5.5(b)). The two-way coupling co-simulation will be performed in Section 5.3.2 and will be referred to as 2W-n, with n standing for "narrow". The time history of this co-simulation will be compared to the results obtained with the Lagrangian solver alone[2].

In both cases, the Eulerian subdomain does not surround the acoustic pulse at time $t = 0$. The proper imposition of the boundary conditions to the Eulerian solver will thus be crucial to obtain good quality results in the Eulerian solver. The capability of the DG solver to use unstructured curved elements is here taken advantage of and third order interpolation functions are used in all cases. The initialization was performed in each solver using the analytical initial condition. The same time step was used in both solvers. The recursion level on the Eulerian elements was chosen using the automatic detection described in Section 4.4.2 and Eq. (4.36). The Lagrangian elements were not recursively subdivided to compute the correction for the dilatation and the circulation.

5.3.1 Wide Eulerian subdomain

The Eulerian subdomain is such that $\Omega_E = [-3.0, 3.0] \times [-1.0, 2.5]$ and is discretized using 1102 curved triangles and third order interpolation functions. The resolution at the wall is such that 32 curved elements cover the perimeter of the cylinder. As stated above, when compared to Chapter 3, the resolution used in the Lagrangian solver was here halved to show the improvements brought by the near-body Eulerian solver but was still sufficiently fine to isolate the error due to the body treatment; the resolution was such that 20 particles laid across the half-width diameter of the initial pressure disturbance. Figure 5.6 shows close-up snapshots of the pressure field for the wide configuration, both for the one-way coupling, 1W-w, and the two-way coupling, 2W-w, co-simulations. As already stated in the introduction of this Section, the field shown in the Eulerian subdomain is the solution computed by the Eulerian solver. The acoustic pulse smoothly propagates into the Eulerian subdomain

[2]When the VPM solver is used in a standalone mode (one-way coupling), the Lagrangian solver is evidently independent of the Eulerian solver, and hence also of the configuration considered in the Eulerian subdomain. The results from the Lagrangian solver obtained from the previous one-way coupling co-simulation, 1W-w, may thus be used in the second case as well.

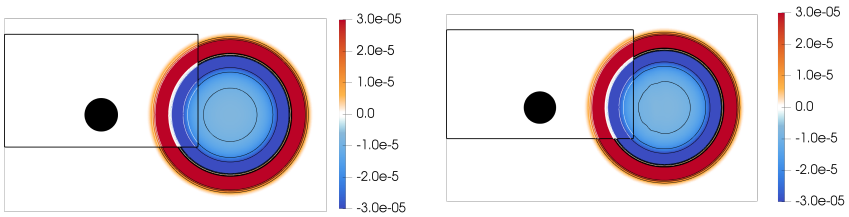


(a) Wide Eulerian subdomain (b) Narrow Eulerian subdomain

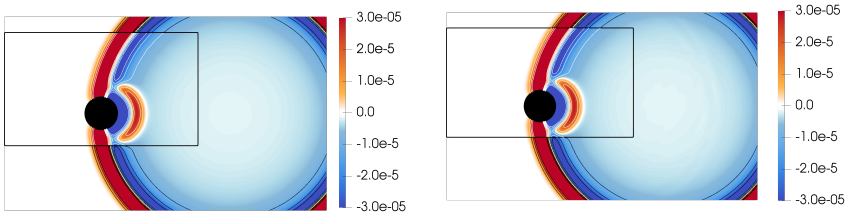
Figure 5.5: Acoustic scattering of a cylinder: sketch of the computational domains and observation points lettered from A to E. The cylinder and the initial location of the acoustic pulse are depicted as the filled black and open black circles, respectively. The dashed line marks the beginning of the zone where the radiation condition is applied.

at time $ta_\infty/D \simeq 2$. The same observation can be made for the reflections leaving the Eulerian subdomain, at times $ta_\infty/D \simeq 6$ and $ta_\infty/D \simeq 8$. The absence of correction in the 1W-coupling co-simulation causes a slight offset between the contours of the Eulerian and Lagrangian solutions at the boundaries of the Eulerian solver, as best seen in the zoom in Fig. 5.7(a)). On the contrary, the contours of the 2W-coupling co-simulation are continuous across the boundaries of the Eulerian subdomain at all times (see also the zoom in Figure 5.7(b)).

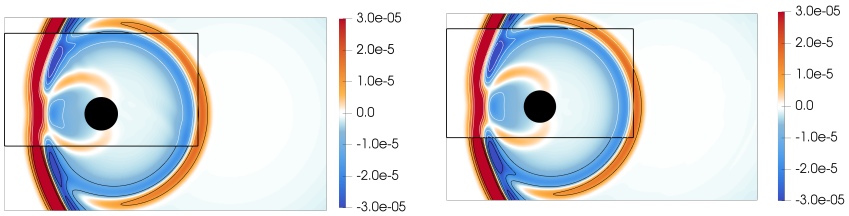
The improvements brought by the near-body Eulerian solver can also be observed on the time history of the pressure fluctuations as monitored at the five locations depicted in Fig. 5.5(a). Let us first consider the one-way coupling co-simulation whose results are depicted in Figs. 5.8-5.10. Using this resolution, the Lagrangian solver is capable of capturing the primary wave up to the sensor A as shown in Fig. 5.8(a). Sensor B, C and D exhibit a slight amplitude error in their representation of the primary wave (the first oscillation in Figs. 5.8(b), 5.9(a) and 5.9(b)). The underresolution is even more acute when considering the secondary and tertiary waves, as can be best observed in Fig. 5.10(a) where both amplitude and phase errors are exhibited. The resolution indeed prevents the penalization to correctly account for the solid obstacle. The slight discrepancies that were observed at the boundary of the



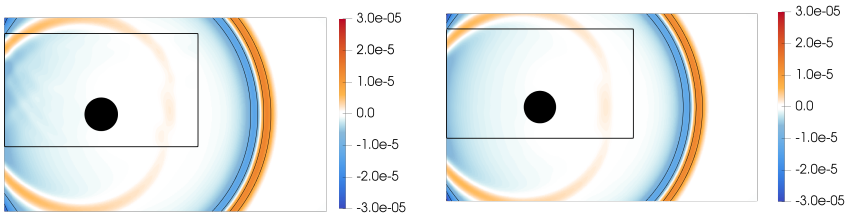
(a) $\frac{t\,a_\infty}{D} = 2$



(b) $\frac{t\,a_\infty}{D} = 4$

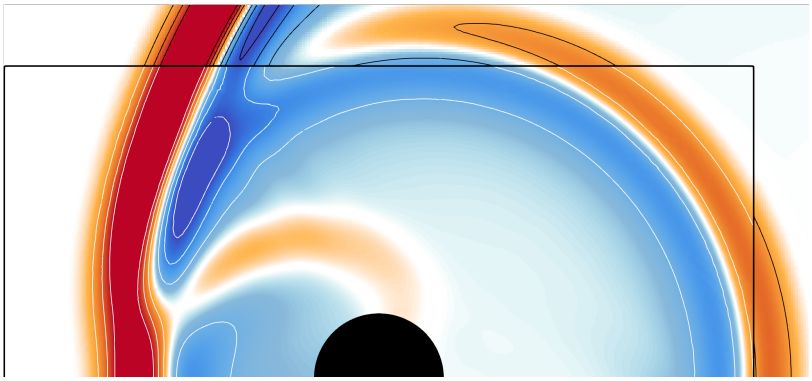


(c) $\frac{t\,a_\infty}{D} = 6$

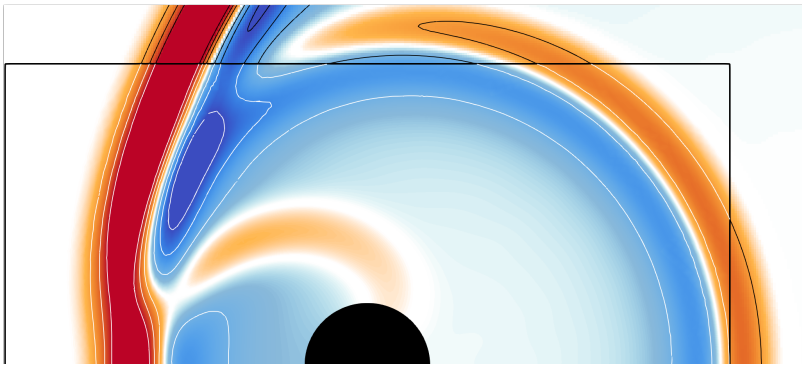


(d) $\frac{t\,a_\infty}{D} = 8$

Figure 5.6: Acoustic scattering of a cylinder (wide Eulerian subdomain): Pressure perturbations $(p-p_\infty)/(\rho_\infty a_\infty^2)$ in a close-up view around the cylinder (1W-w: left, 2W-w: right). The solution shown in the near-body subdomain is the Eulerian solution. Pressure perturbation contours from the Eulerian solver (white) and from the Lagrangian solver (black, not shown in the near-body subdomain) are in the range $[-3.0\cdot10^{-5}, 3.0\cdot10^{-5}]$ in increments of $1.0\cdot10^{-5}$. The zero-level contour is not shown.



(a) One-way coupling co-simulation (1W-w)



(b) Two-way coupling co-simulation (2W-w)

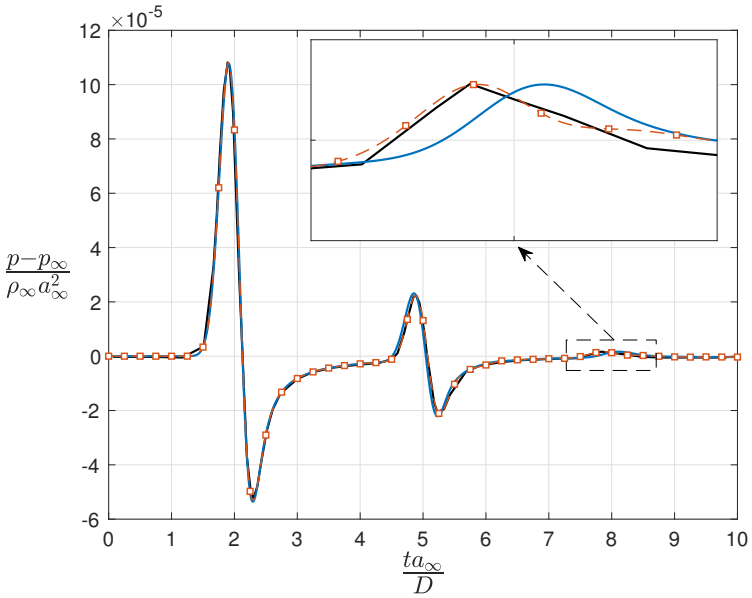
Figure 5.7: Acoustic scattering of a cylinder: Zoom on the upper half part of the Eulerian subdomain at time $ta_\infty/D = 6$. Contrary to the contours of the one-way coupling co-simulation (a), the contours of the two-way coupling co-simulation (b) are continuous across the boundaries of the Eulerian subdomain.

Eulerian subdomain do not impair the quality of the results monitored in the Eulerian solver; although starting from an initial quiescent state, the Eulerian solution is indeed in almost perfect accordance with the analytical solution. The boundary conditions were thus properly provided to the Eulerian solver.

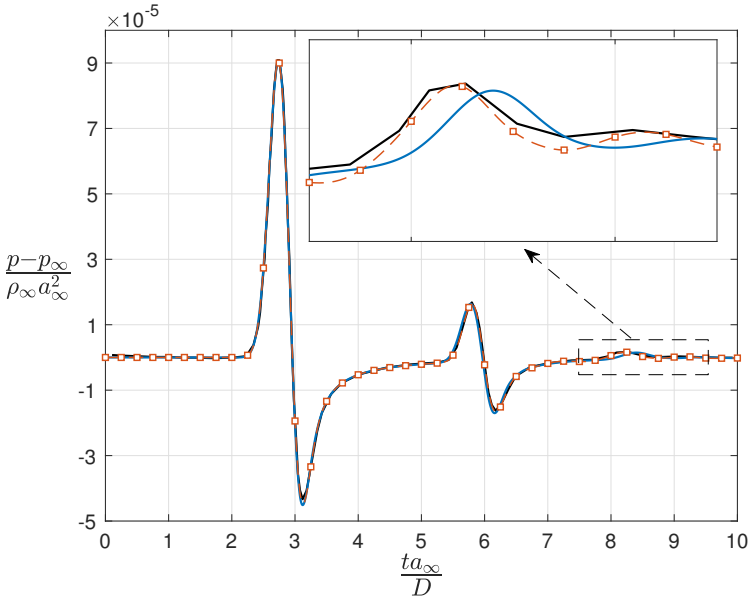
The corresponding time histories for the 2W-coupling co-simulation are depicted in Figs. 5.11-5.13. The Lagrangian solution of the 1W-coupling is also reproduced for comparison purpose. Due to the treatment of the near-body region by the Eulerian solver, both the Eulerian and the Lagrangian solutions are in accordance with the analytical solution. Even the tertiary wave is here satisfactorily accounted for. As shown in Fig. 5.13, the pressure fluctuation at the location of Sensor E, i.e the sensor behind the cylinder, exhibits the most significant improvement.

5.3.2 Narrow Eulerian subdomain

The Eulerian subdomain is here chosen in accordance with the intended use of the coupling methodology; it tightly surrounds the cylinder. The Eulerian subdomain is such that $\Omega_E = [-1.0, 1.0] \times [-1.0, 1.0]$ and is discretized using 258 elements with third order interpolation functions. The same resolution at the wall as in the previous case was adopted. Figures 5.14-5.16 show the time history of the pressure perturbations for this configuration. The results are compared with those obtained when using the VPM solver alone (i.e. the results from the co-simulation $\mathrm{1W\text{-}w}$). Here again, the body-fitted mesh and the high-order capability of the near-body solver significantly improve the quality of the solution at the monitoring points. Of course, as the correction region is here narrower than in the first configuration, the improvements brought to the Lagrangian solution are smaller. Remarkably enough, the solution using the coupling methodology is here of a quality similar to the solution obtained in Chapter 3 (see Figs. 3.5-3.7) using a Lagrangian grid spacing twice smaller than the one used here. Figure 5.17 shows close-up snapshots of the pressure field and a zoom on the Eulerian subdomain at time $(ta_\infty)/D = 4$ is shown in Fig. 5.18.

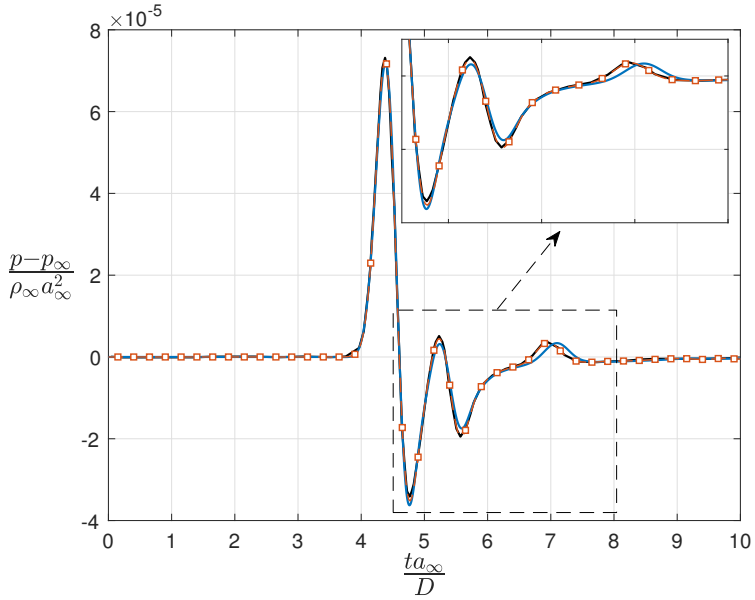


(a) Sensor A at (2,0)

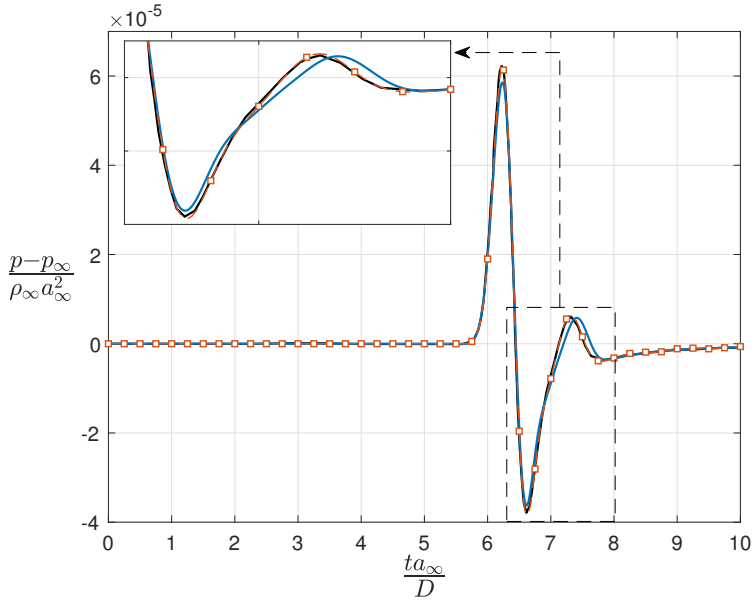


(b) Sensor B at (2,2)

Figure 5.8: Acoustic scattering of a cylinder (1W-w): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five different locations (see Fig. 5.5(a)). Analytical solution (—), Lagrangian solver (—), Eulerian solver (-□-). Zooms on relevant time windows are provided in the insets.

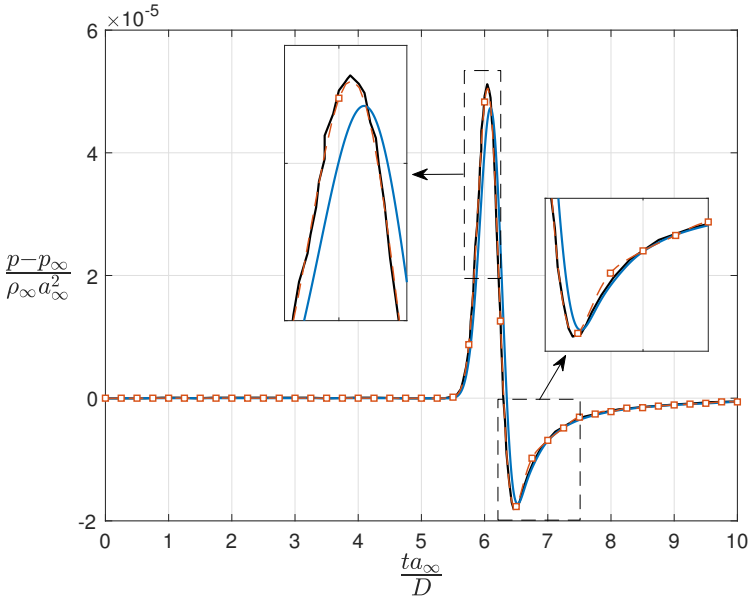


(a) Sensor C at (0,2)



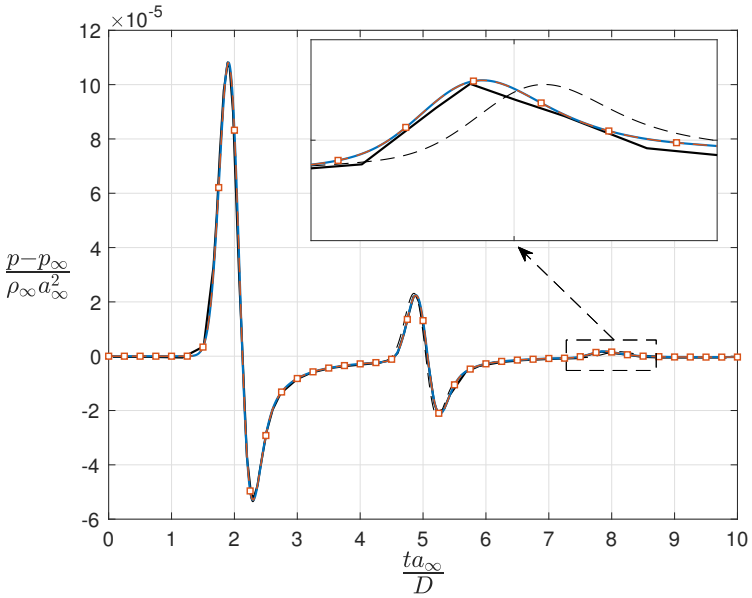
(b) Sensor D at (-2,2)

Figure 5.9: Acoustic scattering of a cylinder (1W-w): continued

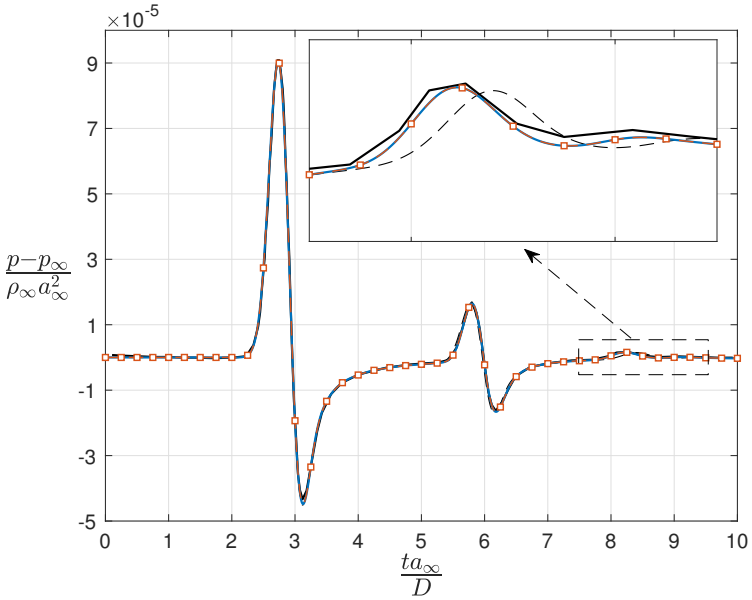


(a) Sensor E at (-2,0)

Figure 5.10: Acoustic scattering of a cylinder (1W-w): continued

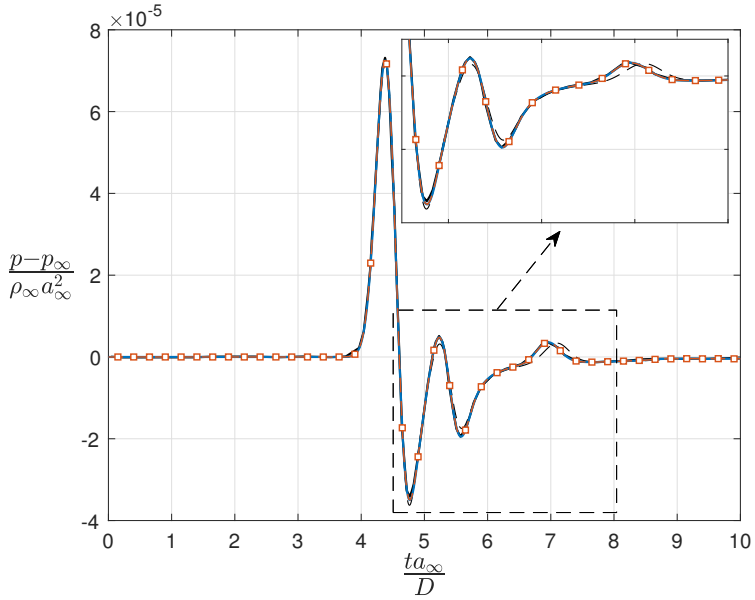


(a) Sensor A at (2,0)

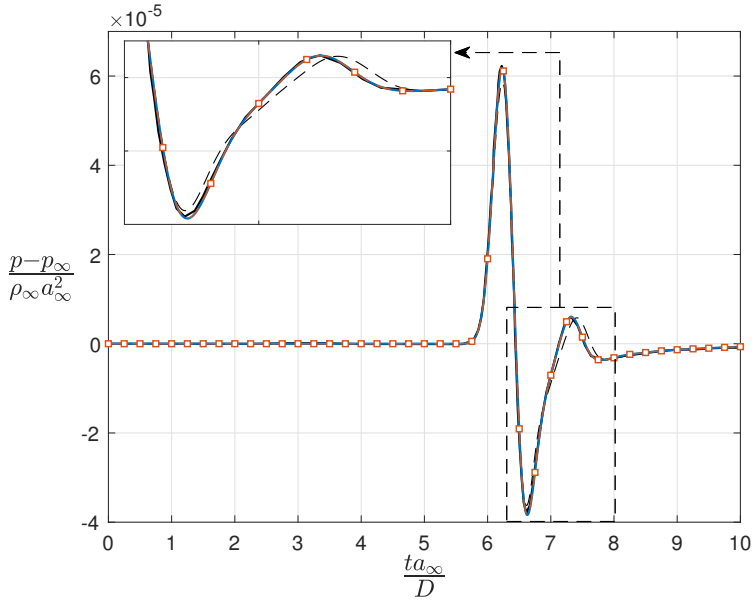


(b) Sensor B at (2,2)

Figure 5.11: Acoustic scattering of a cylinder (2W-w): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five different locations (see Fig. 5.5(a)). Analytical solution (—), Lagrangian solver (1W-w: - -, 2W-w: —), Eulerian solver ( ). Zooms on relevant time windows are provided in the insets.

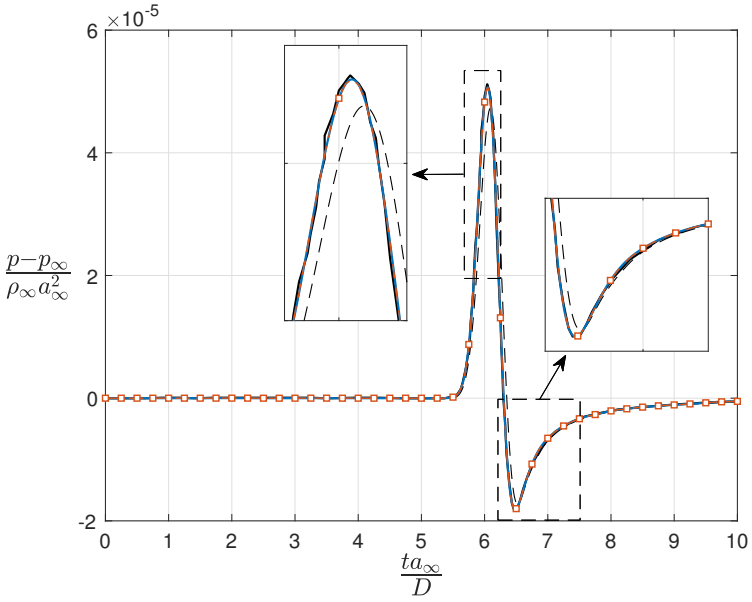


(a) Sensor C at (0,2)



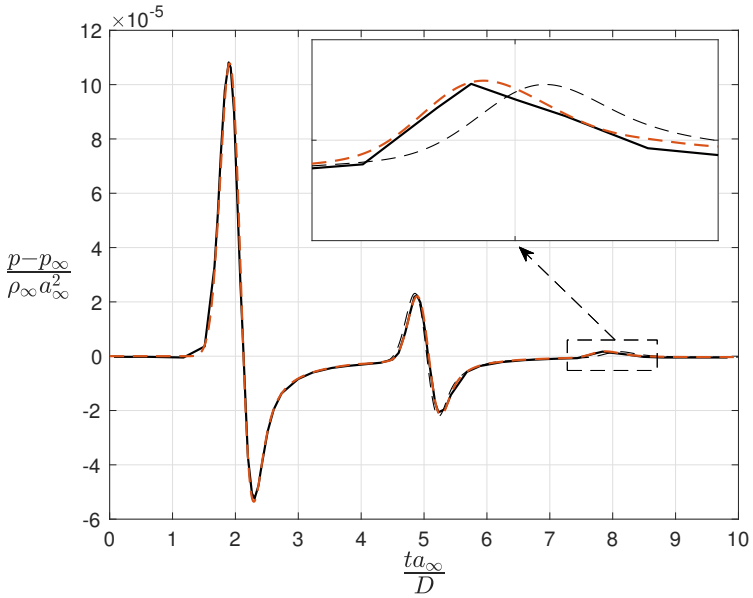
(b) Sensor D at (-2,2)

Figure 5.12: Acoustic scattering of a cylinder (2W-w): continued

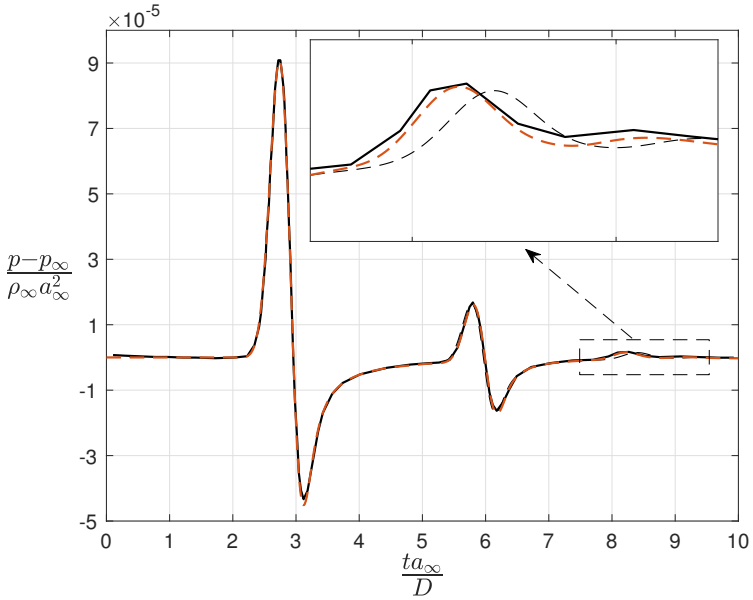


(a) Sensor E at (-2,0)

Figure 5.13: Acoustic scattering of a cylinder (2W-w): continued

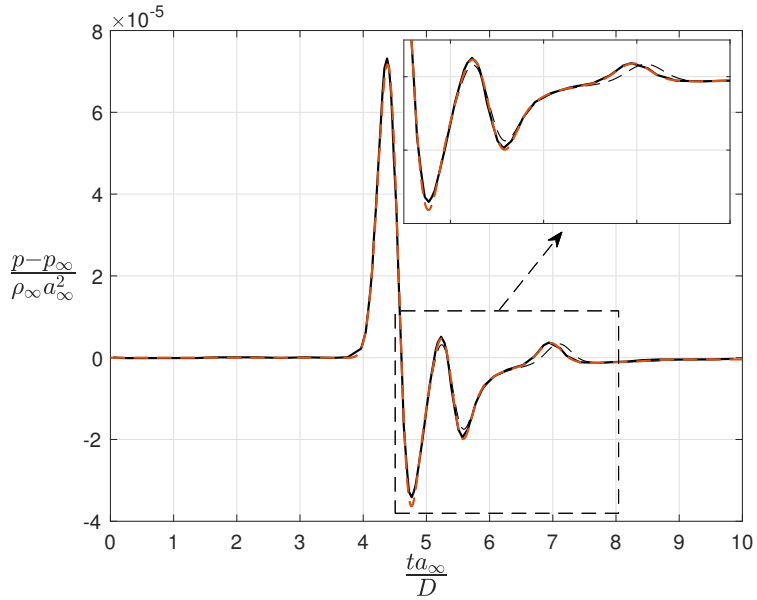


(a) Sensor A at (2,0)

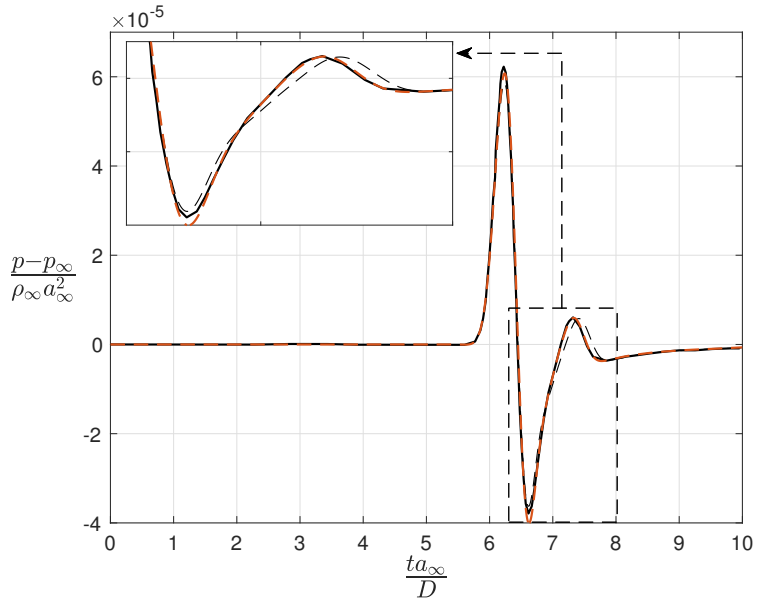


(b) Sensor B at (2,2)

Figure 5.14: Acoustic scattering of a cylinder (2W-n): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five different locations (see Fig. 5.5(b)). Analytical solution (—), Lagrangian solver (1W: - -, 2W-n: - -). Zooms on relevant time windows are provided in the insets.

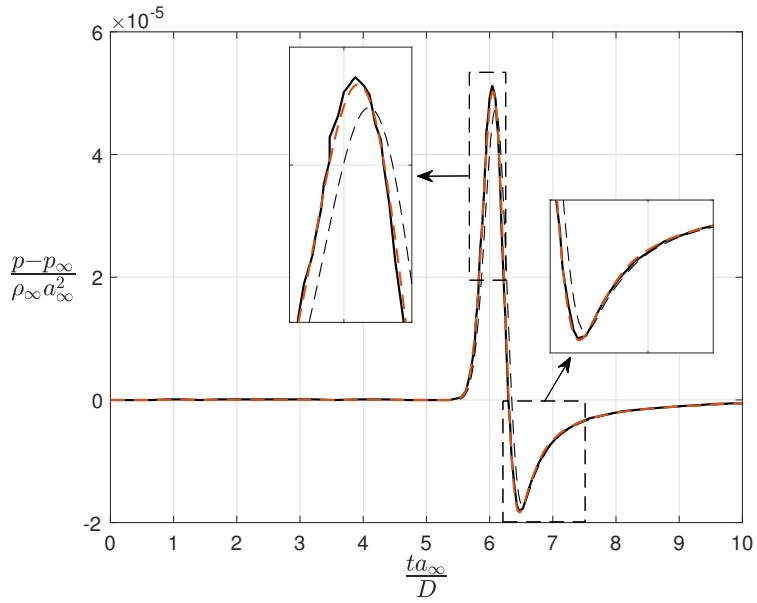


(a) Sensor C at (0,2)



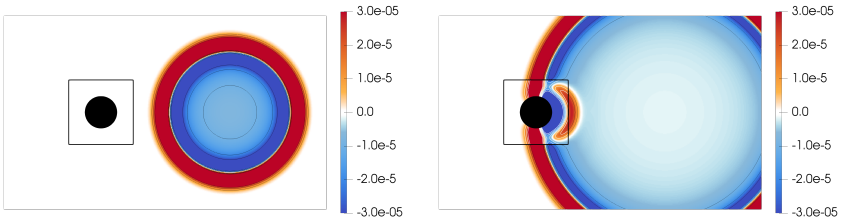
(b) Sensor D at (-2,2)

Figure 5.15: Acoustic scattering of a cylinder (2W-n, continued): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five different locations (see Fig. 5.5(b)). Analytical solution (—), Lagrangian solver (1W: - -, 2W-n: - -). Zooms on relevant time windows are provided in the insets.

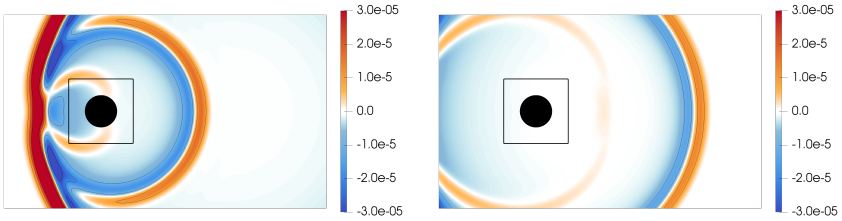


(a) Sensor E at (-2,0)

Figure 5.16: Acoustic scattering of a cylinder (2W-n, continued): time history of the pressure perturbation $\frac{p-p_\infty}{\rho_\infty a_\infty^2}$ at five different locations (see Fig. 5.5(b)). Analytical solution (—), Lagrangian solver (1W: - -, 2W-n: - -). Zooms on relevant time windows are provided in the insets.



(a) $\frac{t\,a_\infty}{D} = 2$ (b) $\frac{t\,a_\infty}{D} = 4$



(c) $\frac{t\,a_\infty}{D} = 6$ (d) $\frac{t\,a_\infty}{D} = 8$

Figure 5.17: Acoustic scattering of a cylinder (2W-n): Pressure perturbations $(p - p_\infty)/(\rho_\infty a_\infty^2)$ in a close-up view around the cylinder. The solution shown in the near-body subdomain is the Eulerian solution. Pressure perturbation contours from the Eulerian solver (white) and from the Lagrangian solver (black, not shown in the near-body subdomain) are in the range $[-3.0 \cdot 10^{-5}, 3.0 \cdot 10^{-5}]$ in increments of $1.0 \cdot 10^{-5}$. The zero-level contour is not shown.

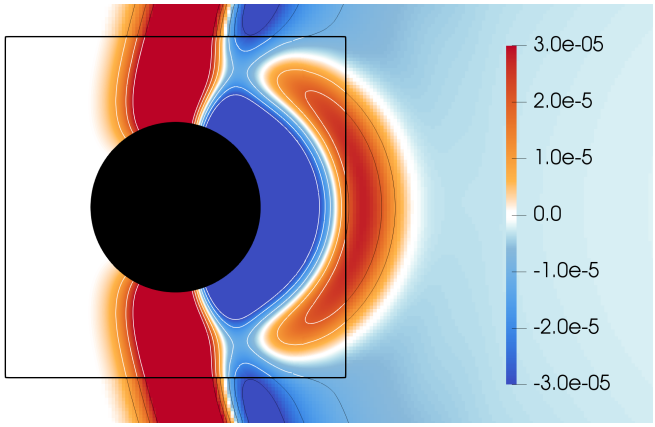


Figure 5.18: Acoustic scattering of a cylinder (2W-n): Zoom on the Eulerian subdomain at time $ta_\infty/D = 4$. The contours are continuous across the boundaries of the Eulerian subdomain.

5.4 Flow past a cylinder at Re = 100 and M = 0.1

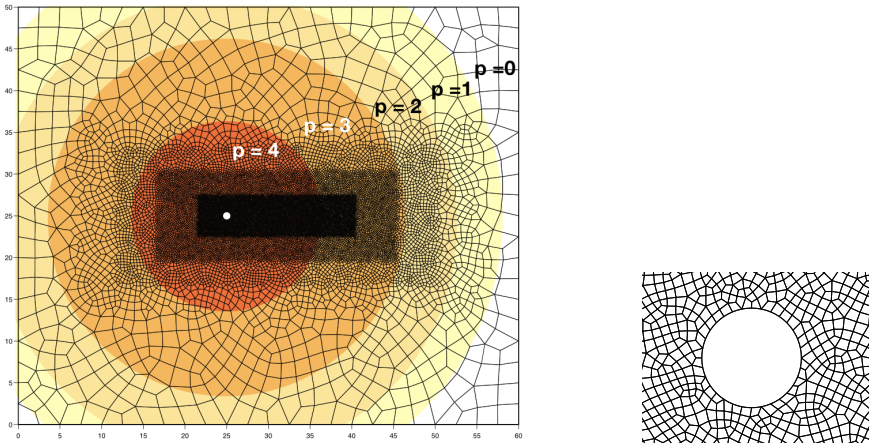
Canuto and Taira [20] performed the DNS of two-dimensional viscous and shock-free compressible flow past a circular cylinder at several Reynolds and Mach numbers. The purpose of the paper was to study the compressibility effects on the characteristic parameters of such a flow near the onset of two-dimensional shedding. The Reynolds number, $Re = (\rho_\infty U_\infty D)/\mu$, was in the range $[20, 100]$ and the free-stream Mach number, $M = U_\infty/a_\infty$, in the range $[0, 0.5]$. A second order accurate Finite Volume solver with a third order Runge-Kutta time integration scheme was used. The computational domain, grid resolution and domain size, which were further used for all simulations, were chosen using a grid refinement study, by varying the domain size and by comparing the results with reference results from the literature. One of the interesting contributions of [20] includes tabulated reference values for the characteristic parameters as a function of the Reynolds and Mach numbers.

The purpose of this Section is to perform the simulation of the flow past a circular cylinder at $Re = 100$ and $M = 0.1$ using the coupling methodology introduced in Chapter 4. The results will be compared to the reference results obtained by Canuto and Taira [20]. A simulation using the Eulerian solver alone was first performed to provide reference results, which are presented in the next Section.

5.4.1 Simulation using the DG solver alone

The computational domain used by Canuto and Taira [20] extended over $\Omega = [-20, 40] \times [-20, 20]$, the center of the cylinder being placed at the origin of the reference frame. Freestream boundary conditions were imposed at the inlet of the computational domain while the sponge zone methodology proposed by Freund [53] was used to damp the exiting flow structures at the outflow. The width of the sponge zone was chosen as $l_{sp} = 10$. In addition to the additional damping term as in Section 4.5.1, the methodology proposed by Freund also uses an artificial advection velocity in the sponge zone. Symmetry boundary conditions were used at the upper and lower boundaries of the computational domain. Two different numerical setups are here considered; the first computational domain is defined as $\Omega_1 = [-25, 25] \times [-25, 25]$ while the second is closer to the configuration of Canuto and Taira [20], $\Omega_2 = [-25, 35] \times [-25, 25]$. The meshes used to discretize the computational domains both make use of curved quadrilateral elements. Freestream boundary conditions are applied at all the boundaries. The first simulation used second order interpolation func-

tions in the whole domain and 9097 elements to discretize the computational domain. The resolution at the wall was such that 64 elements covered the perimeter of the cylinder. In the second simulation, the mesh resolution close to the wall was reduced to 32 elements on the perimeter of the cylinder, which was compensated by the use of higher order interpolation functions as depicted in Fig. 5.19. A smoother mesh resolution variation was also achieved. The second numerical setup is thus seen to be of better quality. Although a sponge layer methodology is available in Argo, the second simulation relies on hp-adaptivity to damp the exiting flow structures; the size of the elements and the interpolation order are progressively degraded, thus relying on the two possibilities allowing to increase the dissipation error (see Fig. 4.4). This is depicted in Fig. 5.19. The first simulation relied on h-adaptivity only. A total of 23100 quadrangles were used in the second numerical setup. It is important to stress that the resolutions used in these two simulations are here clearly exaggerated in order to generate reference results.



(a) (b)

Figure 5.19: Second numerical setup: (a) the size of the quadrilateral elements is progressively increased to damp the exiting flow structures. To the same end, the interpolation order p decreases with the distance to the center of the cylinder. As can be seen on the zoom on the cylinder (b), curved elements are used to discretize the circular cylinder.

The second order 3BDF implicit scheme was used with a fixed time step, $(\Delta t U_\infty)/D = 6.865 \cdot 10^{-3}$. A uniform flow corresponding to the freestream conditions was used as the initial condition. The parallelization capability of Argo was here taken advantage of for the second simulation; the simulation

was performed on the Zenobe cluster using 24 cores, thus dividing the mesh in approximately a thousand elements per core. Snapshots of the vorticity field, $(\omega D)/U_\infty$, at different times near the onset of the instability are shown in Fig. 5.20. Similar results obtained using a VPM solver for incompressible flows are available in [102]. The impoverished zone where first order interpolation functions and coarsened elements are used (see Fig. 5.19(a)) is also partially shown in Fig. 5.20.

The time histories of the lift and drag coefficients are shown in Fig. 5.21 for both simulations. The vortex shedding is here triggered by the interaction of exiting vortex structures with the outflow boundary; the onset of the instability thus occurs later in the second simulation, which uses a longer computational domain. Table 5.1 lists the time-averaged drag coefficient, $\overline{C}_D$, the amplitude of lift and drag fluctuations, ΔC_D and ΔC_L as well as the Strouhal number St of the simulations together with the reference results from the literature. They are found to be in good agreement. The results from the second simulation will thus be used as the reference results for the co-simulation presented in the next Section.

	$\overline{C}_D$	ΔC_D	ΔC_L	St
Canuto & Taira [20]	1.35	0.0096	0.328	0.166
DG solver (1^{st} setup)	1.347	0.0094	0.326	0.167
DG solver (2^{nd} setup)	1.346	0.0094	0.328	0.167

Table 5.1: Flow past a cylinder: Time-averaged drag coefficient $\overline{C}_D$, amplitude of drag fluctuations ΔC_D and of lift fluctuations ΔC_L and Strouhal number St. Canuto and Taira [20] did not provide more digits for $\overline{C}_D$.

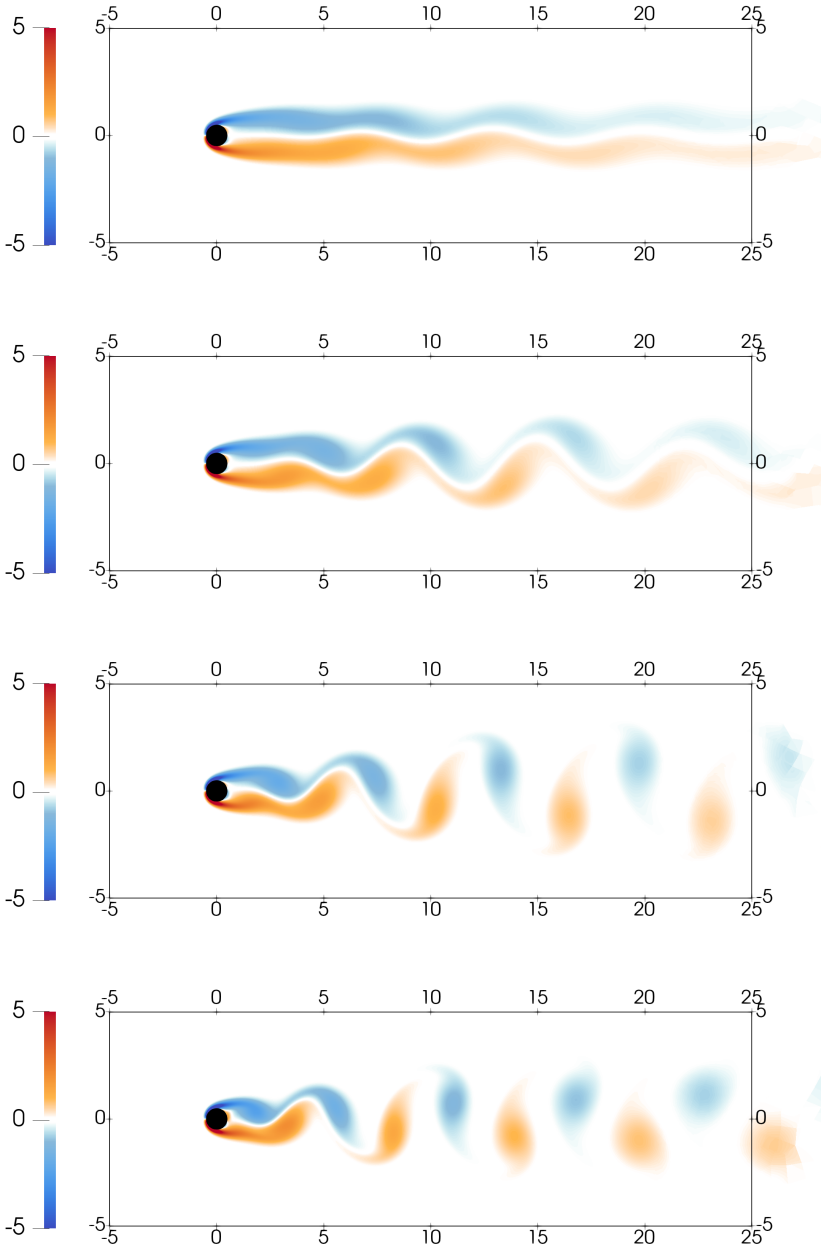
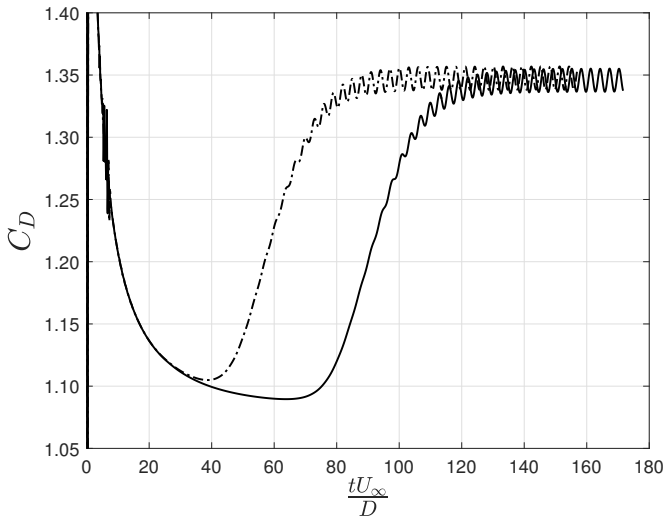
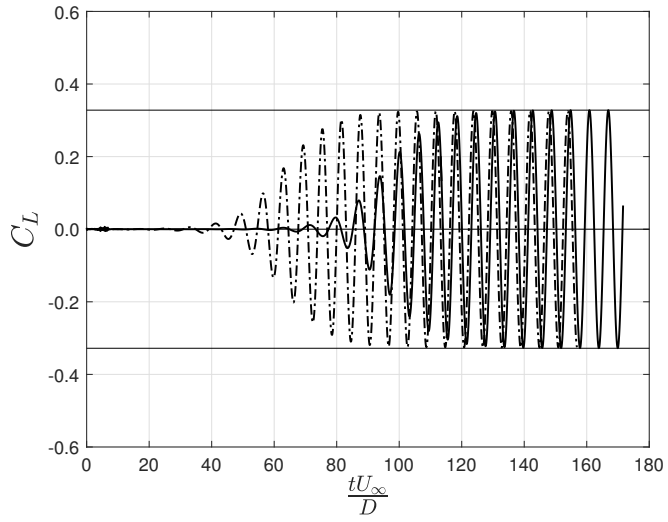


Figure 5.20: Flow past a cylinder: snapshots of the vorticity, $(\omega D)/U_\infty$, at different times near the onset of the two-dimensional vortex shedding; $(tU_\infty)/D = 68.65, 79.63, 90.62$, and 101.60. As in [102], the time between the snapshots approximately corresponds to twice the shedding period. The right-hand side limit of the box indicates the transition to the impoverished zone (using both aspects of the hp-adaptivity) near the outflow boundary.



(a) Drag coefficient



(b) Lift coefficient

Figure 5.21: Flow past a cylinder: Drag (a) and lift coefficients (b). First numerical setup (- · -) and second numerical setup (—). The thin horizontal lines (—) in Fig. 5.21(b) corresponds to the extrema from Canuto and Taira [20].

5.4.2 Simulation using the DG-VPM solver

Figure 5.22 shows the computational domain of the co-simulation. The Lagrangian computational domain is such that $\Omega_L = [-12.6, 28.36]\times[-12.6, 13.32]$. The subregion where the Navier-Stokes equations are solved is such that $\Omega_L^{NS} = [-10.0, 25.0] \times [-10.0, 10.0]$, thus matching the subregion of the second simulation in Section 5.4.1 where no damping is applied using hp-adaptivity (see Fig. 5.19(a)). The sponge layer introduced in Section 4.5.1 is then applied on a two-diameter wide region. A quadratic sponge with a target-damping $\eta_{\text{target}} = -20dB$ is used. The sponge layer is then followed by a 10-particles wide region where the radiation boundary condition, Eq. (2.21), is applied. The last subregion of the Lagrangian computational is used to reach suitable Lagrangian grid sizes in each direction for the FFTs for the Poisson problems as in Section 3.3.

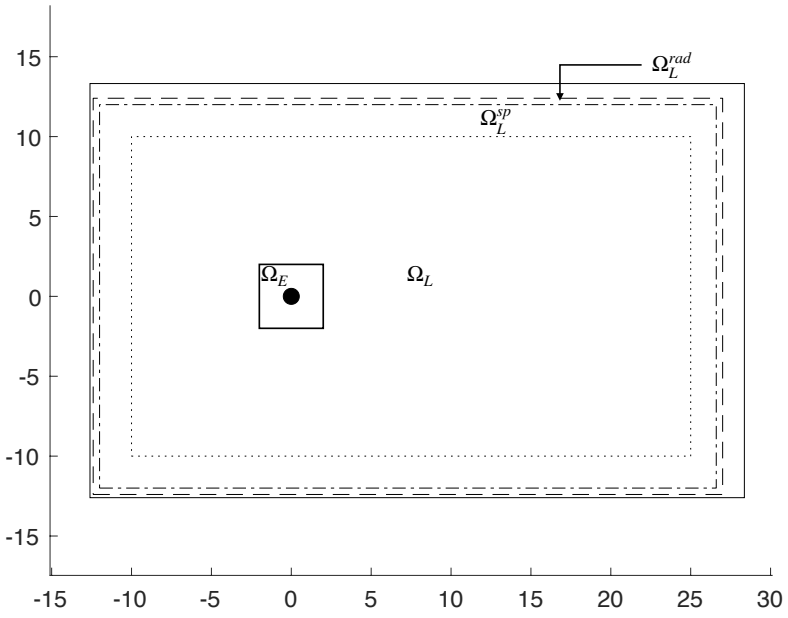


Figure 5.22: Flow past a cylinder (DG-VPM): Lagrangian computational domain, Ω_L (—), and near-body Eulerian subdomain, Ω_E (—). The cylinder is depicted as the filled black circle. A sponge layer is used in the region denoted as Ω_L^{sp} (·····) and comes before a radiation boundary condition in Ω_L^{rad} (-·-). The additional outer zone (- -) aims at reaching a suitable mesh size for the FFTs used for the Poisson problems.

The Lagrangian mesh size was taken as $h_L/D = 1/25$; the VPM solver is thus indeed under-resolved in the near-wall region. The cylinder was taken

into account in the VPM solver using the penalization of Chapter 3 with the implicit time stepping technique. The penalization parameters are such that $\phi = 0.4, (\lambda D)/U_\infty = (\lambda_T D)/U_\infty = 1.0 \cdot 10^5$. A ramp allowing to progressively penalize the mass conservation equation proved useful to limit the initial transient. Second order finite difference operators were used in the Lagrangian solver. The M_6^* redistribution scheme and the discrete filter iterated four times was used to remesh and filter particles on the Lagrangian grid at each time step. The remeshing period at each time step was chosen for the computational efficiency of the correction step as explained in Chapter 4. The optimal use of the VPM solver would, however, require to choose the remeshing period so that it reflects the requirement ensued by the physical Lagrangian distortion.

The Eulerian subdomain is a square region surrounding the cylinder, $\Omega_E = [-2.0, 2.0] \times [-2.0, 2.0]$. Curved quadrilateral elements with fourth order interpolation functions were used; a total of 1020 elements constitutes the Eulerian mesh represented in Fig. 5.23. The resolution in the near-body region is approximately 33% coarser than the resolution used in that region for the second simulation of Section 5.4.1, which was designed to provide reference results. The resolution at the wall was thus such that 24 elements covered the perimeter of the cylinder as depicted in Fig. 5.23. In Fig. 5.23, the elements are colored by the recursion level used for their subdivision for the correction of the thermodynamic quantities. The circulation and dilatation were corrected using contour integrals on VPM cells recursively subdivided twice. The parameters of the blending function were the typical values given in Section 4.4.3; $d_{bnd}/h_L = 2$ and $l_b/h_L = 8$. The M_6^* scheme was also used to interpolate the boundary conditions at the quadrature points of the outer boundary of the Eulerian solver, $\partial\Omega_E$. The Lagrangian mesh size is also in accordance with the requirement of Eq. (4.23). Finally, the time step used in both solver was taken as $\Delta t U_\infty/D = 2.5 \cdot 10^{-3}$.

The initialization was performed using a short pre-simulation with the DG solver only. The purpose of this pre-simulation was to allow the initial acoustic transient to reach the region outside the Eulerian subdomain of the co-simulation. The results of this pre-simulation were then projected onto the mesh used in the co-simulation, which in turn was used to initialize the Lagrangian solver using a single correction step as illustrated in Fig. 5.24 for the vorticity field. Although this approach reduced the acoustic transient, it generated a spurious baroclinic vorticity near the boundaries of the Eulerian

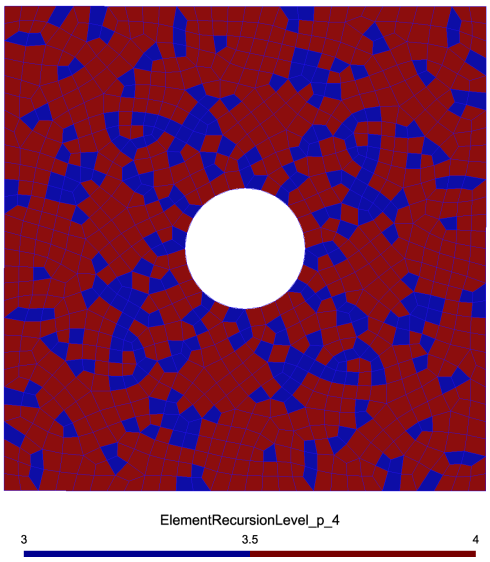
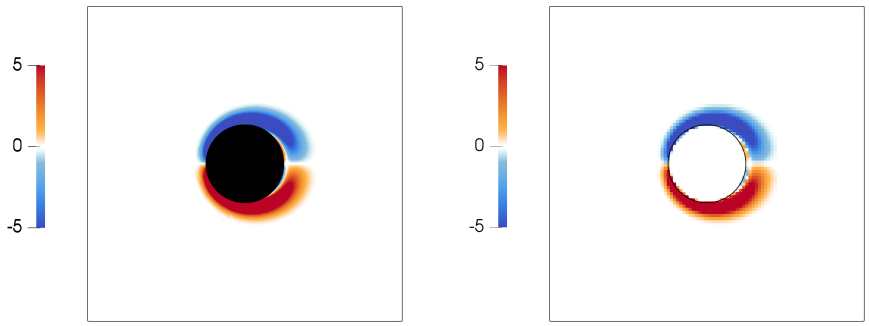


Figure 5.23: Flow past a cylinder (DG-VPM): Curved quadrilateral elements are used in the Eulerian subdomain, $\Omega_E = [-2.0, 2.0] \times [-2.0, 2.0]$. The elements are colored by the recursion level as defined by Eq. (4.36) and used to subdivide them for the correction of the thermodynamic quantities.

solver. This vorticity was successfully evacuated by the sponge layer.



(a) Initial vorticity (Eulerian solver) (b) Initial vorticity (Lagrangian solver)

Figure 5.24: Flow past a cylinder (DG-VPM): Initial vorticity field in the Eulerian solver (a) and in the Lagrangian solver (b). The initial vorticity field in the Lagrangian was obtained using contour integrals on the VPM cells. Only the edge of the cylinder is shown in (b) in order to observe the vorticity computed (and preserved) inside the cylinder.

Similarly to Fig. 5.20, snapshots of the vorticity field, $(\omega D)/U_\infty$, at different times near the onset of the instability are shown in Fig. 5.25. The same subregion as in Fig. 5.20 is shown. The effect of the sponge layer can be seen on the exiting structures on the right of the box outline. Close-up views of the vorticity field in the near-body region are also shown in Figure 5.26 at times spread over a period that approximately corresponds to twice the shedding cycle when the periodic regime is well established; the first and last snapshots are indeed almost identical. These close-up views illustrate the accurate resolution of the near-body region by the DG solver. Here again, the Eulerian and the Lagrangian solutions are continuous across the boundaries of the Eulerian solver.

Figure 5.27 shows snapshots of the vorticity field, of the norm of the momentum, $(\rho\mathbf{u})/(\rho_\infty U_\infty)$, and of the pressure coefficient, $C_p = (p-p_\infty)/(\frac{1}{2}\rho_\infty U_\infty^2)$, at time $(tU_\infty)/D = 150$, which was also considered in Fig. 5.26(b). Figure 5.27(b) thus exhibits the deficit of momentum in the wake of the cylinder, while Fig. 5.27(c) shows the pressure deficit in the near-wake and in the core of the vortices of the von Kármán vortex street.

The results are thus in good qualitative agreement with the results obtained with the DG solver alone. Figure 5.28 compares the time evolution of the drag and lift coefficients with the results obtained with the second numerical setup of Section 5.4.1. Table 5.2 complements Table 5.1 with the results of this co-simulation. The mean drag coefficient is 0.67% lower than the reference results obtained with the DG solver alone. It, however, exhibits stronger oscillations than the reference results, while the lift coefficient, on the contrary, exhibits weaker oscillations. Finally, the shedding frequency is also slightly lower than the reference results (-1.79%). The mean characteristics of the flow are thus found to be in good quantitative agreement with the reference results. The fluctuations are slightly less accurate but it should be noted that the amplitude of the drag fluctuations are two orders of magnitude lower than the mean drag coefficient.

The disparities observed in the sensitive lift and drag fluctuation amplitudes may have been sparked by the propagation of discrepancies between the solvers. Any mismatch between the near-body Eulerian solution and the Lagrangian solution may enter in a feedback loop; a discrepancy may go back and forth from one solver to the other, either through the imposition of the boundary conditions or the correction step. This loop comes into play in this

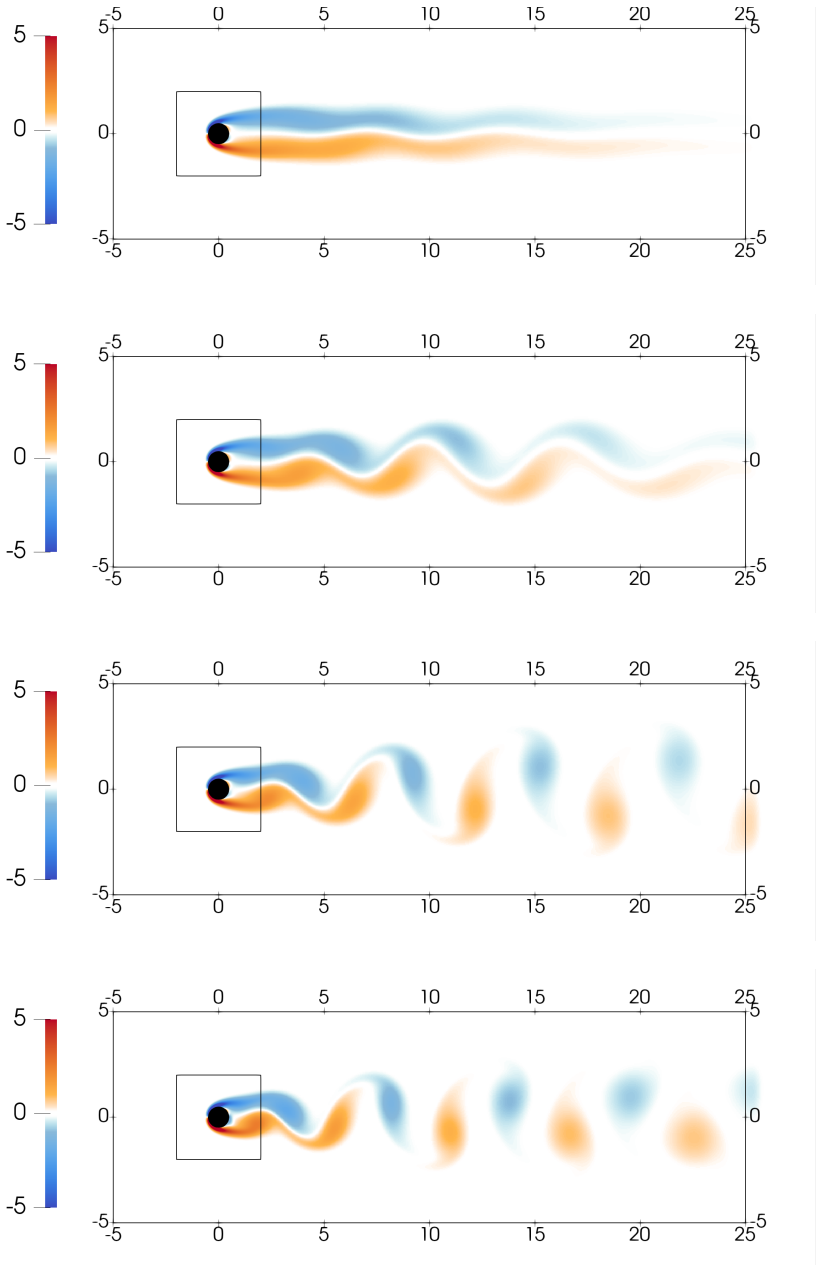


Figure 5.25: Flow past a cylinder (DG-VPM): snapshots of the vorticity, $(\omega D)/U_\infty$, at different times near the onset of the two-dimensional vortex shedding; $(tU_\infty)/D = 40, 52, 64$, and 76. The time between the snapshots approximately corresponds to twice the shedding period. The right-hand side limit of the box indicates the beginning of the sponge region.

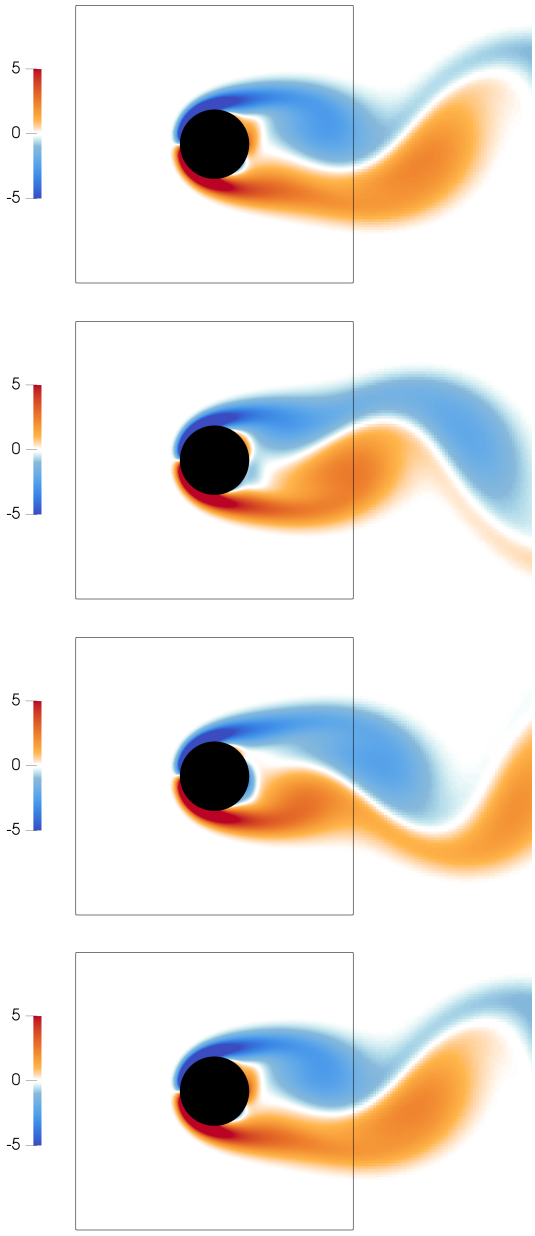


Figure 5.26: Flow past a cylinder (DG-VPM): close-up views of the vorticity field, $(\omega D)/U_\infty$, in the near-body region at different times spread over two shedding periods when the instability is well established ($(tU_\infty)/D = 146, 150, 154$, and 158). The Eulerian solution is shown in the Eulerian subdomain (—).

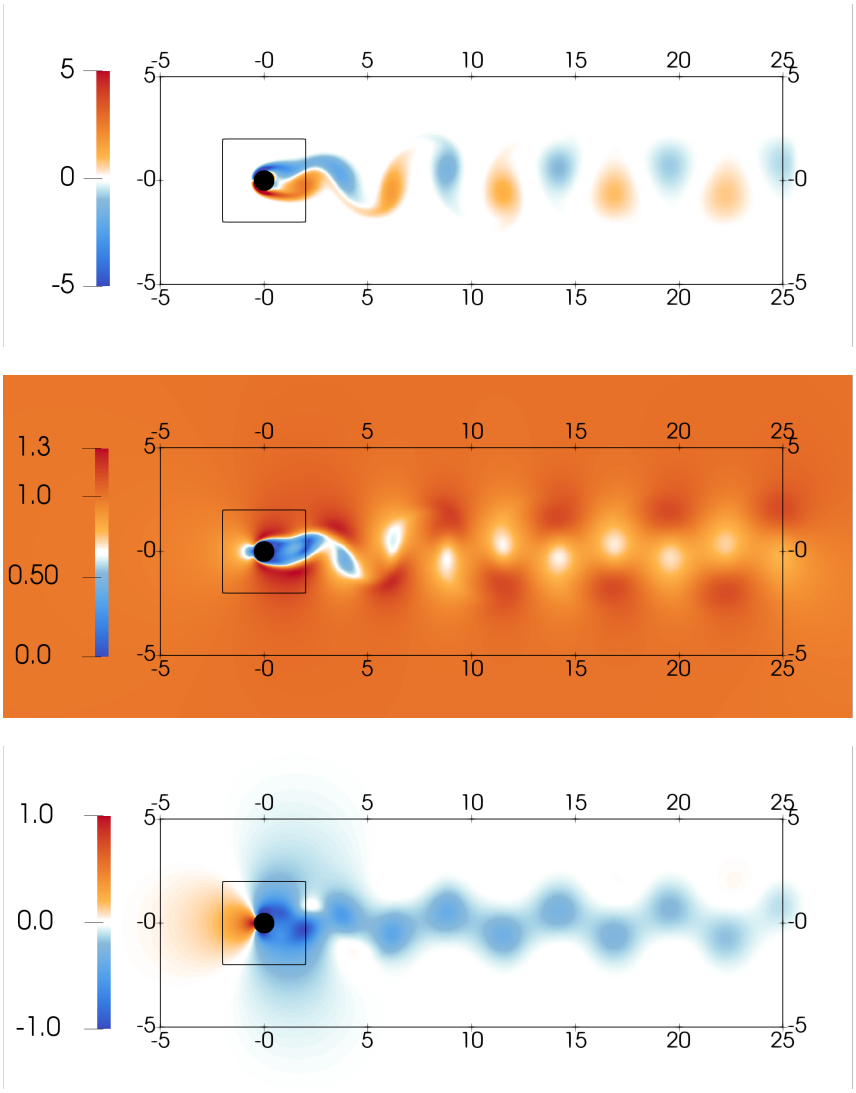
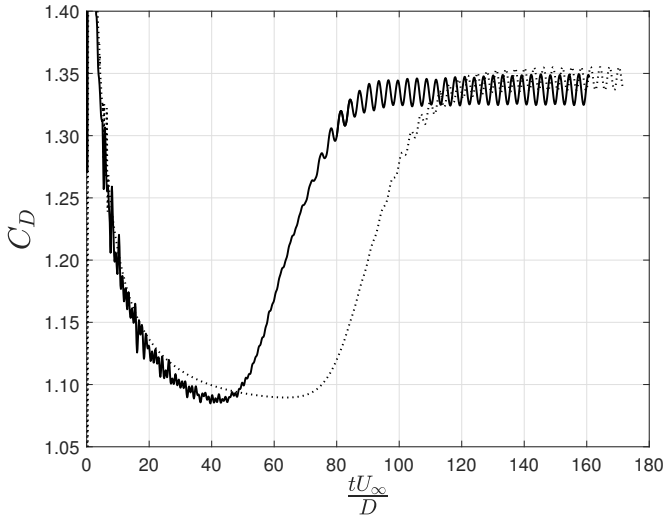
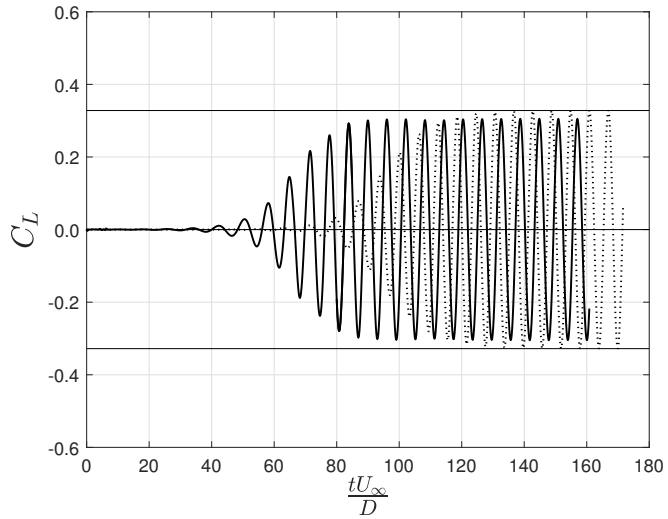


Figure 5.27: Flow past a cylinder (DG-VPM): snapshots of the vorticity field, $(\omega D)/U_\infty$, of the norm of the momentum, $(\rho \mathbf{u})/(\rho_\infty U_\infty)$, and of the pressure coefficient, $C_p = (p - p_\infty)/(\frac{1}{2}\rho_\infty U_\infty^2)$, at time $(tU_\infty)/D = 150$.



(a) Drag coefficient



(b) Lift coefficient

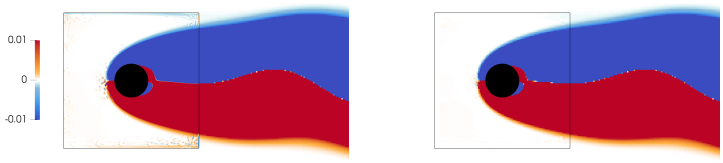
Figure 5.28: Flow past a cylinder (DG-VPM): Drag (a) and lift coefficients (b). DG-VPM solver (—), reference results from Section 5.4.1 using the DG solver alone with the second numerical setup (·····).The thin horizontal lines (—) in Fig.5.28(b) corresponds to the extrema from Canuto and Taira [20].

	$\overline{C}_D$	ΔC_D	ΔC_L	St
Canuto & Taira [20]	1.35	0.0096	0.328	0.166
DG solver (1^{st} setup)	1.347	0.0094	0.326	0.167
DG solver (2^{nd} setup)	1.346	0.0094	0.328	0.167
DG-VPM	1.337	0.0121	0.305	0.164

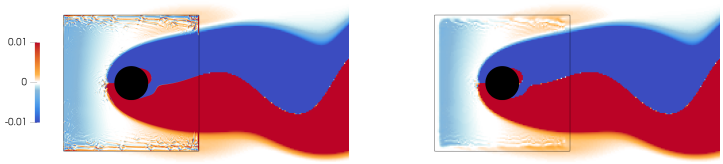
Table 5.2: Flow past a cylinder (DG-VPM): Time-averaged drag coefficient $\overline{C}_D$, amplitude of drag fluctuations ΔC_D and of lift fluctuations ΔC_L and Strouhal number St

co-simulation of a flow exhibiting strong vortex shedding, hence strong global unsteadiness. To bring it to light, a significant saturation of the vorticity field is necessary. Figure 5.29 shows close-up views of the saturated vorticity field; some of them are taken at the same times as in Fig. 5.26. The saturation level is high as the vorticity field is saturated at $[-1.0 \cdot 10^{-2}, 1.0 \cdot 10^{-2}]$ while the maximum value is such that $(\omega_{\max} D)/U_\infty \simeq 20$. Although the Lagrangian solution is, as expected, irrotational at the inflow and on most of the upper and lower boundaries of the Eulerian solver, spurious vorticity is generated at the outer boundary of the Eulerian solver. These spurious structures are then advected in the Eulerian domain and are used to correct the Lagrangian solution as soon as they move into blending transition region (see Fig. 4.14). Spurious vorticity is already visible at $(tU_\infty)/D = 52$ in Fig. 5.29(a) in the form of a narrow and coherent band of negative sign on the upper boundary and of positive sign on the lower boundary. Discontinuities in the Eulerian solution are also made visible using this saturation level. A similar band can be observed at the inflow if the saturation is chosen as $[-1.0 \cdot 10^{-3}, 1.0 \cdot 10^{-3}]$ with a negative sign above $y = 0$ and a positive sign below. The spurious vorticity at the inflow increases and changes topology as the vortex shedding instability reaches the cylinder (Fig. 5.29(b)). Spurious vorticity is then emitted from the inflow boundary in a "curtain-like shape" of alternating sign (Figs. 5.29(c)-5.29(e)). The intensity reaches a maximum as the vortex shedding is established; the feedback loop thus remains stable in this test case. It is likely that these discrepancies are, to some extent, responsible for the less accurate lift and drag fluctuation amplitudes in the co-simulation.

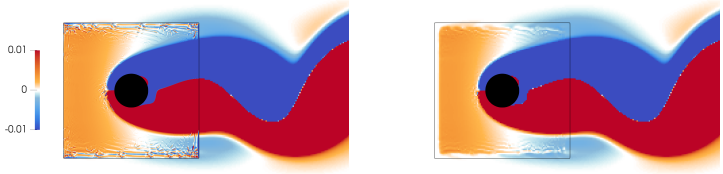
Dividing the time step used in both solvers by a factor 2 allowed to verify that the simulations are converged in time. This thus points to issues in the spatial schemes to explain the slight discrepancies in the lift and drag fluctuation amplitudes as well as the spurious vorticity brought to light in this test



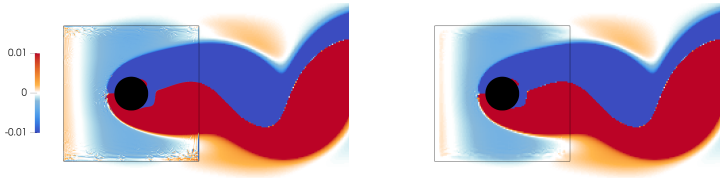
(a) $\frac{t\,U_\infty}{D} = 52$



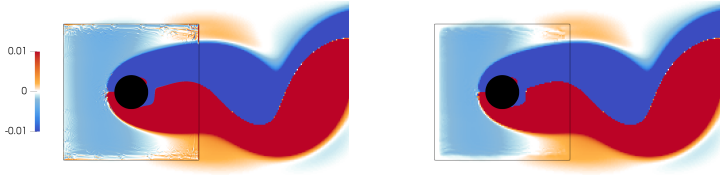
(b) $\frac{t\,U_\infty}{D} = 64$



(c) $\frac{t\,U_\infty}{D} = 76$



(d) $\frac{t\,U_\infty}{D} = 88$



(e) $\frac{t\,U_\infty}{D} = 100$

Figure 5.29: Flow past a cylinder (DG-VPM): Close-up views of the vorticity, $(\omega D)/U_\infty$, in the Eulerian solver (left) and in the Lagrangian solver (right) at times $(tU_\infty)/D = 52, 64, 76, 88$, and 100. The time between the snapshots approximately corresponds to twice the shedding period. The vorticity is saturated between $[-1.0 \cdot 10^{-2}, 1.0 \cdot 10^{-2}]$. The first three close-up views are taken at the same times as the last three snapshots of Fig. 5.25.

case.

For verification purposes, Figures 5.30 and 5.31 compare the integrals of the vorticity, dilatation, and of the solution vector in the Eulerian subdomain as computed by the DG solver and the VPM solver (after the correction step)[3]. These integrals are found to be in good agreement, which indicates that the correction was correctly performed in this test case.

[3]The accuracy of the integrals differs from one solver to the other. The identification of the VPM cells located in the Eulerian subdomain is performed using the cell center. As a consequence, part of the VPM cells near the outer boundary or near the cylinder might be partly outside the Eulerian subdomain or inside the obstacle. Moreover, the polynomial functions of the Eulerian solution are exactly integrated while the integrals in the VPM converges with the Lagrangian mesh size, h_L.

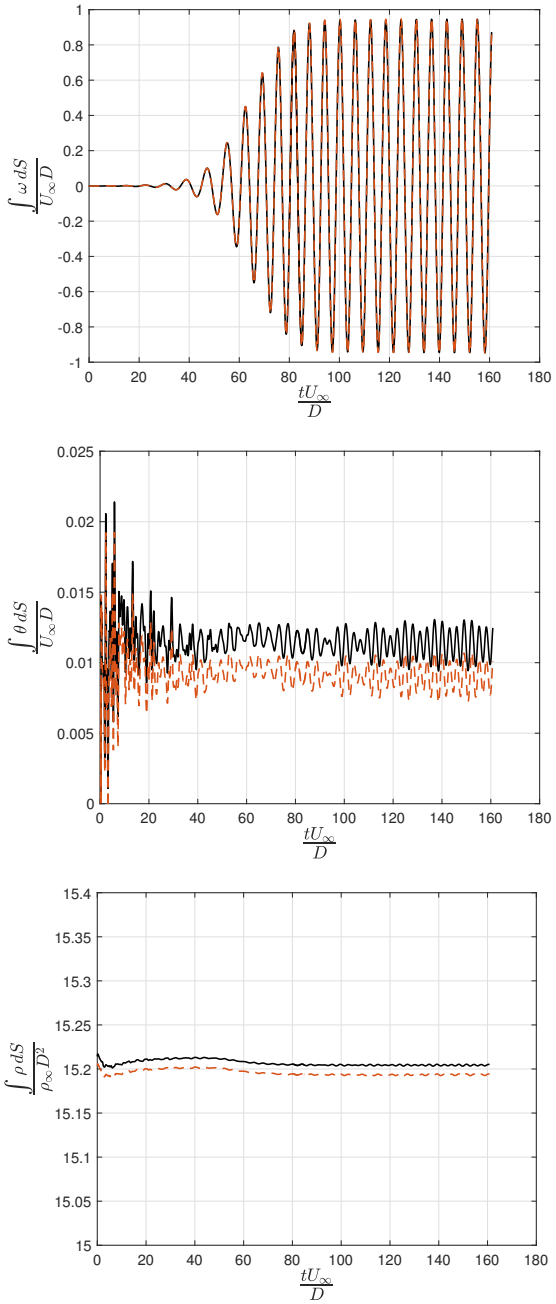


Figure 5.30: Flow past a cylinder (DG-VPM): Integrals of the vorticity (a), the dilatation (b), and the density (c) in the Eulerian subdomain, Ω_E, as measured in the Lagrangian solver after the correction step (—) and in the Eulerian solver (- -).

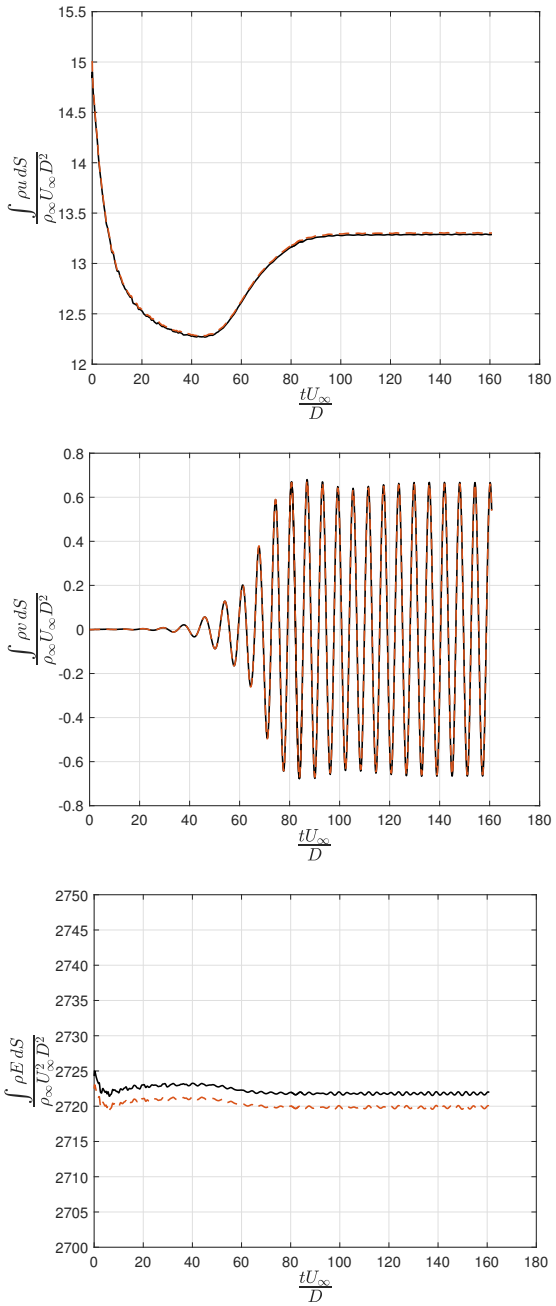


Figure 5.31: Flow past a cylinder (DG-VPM): Integrals of the longitudinal (a) and transverse components (b) of the momentum, and of the energy (c) in the Eulerian subdomain, Ω_E, as measured in the Lagrangian solver after the correction step (—) and in the Eulerian solver (- -).

5.5 Flow past a cylinder at Re = 550 and M = 0.1

The incompressible flow past a cylinder at a Reynolds $Re = \frac{\rho_\infty U_\infty D}{\mu} = 550$ has been extensively studied and is a classical benchmark test case. It was already studied using the VPM solver for compressible flows reduced to studying incompressible flows in Chapter 3. The cylinder was then taken into account using a penalization of the Navier-Stokes equations. The purpose of this Section is to assess the coupling methodology for compressible flows against reference results from the literature on that test case. The freestream Mach number will thus be taken sufficiently small, $M = \frac{U_\infty}{a_\infty} = 0.1$, for it to be considered as an incompressible flow. The temporal evolution of the drag coefficient will be compared to reference results at shorter times than it the previous section in which the Reynolds and Mach numbers were such that $Re = 100$ and $M = 0.1$. This assessment was not provided by Canuto and Taira [20] and the validity of the coupling methodology in shorter times $\left(\frac{tU_\infty}{D} \leq 5.0\right)$ has thus not been assessed yet.

The initial condition was here imposed following the work of Van Dyke [155]. In this work, a perturbation of the equation of motion is performed using the freestream Mach number as the perturbation parameter. The flow is thus weakly compressible. The expression of the potential field, ϕ, accounting for the first order effects of compressibility is obtained as

$$\begin{aligned}\frac{\phi}{U_\infty R} = &\left(\frac{r}{R} + \frac{R}{r}\right)\cos\vartheta \\ &+ M^2\left[\left(\frac{13}{12}\frac{R}{r} - \frac{1}{2}\left(\frac{R}{r}\right)^3 + \frac{1}{12}\left(\frac{R}{r}\right)^5\right)\cos\vartheta \right. \\ &\left. + \left(\frac{1}{12}\left(\frac{R}{r}\right)^3 - \frac{1}{4}\frac{R}{r}\right)\cos 3\vartheta\right]\end{aligned} \tag{5.1}$$

The dilatation field is readily obtained as $\theta = \nabla^2\phi$,

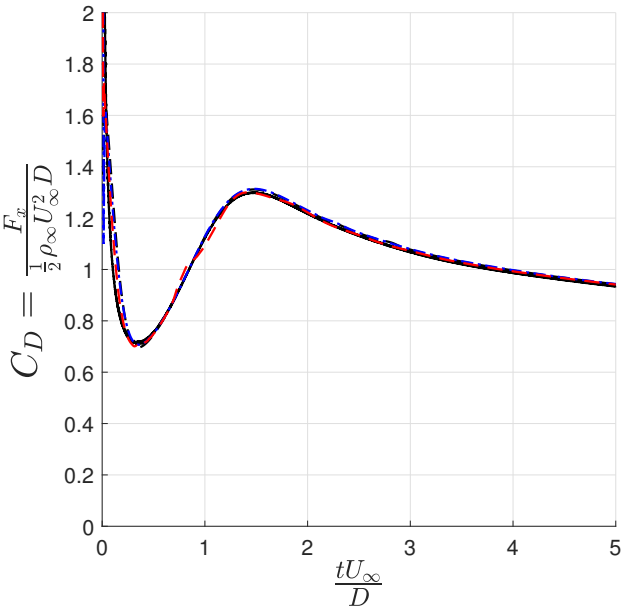
$$\frac{\theta R}{U_\infty} = 2M^2\left[\left(\left(\frac{R}{r}\right)^7 - 2\left(\frac{R}{r}\right)^5\right)\cos\vartheta + \left(\frac{R}{r}\right)^3\cos 3\vartheta\right] \tag{5.2}$$

The initial vorticity field is zero. The initial velocity field is thus given by $\mathbf{u} = \nabla\phi$. As explained in [155] and observed in Eqs. 5.1 and 5.2, the first order effects of the compressibility are independent of the thermodynamic properties of the fluid. The temperature, density and the pressure fields are then obtained assuming an homentropic flow. This initialization methodology is

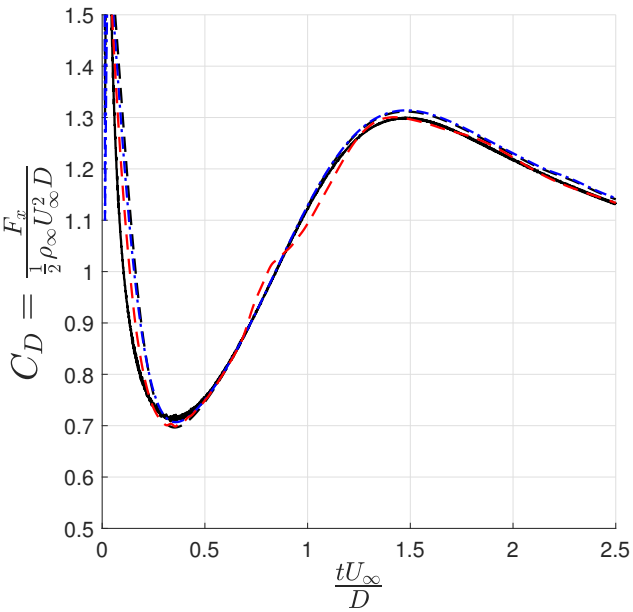
also described in Chapter 11 of [3] starting from the general *velocity potential equation*. This initialization generates a much weaker initial transient than the one used in Section 5.4.

When using the Eulerian solver, the same computational setups as in Section 5.4.1 were here also used. Indeed, they should be well suited to simulate the flow at this higher Reynolds number as they constitute exaggerated setups for the simulation of the flow at $Re = 100$. The simulation using the coupling methodology was performed using the same numerical parameters as in Section 5.4 except that the Lagrangian domain was here chosen as $\Omega_L = [-6.4, 6.4] \times [-6.4, 6.4]$. Moreover, only a radiation boundary condition was used as only acoustic perturbations reach the numerical boundaries in the course of this test. As the freestream Mach number is here small, using a square Lagrangian computational domain indeed allows to easily discriminate the effect of the truncation of the dilatation field on the drag coefficient. The results will be compared to those obtained with a VPM method for incompressible flows fitted with a state-of-the-art Immersed Interface Method to account for the cylinder [58] (with $h/D \simeq 200$).

Figure 5.32 shows the drag coefficient obtained using the Eulerian solver with the two different numerical setups as well as the results obtained with the coupled solver. As expected for such a low Mach number, the results using the compressible flow solvers are in good agreement with the results from the incompressible flow solver. However, the initialization from [155] is still only a first order solution of the compressible and impulsively started flow past a cylinder, thus not the exact solution of the Euler equations. Moreover, the temperature field obtained at the wall is different from the isothermal boundary condition that we here use. As a consequence, a moderate acoustic transient is still generated from the wall, and that translates into a cylindrical wave in the dilatation field. As can be best observed in the zoom on the early times provided on Fig. 5.32(b), the truncation of this transient dilatation at the outer boundary of the Lagrangian subdomain, which starts at $t \simeq 0.6$, results in a slight oscillation around the reference results. After that, the results using the coupling methodology are also found to be in good agreement with the reference results.



(a) Drag coefficient



(b) Drag coefficient (zoom on early times)

Figure 5.32: Drag coefficient for the flow past a cylinder ($Re = 550$, $M = 0.1$): Reference results from [58] (—), DG solver using the first (- -) and second (-·-) numerical setups of Section 5.4.1, DG-VPM solver (- -).

5.6 Conclusions

In this chapter, the DG-VPM solver, whose building blocks were developed throughout this manuscript, was applied to a series of test cases. In the first two test cases, two types of co-simulations were performed, a *one-way coupling* co-simulation and a *two-way coupling* co-simulation. In a one-way coupling co-simulation, the Lagrangian solution is not corrected and the purpose of the co-simulation is to assess the imposition of the boundary conditions of the Eulerian solver, without any possible interactions with the correction step. The two-way coupling aims at testing the full coupling methodology. In all test cases involving an obstacle, the capability of the Eulerian solver to use curved elements was exploited.

The co-simulation of the co-rotating vortex pair, which was performed in Chapter 2 using the VPM solver only, did not involve any obstacle. A one-way coupling co-simulation showed that the boundary conditions were properly imposed to the Eulerian solver. A two-way coupling using two coincident meshes then allowed to test the correction procedure without a nearest-neighbor projection, and the required non-conservative renormalization. A co-simulation moving closer to the intended use of the Eulerian solver was then performed using an unstructured mesh. The results of all three cases were shown to be in an excellent quantitative agreement with the reference results from the literature.

The second test case is the scattering of sound of a cylinder generated by a single localized source. This constitutes one of the benchmark test cases proposed in in the Second Computational Aeroacoustics workshop on Benchmark Problems [149]. It was already considered in Chapter 3 to validate the penalization allowing the VPM solver to take the obstacle into account. A wide Eulerian subdomain surrounding the cylinder as well as the locations of the observation points of [99] was first considered. The source of sound, however, was initially located outside of the Eulerian domain; a flawless transmission of the flow from the Lagrangian solver to the Eulerian solver through the proper imposition of the boundary conditions was thus of prime importance to match the analytical solution. A one-way coupling co-simulation allowed to verify it without any possible interaction with the correction procedure; the Eulerian solution was shown to be in excellent agreement with the analytical solution. A two-way coupling co-simulation was then performed to test the full methodology. The continuity of the solution across the outer boundary of the Eulerian subdomain was demonstrated. The Eulerian solution was here

again in excellent agreement with the analytical solution, which demonstrates that the correction of the Lagrangian solution did not interfere with the imposition of the boundary conditions. The correction provided by the Eulerian solver to the Lagrangian solver allowed the latter to also exhibit an excellent agreement with the analytical solution. A second and more ambitious two-way coupling co-simulation involving an Eulerian subdomain narrowly surrounding the cylinder was then performed; the Eulerian subdomain was restricted to a square of 2D side. Here again, the results were found to be in excellent agreement with the analytical solution. The improvements brought by the near-body Eulerian solver to the Lagrangian solution allowed to obtain results of quality similar to those of the Lagrangian solver alone using a grid spacing twice as small.

The flow past a cylinder at $Re = 100$ and $M = 0.1$ was then simulated, first using the Eulerian solver alone as a reference simulation and then using the DG-VPM solver. As expected, the DG solver provided results in excellent quantitative agreement with reference results from the literature. It comes as no surprise as Argo has already been successfully applied to a variety of validation test cases and to challenging industrial cases [73, 22, 25, 72]. The results provided by the co-simulation were mixed. The method proved to be robust; it was able to evacuate the initial transient and to complete the full establishment of the 2-D vortex shedding. The solution was, furthermore, shown to be continuous across the boundaries of the Eulerian near-body solver. Overall, the results were thus in qualitative agreement with the results obtained with the DG solver only. A good quantitative agreement was obtained on the mean flow characteristics; the mean drag coefficient was only 0.67% lower than the reference results. The shedding frequency obtained in the co-simulation was also in good agreement with the reference value. The small and sensitive lift and drag fluctuation amplitudes were, however, less accurately captured. It was verified that the simulation was converged in time. The roots of these discrepancies are of course to be found in the imposition of the coupling boundary conditions and / or in the correction step. As stated before, failing to communicate good quality information from one solver to the other leads to a feedback loop of damaged quantities. In that respect, the periodic vortex shedding that occurs in this flow is unforgiving as it continuously (periodically) shed strong vortices of alternating signs. At this stage, this strong global unsteadiness seems to perturb the coupling methodology.

Then, in order to verify that discrepancies were not already observable at earlier times, the flow past a cylinder was simulated again using a suitable initialization. The initialization of the compressible solvers was here performed using an approximate solution accounting for the first order effects of compressibility. Rather than considering the flow at $Re = 100$ and $M = 0.1$ as previously, the Reynolds number was here increased to $Re = 550$, as the flow characteristics at these earlier times were not studied by Canuto and Taira [20]. On the contrary, the flow past a cylinder at $Re = 550$ is a very well documented test case at early times using incompressible flow solvers. The results using the Eulerian solver and the coupled solver were shown to be in good agreement with the results obtained using a VPM solver for incompressible flows fitted with an Immersed Interface Method [58].

As stated above, the spurious vorticity observed at the outer boundary of the Eulerian solver takes its roots from discrepancies between the Eulerian and Lagrangian solutions. It is first observed in the Eulerian solution and is thus probably due to the coupling boundary conditions. Some light could be shed on the source of these discrepancies by:

- evaluating the baroclinic term, $\frac{\nabla\rho\times\nabla p}{\rho^2}$, in a region near the left, lower and upper faces of the outer boundary of the Eulerian subdomain. Indeed, the flow should remain irrotational on these three faces of the square domain used here. A non-zero value observed in that region would produce spurious vorticity in the Eulerian solver. Along the same lines, it would be interesting to redo the simulation using fourth order finite differences in the VPM solver rather than the second order finite differences used in the simulations of the flow past a cylinder.

- comparing the velocity field provided by the Lagrangian solver with the resulting velocity after the weak imposition of the boundary condition by the Eulerian solver. Indeed, the presence of the spurious narrow bands on the upper and lower faces of the Eulerian domain, as may be observed on Fig. 5.29(a), as well as the "curtain-like shaped" vorticity of alternating sign emitted from the inflow boundary could be the manifestation of discrepancies in the velocity of the Eulerian solver and the coupling boundary conditions.

- assessing the exact role of the penalization in the computation of the boundary conditions. As explained in Section 4.3, the penalization proves useful to avoid spurious gradients and oscillations at the solid wall. This allows to preserve a good integral representation of the solution in the

Lagrangian solver between two corrections from the Eulerian solver. At this stage, however, it remains unclear if the penalization of the kinematic quantities has a beneficial effect on the computation of the coupling boundary conditions. Indeed, when studying the flows past a cylinder at $Re = 100$ and $Re = 550$, deactivating the penalization on the kinematic quantities, ω and θ, which are elliptically involved in the computation of the coupling boundary conditions as explained in 4.3, did not significantly affect the results. The effect of this deactivation is, however, mitigated by the use of identical time steps in both solvers; the Lagrangian solution is corrected at the beginning of each of its time steps (see Fig. 1.7). It would thus be interesting to assess whether or not the penalization of the kinematic quantities is necessary for the consistency of the coupling algorithm by performing sensitivity or convergence studies and using different time steps in both solvers.

CHAPTER 6

CONCLUSIONS AND PERSPECTIVES

6.1 Conclusions

This doctoral project paves the way to a hybrid Eulerian-Lagrangian approach for compressible flows. The intended use of such a coupled solver is the accurate and computationally efficient simulation of systems for which it is necessary to capture both complex anisotropic phenomena in the near-body region and complex vortex dynamics in the wake of the system. As explained in Chapter 1, applications include the simulation of jet and wake noise, aircraft wakes in cruise condition as well as blade vortex interactions for rotorcraft. The progress made in that direction during this thesis may be summarized as follows:

- the VP method for compressible flows introduced by Eldredge in [50] and then also successfully applied in [49] has been adapted in Chapter 2 to benefit from the advances already in use in modern VPM solvers for incompressible flows. This was motivated by the efficiency analysis performed by Cottet and Poncet in [41] and Cocle et al. in [31], which demonstrated the speed-up provided by a VPM method in comparison to a grid-free methodology such as that of [50]. The underlying grid used in a VPM method indeed allows the evaluation of the spatial differential operators on the grid and the use of efficient grid-based Poisson solvers. We here elected to use a fourth-order Fourier-based Poisson solver im-

posing unbounded boundary conditions [74, 68]. Similarly to [50], radiation boundary conditions were implemented to avoid spurious reflections of radiated perturbations at the numerical boundaries. It was later complemented with a sponge methodology allowing to damp larger-amplitude flow disturbances, which is typically necessary at an outflow boundary. The VPM solver for compressible flows demonstrated its capability to capture the vortex dynamics as well as the sound generation during the merging of a pair of co-rotating vortices. This work was published in the form of a journal article [120].

- In order to allow the VPM solver to take solid obstacles into account, the Brinkman penalization of Liu and Vasilyev [99] was adapted. It imposes Dirichlet boundary conditions on the velocity and on the temperature fields inside one or multiple obstacles. This methodology comes with a number of advantages that were sought for in this work; it is a mature and non-invasive methodology that only requires to integrate additional terms in the Navier-Stokes equations without any special treatment at the boundary.

  Two different techniques were implemented and validated against benchmark test cases; an explicit and an implicit integration of the penalization terms. The second technique has the advantage to come with relaxed stability constraints when compared to the first one. To the author's knowledge, it is the first implementation of a VPM method for compressible flows fitted with a penalization technique. As explained in [102], this methodology does not offer the sharp treatment of Immersed Interface Methods [103, 104, 60] due to the regularization of the volume forcing terms. This prevents the penalization to be applied for the simulation of high Reynolds number flows at a reasonable computational cost.

The developments that were made in the first part of this manuscript may actually be seen as contributions independent from the coupling methodology. If one refers to the denominations introduced in Section 1.3.2, a whole independent *simulation unit* was here developed: a Vortex Particle-Mesh method for subsonic compressible flows past bodies. This Lagrangian solver is capable of simulating two-dimensional viscous compressible flows past bodies in an unbounded domain. It is restricted to the subsonic regime as no shock capturing methodology is implemented. When used in conjunction with the penalization method, it is also limited to moderate Reynolds number flows. How-

ever, the purpose of the near-body Eulerian solver in the coupling methodology is precisely to suppress this limitation.

- the VPM solver was coupled with Argo, a Discontinuous Galerkin (DG) solver developed at Cenaero (Gosselies, Belgium). The scales-resolving capability, the high order accuracy on unstructured mesh, the locality of the method allowing an efficient parallelization as well as appealing diffusion and dispersion properties - a capability shared with the VPM solver-, motivated the choice of Argo as the near-body Eulerian solver.

  Relying on a completely overlapping domain decomposition technique initially introduced in [44], two different pieces of information need to be communicated from one solver to the other at appropriate times: the boundary conditions for the Eulerian solver and the correction of the Lagrangian solution in the Eulerian subdomain, where it is purposely under-resolved. The presented coupling methodology is not completely agnostic about the type of solver used in the near-body region: the proper imposition of the boundary conditions here relies on the existence of a Riemann solver in the near-body Eulerian solver. The inherent implementation of characteristic boundary conditions is indeed a key element of the presented approach.

  Regarding the correction step, the whole coupling methodology relies on the good integral representation, on the Lagrangian particles, of the Eulerian solution: the correction procedure should be conservative. For the kinematic quantities, the correction was performed using high accuracy contour integrals performed on the Lagrangian cells. The thermodynamic quantities were corrected using integrals performed on the Eulerian elements followed by a projection on the nearest particles with proper renormalization, hence providing a smooth representation on the Lagrangian grid.

  The coupling methodology was tested on a series of benchmark test cases, some of them already considered in Chapter 2 and 3. Improvements brought by the DG solver were clearly exhibited, especially on the benchmark test case of the scattering of sound by a cylinder generated by a single localized source; the results obtained using the DG-VPM solver were of similar quality as the results obtained using the

VPM alone with a grid spacing twice as small. The DG-VPM solver was then tested on the flow past a cylinder with low Reynolds ($Re = 100$) and Mach ($M = 0.1$) numbers for validation purposes against [20]. The results were found to be in good quantitative agreement with the results obtained using the DG solver alone and with reference results from the literature [20]. Some discrepancies were, however, observed in the fluctuation amplitudes of the lift and drag coefficients. In that respect, the feedback loop that may occur between the imposition of the boundary conditions and the correction procedure was brought to light. This feedback loop is really inherent to the combination of the boundary conditions for the Eulerian solver and the correction it provides to the Lagrangian solver; this feedback loop is independent of the Eulerian solver used to capture the near-body region, and would also occur when coupling incompressible flow solvers. Using a suitable initialization for compressible flows, the results at early times for the flow past a cylinder at a higher Reynolds number, $Re = 550$, were shown to be in good agreement with reference results obtained using a VPM solver for incompressible flows fitted with an Immersed Interface Method. As no reference results were provided in [20] at early times, this allowed to verify that the DG-VPM solver was also in good agreement with the DG solver used alone and reference results at these early times.

The capability of the hybrid DG-VPM solver to simulate 2-D viscous compressible flows past bodies was partially demonstrated:

- the method was shown to be robust. It was indeed capable of accommodating initial acoustic and baroclinic transients and to complete full CFD simulations.
- the contours of the solution were shown to be continuous across the outer boundaries of the Eulerian solver.
- excellent agreement with analytical and reference solutions was obtained the acoustics test case of Chapter 3 while the results obtained on the flow past a cylinder were only satisfactory. This difference of performance calls for a better understanding of the intertwined effects of the imposition of the boundary conditions and the correction procedure.

At this stage, preliminary results have also been obtained when studying the subsonic flow past a stalled NACA0012 airfoil at 20° of angle of attack. The Reynolds and freestream Mach number were here chosen as $Re = \frac{\rho_\infty U_\infty c}{\mu} =$

5000 and $M = \frac{U_\infty}{a_\infty} = 0.4$ with c the chord of the airfoil. The dimensions of the Lagrangian computational domain and of the Eulerian subdomain were chosen such that $\left[\frac{L_L}{c}, \frac{H_L}{c}\right] = [12.0, 6.0]$ and $\left[\frac{L_E}{c}, \frac{H_E}{c}\right] = [1.75, 1.25]$, respectively. The Eulerian mesh comprised 2270 linear elements and fourth order interpolation functions were used in the whole Eulerian subdomain. The Lagrangian mesh size was such that $\frac{h_L}{c} = 0.01$. In order to favor a proper imposition of the coupling boundary conditions, similar cutoff wavenumbers were used at the outer boundary of the Eulerian subdomain. Here again, identical time steps were used in both solvers, $\frac{\Delta t U_\infty}{c} = 8.0 \cdot 10^{-4}$. The simulation was initialized by imposing the freestream conditions in the whole computational domain. Figure 6.1 show the vorticity field at different times. Complex vortex dynamics are generated due to the massive flow separation. The presence of jumps in the saturated vorticity near the leading edge of the airfoil indicates that the Eulerian solution is underresolved in that region. Small jumps near the downstream boundary of the Eulerian subdomain are related to the issue already encountered when studying the flow past a cylinder and discussed in Chapter 5.4. As already stated in [44] for a similar test-case, this simulation is of course artificial as three-dimensional effects are not taken into account. It should thus be seen as a promising proof-of-concept of the hybrid methodology.

In this doctoral project, efforts towards a hybrid Eulerian-Lagrangian solver strongly benefitting from the accurate resolution of a high order Discontinuous Galerkin solver were made. To the author's knowledge, it is the first combination of a VPM solver for compressible flows with a high order DG method for the near-body region. The intended use of this coupled solver is the simulation of high Reynolds number flows also exhibiting compressibility effects.

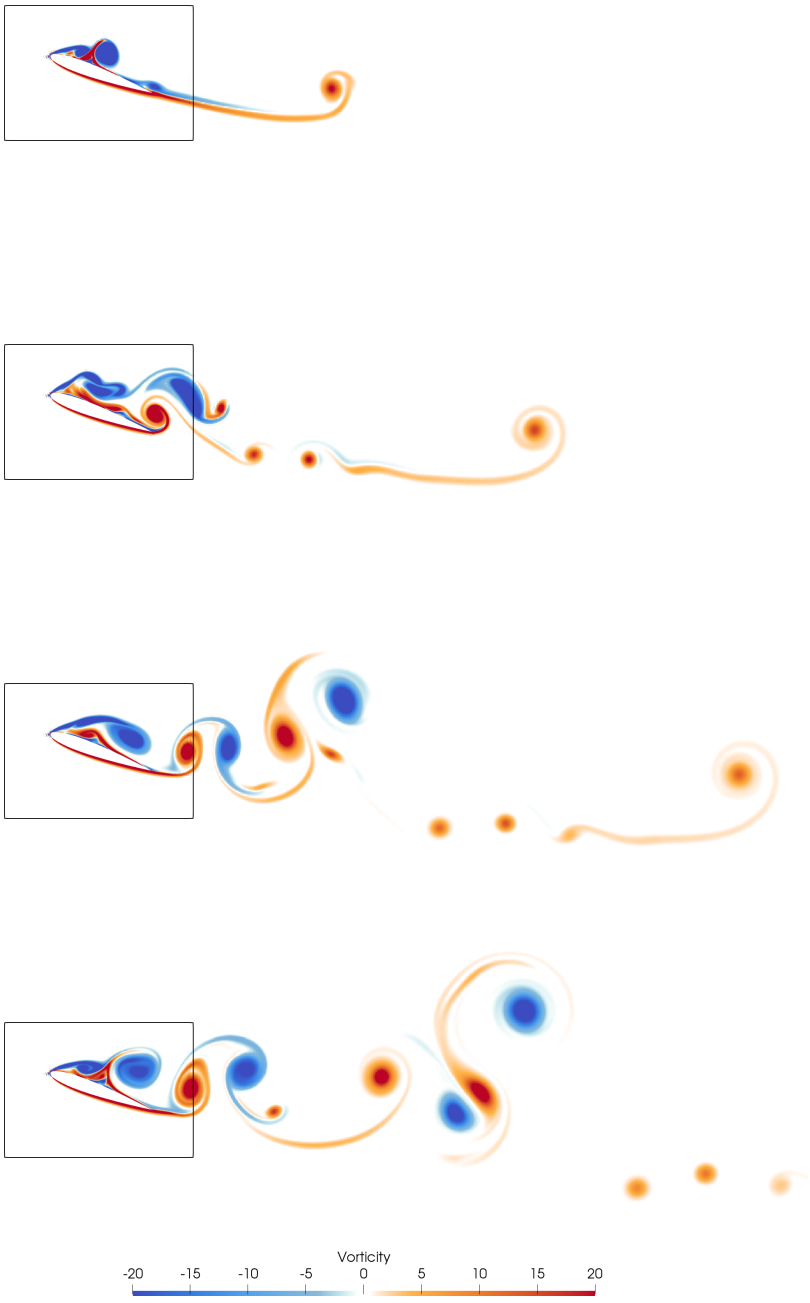


Figure 6.1: Flow past a stalled NACA0012 airfoil at $Re = 5000$ and $M_\infty = 0.4$ using the hybrid DG-VPM solver: snapshots of saturated vorticity, $(\omega c)/U_\infty$ at different times; $(tU_\infty)/c = 2, 4, 6$, and 8. The Eulerian solution is shown in the near-body region.

6.2 Perspectives

The developments made in this work towards a coupled DG-VPM solver for compressible flows bring some new interesting questions and prospects,

- Thorough sensitivity analyses could be performed in order to better characterize the hybrid solver developed in this work. These analyses could concern the following elements:
  - the extent of the Eulerian subdomain; ideally, the near-body region should be limited to a narrow region surrounding the obstacle and containing the near-body anisotropic flow phenomena.
  - the extent of the Lagrangian domain,
  - the parameters of the boundary conditions used in the VPM solver, i.e. the combination of a sponge layer and a radiation boundary condition. The radiation boundary condition could also be further improved to take into account a freestream velocity as in [150].
  - the parameters of the correction blending function,
  - the convergence properties of the methodology by performing temporal and spatial convergence studies on academic test-cases.

  As already discussed in Chapter 4, although it uses a high-order near-body Eulerian solver, the coupling methodology itself do not exhibit a high-order capability. These different sensitivity and convergence studies could also help identify necessary improvements to the numerical techniques used in the current approach.

  Having gained mastery of the different parameters of the methodology, the expected reduction of the computational cost for the applications envisioned in this work could be truly (and fairly) assessed.

- A multi-resolution technique [44, 100, 132, 69, 146] could also be used in order to better extend the far-field region in the VPM solver: a coarser resolution in the far-field region would allow to truncate the dilatation field, which is used to compute the velocity field, at a greater distance while maintaining a reasonable computational cost. A coarser resolution would increase the numerical dissipation in that region and would also facilitate the imposition of the boundary conditions. This amounts to perform a h-adaptation per block.

- An alternative to the strong and abrupt imposition of the correction provided by the Eulerian solver through a blending function, as explained in Section 4.4.3, would be to progressively drive the Lagrangian solution towards the Eulerian solution. This could be performed using a technique that would share many aspects with the sponge layer introduced in Section 4.5.1, which is itself close to the penalization of Chapter 3. The prescribed reference solution in the sponge term of Eq. (4.45) would then be the Eulerian solution. Of course, a parameter study would be necessary to determine the best "correction profile", σ, as well as the "correction strength" in order to properly enforce the correction.

- In order to reduce the important footprint of the Poisson problems on the computational cost of the VPM solver, the implementation of inflow-outflow boundary conditions would prove beneficial. The transverse direction would still be unbounded. When using a FFT-based Poisson solver and the domain doubling technique of Hockney and Eastwood [74], such boundary conditions would thus require to double the size of the computational domain in the transverse direction only. Such effort has already been made in the 2-D and 3-D incompressible flow solvers in the research group at UCLouvain in the same vein as in [28]. Although not shown in this work, a similar implementation was performed in the compressible flow solver; only the symmetries in the axial direction of the Poisson problem, Eq. (2.9), for the irrotational part of the velocity field need to be adapted.

- So far, the hybrid solver is capable of taking a single motionless and non-deforming obstacle into account. It could be extended to moving and/or deforming bodies.

- An interesting prospect is the study of higher Mach number flows. In particular, the simulation of the transonic flow past an airfoil would be the natural following step of the simulation presented in the previous Section. A smaller angle of attack would then be considered. For instance, one would be interested in the first two out of four stages of the transonic regime [158] where the regions of supersonic flow are still limited to pockets and hence are entirely contained in the Eulerian subdomain. One would then rely on a shock capturing capability of the Eulerian solver, for instance using an artificial viscosity method [126, 125].

- A 3-D implementation of the coupled DG-VPM solver is, of course, one of the exciting prospects of this work. As already stated in Chapter 2,

the additional difficulties for the VPM solver, i.e. the zero-divergence problem of the vorticity field and the evaluation of the stretching term, may be treated as efficiently and accurately as in the incompressible flow VPM solver. The extension to 3-D of the penalization introduced in Chapter 3 is straightforward and no additional difficulties in the coupling algorithm are anticipated. As a matter of fact, the simulation of the flow past a sphere might prove easier to perform than the 2-D vortex shedding behind a cylinder considered in Chapter 5. Indeed, the flow past such an obstacle rapidly transitions to a turbulent state and does not exhibit the coherent structures encountered during a 2-D vortex shedding behind a cylinder. This can be related to the general phenomenon for three-dimensional flows called the *three-dimensional relieving effect* [3]. The flow past a sphere is indeed less constrained than the flow past a cylinder due to the extra dimension.

As already expressed in Chapter 1, the dream simulation would allow studying the flow past the helicopter rotor shown in Fig. 1.3. The high-order DG solver would be responsible for capturing the strongly anisotropic flow features in the close vicinity of the blades (boundary layers, dynamic stall, compressibility effects, etc.). The Lagrangian solver would be responsible for the advection-dominated region close to the rotor, i.e. the region between the blades in the rotor plane and the near and far wakes. Part of the vortical structures would then reenter in the Eulerian region surrounding the following blade, which would allow to study blade-vortex interactions (BVI). The rest of the vortical structures would be advected downstream, the complex vortex dynamics in the wake being accurately captured by the VPM solver.

APPENDIX A

A 2-D INCOMPRESSIBLE VPM METHOD APPLIED TO THE LID-DRIVEN CAVITY

The solver for compressible flows introduced in Chapter 2 is based on a fast 2-D framework for incompressible flows developed in the group. The purpose of this appendix is to illustrate the capabilities of the latter on the benchmark problem of a lid-driven cavity. The Navier-Stokes equations for incompressible flows are solved in a domain $\Omega = [0, L] \times [0, L]$. The boundary conditions associated to the problem are $u(x, y = L) = -U, v(x, y = L) = 0$ on the moving side, and $u(x, y) = v(x, y) = 0$ on the other sides as depicted in Fig. A.1.

This benchmark test case constitutes a classical test problem for the assessment of numerical methods, despite the singularities in the corners. The problem that is actually solved is generally referred to as *regularized driven cavity* as explained in [16]. The idea is to smooth the velocity (as well as its derivatives) so that it vanishes at the corners of the moving side. We here elected to use a smoothed boundary condition. A simple piecewise linear smoothing function introduced in [142] is used:

$$\frac{u(x, y = L)}{U} = \begin{cases} -\frac{(1-x/L)}{0.05} & \text{if } \frac{x}{L} > 0.95 \\ -1 & \text{if } 0.05 \leq \frac{x}{L} \leq 0.95 \\ -\frac{x/L}{0.05} & \text{if } \frac{x}{L} < 0.05 \end{cases} \tag{A.1}$$

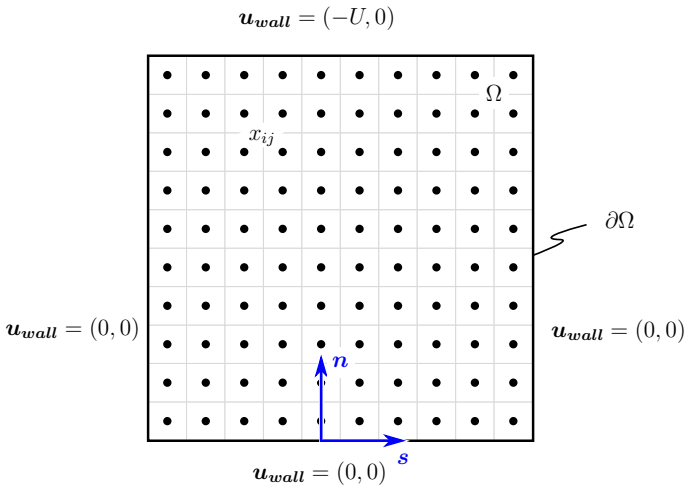


Figure A.1: Lid-driven cavity: sketch of the computational domain Ω and boundary conditions. The nodes of the Lagrangian underlying grid are represented as bullets. This sketch is an adaptation of the sketch available in [102].

Other possibilities exist; for example, the smoothing function introduced by Kuhlmann et al. [90]. One could also have a smoothed Heaviside function as defined by Eq. (4.44). The results will be compared to reference results from the literature [57, 16]. In [57], Ghia et al. solve the vorticity-stream function formulation of the two-dimensional incompressible Navier-Stokes equations using a multigrid technique. Botella and Peyret [16] used a Chebyshev collocation method together with a substraction of the leading part of the singularity as the regularization technique with the intent to produce benchmark results.

We here use the methodology developed by Marichal in his PhD dissertation [102] to enforce a no-slip condition at solid walls coinciding with the grid boundaries. This simplified geometry allowed him to study the performance of the no-slip enforcing procedure. It is here slightly modified to account for a translating wall. The no-slip condition is enforced by first computing the singular vortex sheet $\Delta\gamma$ associated to the tangential slip velocity at the wall,

$$\Delta\gamma = -\frac{\partial\Psi}{\partial n} = -(\mathbf{u}^+ - \mathbf{u}^-) \cdot \boldsymbol{s} \tag{A.2}$$

with $\mathbf{u}^+$ and $\mathbf{u}^-$ the flow velocity above and below the vortex sheet respectively, and $\boldsymbol{s}$ the unit vector tangential to the wall (see Fig. A.1). This vortex sheet is then diffused into the flow. For further details about the methodology

and the related theoretical background, see [102].

The Reynolds number is $Re = \frac{UL}{\nu} = 1000$. The mesh resolution is such that $N_x = N_y = 512$. An adaptive time step was used and chosen so that it satisfies the following stability constraints,

$$|\omega|_\infty \Delta t < 0.25 \tag{A.3}$$

$$\|\mathbf{S}\| \, \Delta t < 0.25 \tag{A.4}$$

$$\frac{\nu \Delta t}{h^2} < 0.2 \tag{A.5}$$

with $\mathbf{S} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$ the strain rate tensor. The same bounds were selected in [102]. The particles were remeshed at each time step. The simulation was run up to convergence to a steady state, defined using the following criteria:

$$\frac{|u^{n+1} - u^n|_\infty}{\Delta t |u^{n+1}|_\infty} \leq \epsilon \tag{A.6}$$

$$\frac{|v^{n+1} - v^n|_\infty}{\Delta t |v^{n+1}|_\infty} \leq \epsilon \tag{A.7}$$

where $|\cdot|_\infty$ denotes the infinite-norm computed over the computational domain. The threshold value, ϵ, was set to $1.0 \cdot 10^{-5}$ (a similar criterion was used in [16]). Figure A.2 shows the streamlines and vorticity contours at convergence to the steady-state. The contours could not be superimposed to those in Botella et al. [16] due to skewed figures in the original paper. Figure A.3 compares the vorticity profile on the moving wall as obtained with the VPM solver of [102] with results from the literature. Figures A.4 and A.5 compare the vorticity profiles on the horizontal ($y/L = 0.5$) and vertical ($x/L = 0.5$) centerlines with the reference results of [16], respectively. All of these results are found to be in excellent agreement with the benchmark results of Botella et al. [16].

As demonstrated in Marichal's PhD dissertation [102] and exhibited in this appendix in a new application of his work, the VPM method for incompressible flow thus was a sound basis for the development of the VPM solver for compressible flows introduced in Chapter 2.

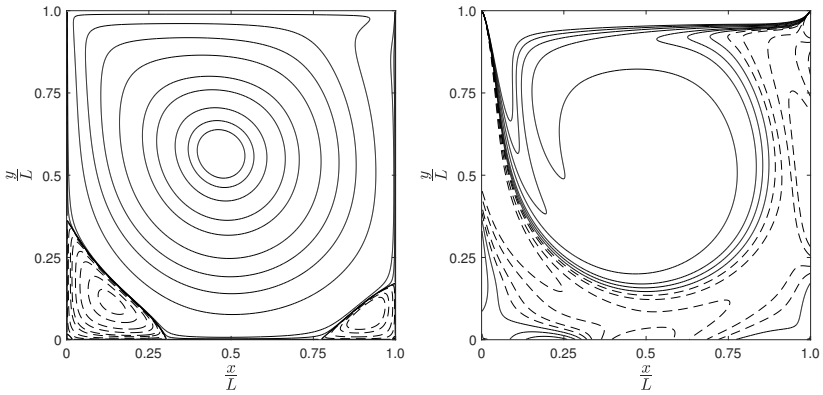


Figure A.2: Lid-driven cavity: streamlines (left) and contours of vorticity (right). The values for contours of the streamfunction and of the vorticity are taken as in [16] (Table 7 and Table 8 in their paper). The negative and positive contours are shown using dotted lines and solid lines, respectively. The zero-level contour is shown using a solid line.

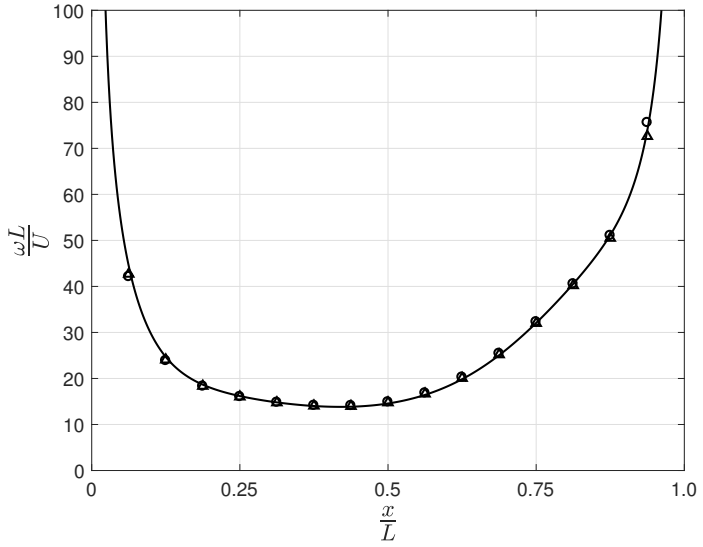


Figure A.3: Lid-driven cavity: Vorticity profile on the moving wall ($y = L$). Ghia et al. [57]: circles, Botella et al. [16]: triangles, VPM method of [102]: solid line

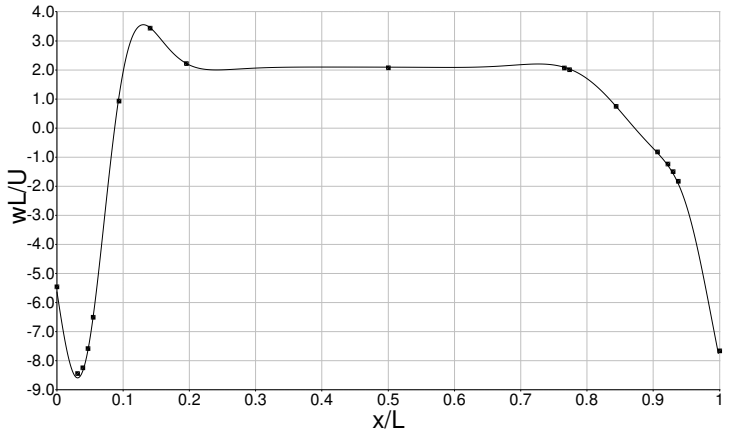


Figure A.4: Lid-driven cavity: Vorticity profile on the horizontal centerline ($y = 0.5L$). Botella et al. [16]: squares, VPM method of [102]: solid line

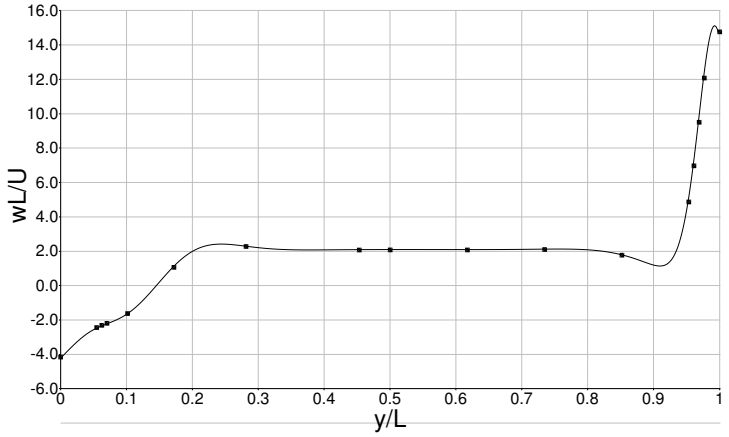


Figure A.5: Lid-driven cavity: Vorticity profile on the vertical centerline ($x = 0.5L$). Botella et al. [16]: squares, VPM method of [102]: solid line

APPENDIX B

SECOND AND FOURTH ORDER "ISOTROPIC" FINITE DIFFERENCES SCHEMES

Explicit finite difference schemes that are less sensitive to the grid orientation than the usual "cross" schemes were used in this study, both for first order derivatives and for the Laplacian. Their expressions are provided hereunder and their performance is assessed by modal analysis performed on the 2-D advection equation for the former, and 2-D diffusion equation for the latter.

Laplacians

The second-order isotropic discretization for the Laplacian scheme is built under the constraint that the leading order term of the truncation error is itself a Laplacian. The 2-D and 3-D schemes have the following form

$$\nabla_h^2 \boldsymbol{\omega}\Big|_{i,j,k} = \nabla^2 \left(\boldsymbol{\omega} + \frac{h^2}{12} \nabla^2 \boldsymbol{\omega} \right) \Bigg|_{i,j,k} + \mathcal{O}\left(h^4\right). \tag{B.1}$$

Figure B.1 provides the stencils and the coefficients of the 2-D "isotropic" or "square" scheme that was already considered in [32] and of the classical "cross" scheme. In 3-D, a family of such schemes can be derived; all of them having the exact same leading order term of the truncation error in Eq. (B.1).

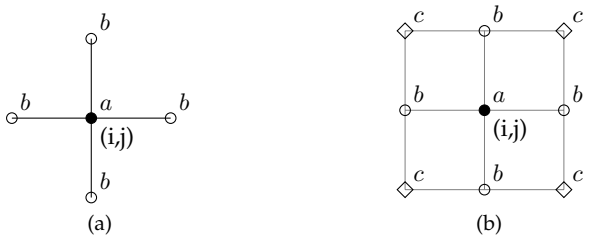


Figure B.1: Illustration of 2-D second order Laplacian schemes: the classical "cross" scheme (B.1(a)), and the "square" scheme (B.1(b)). The coefficients for the classical "cross" scheme are $a = -4$ and $b = 1$. The coefficients for the square scheme are $a = -20/6$, $b = 4/6$ and $c = 1/6$ ([32], the figure is reproduced from [31])

Schemes of this family were derived in different publications (see for example [91], [123] and [152]). Cocle et al. [31] used such a second-order isotropic scheme to simulate complex and turbulent unbounded flows using a VPM method for incompressible flows.

We now present the 2-D fourth-order scheme previously derived by Patra and Karttunen in [123] and meeting a similar constraint; the leading order term of the truncation error is here required to be a bi-Laplacian of the Laplacian itself. Using Taylor series, one obtains

$$\nabla_h^2 f\Big|_{i,j} = \nabla^2 \left(f - \frac{h^4}{90} \left(\nabla^2 \left(\nabla^2 f \right) \right) \right) \Bigg|_{i,j} + \mathcal{O}\left(h^6\right). \tag{B.2}$$

The leading order term of the truncation error term is thus equal to $\left(-h^4/90\right) \nabla^6 f\big|_{i,j}$. The stencil and the coefficients of this fourth-order "isotropic" scheme as well as those of the classical cross scheme are provided in Fig. B.2. The use of the 8 other points in the 5x5 square would have lead to a family of schemes also meeting the constraint on the truncation error. It is however straightforward to show that the constant in the leading order term of the truncation error remains unchanged (i.e. $C = -1/90$ for all the members of the family).

The performance of the isotropic 2-D Laplacian schemes (Eqs. (B.1) and (B.2)) and of the usual cross schemes are now compared through a modal analysis performed on the 2-D diffusion equation, $\frac{\partial f}{\partial t} = \nu \nabla^2 f$. Given $f(x, y, t) = A(t) \exp\left(i \left(k_1 x + k_2 y\right)\right)$ with k_1 and k_2 the wave numbers in the x- and y-

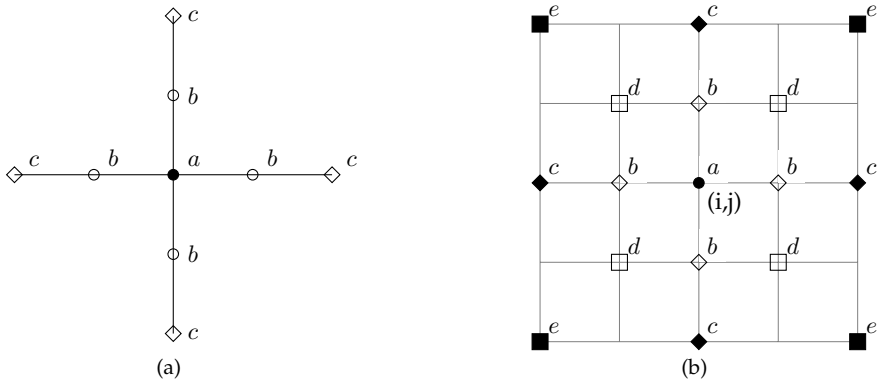


Figure B.2: Illustration of 2-D fourth order Laplacian schemes: the classical "cross" scheme (B.2(a)), and the "square" scheme (B.2(b)). The parameters for the cross scheme are $a = -60/12$, $b = 16/12$ and $c = -1/12$. The parameters for the square scheme are $a = -9/2$, $b = 16/15$, $c = -1/15$, $d = 2/15$ and $e = -1/120$.

direction respectively,

$$\frac{dA(t)}{dt} = -\nu k^2 t\, A(t) \tag{B.3}$$

with $k^2 \triangleq k_1^2 + k_2^2$, and the exact solution is thus given by

$$A(t) = A(0)\, e^{-\nu k^2 t}. \tag{B.4}$$

The numerical eigenvalues associated with a numerical scheme are determined by solving the semi-discrete problem. The temporal evolution of the amplitude of the mode (k_1, k_2) then involves an effective – or numerical – value of k^2, denoted as k_e^2

$$\frac{dA(t)}{dt} = -\nu k_e^2 A(t) \tag{B.5}$$

and thus

$$A(t) = A(0)\, e^{-\nu k_e^2 t}. \tag{B.6}$$

The expressions of $k_e^2 h^2$ for the different schemes are given hereunder.

Second-order cross scheme:

$$k_e^2 h^2 = 4\sin^2\left(\frac{k_1 h}{2}\right) + 4\sin^2\left(\frac{k_2 h}{2}\right) \tag{B.7}$$

Second-order isotropic scheme:

$$k_e^2h^2 = \frac{2}{3}\left(4\sin^2\left(\frac{k_1h}{2}\right) + 4\sin^2\left(\frac{k_2h}{2}\right) + \sin^2\left(\frac{k_1h+k_2h}{2}\right) + \sin^2\left(\frac{k_1h-k_2h}{2}\right)\right) \tag{B.8}$$

Fourth-order cross scheme:

$$k_e^2h^2 = 4\left[\frac{16}{12}\sin^2\left(\frac{k_1h}{2}\right) - \frac{1}{12}\sin^2\left(k_1h\right)\right] + 4\left[\frac{16}{12}\sin^2\left(\frac{k_2h}{2}\right) - \frac{1}{12}\sin^2\left(k_2h\right)\right] \tag{B.9}$$

Fourth-order isotropic scheme:

$$\begin{aligned} k_e^2h^2 = \ & \frac{4}{3}\left(\frac{8}{10}\left(4\sin^2(\frac{k_1h}{2}) + 4\sin^2(\frac{k_2h}{2})\right)\right. \\ & \left. + \frac{2}{10}\left(2\sin^2\left(\frac{k_1h+k_2h}{2}\right) + 2\sin^2\left(\frac{k_1h-k_2h}{2}\right)\right)\right) \\ & - \frac{1}{3}\left(\frac{8}{10}\left(\sin^2(k_1h) + \sin^2(k_2h)\right)\right. \\ & \left. + \frac{2}{10}\left(\frac{1}{2}\sin^2(k_1h+k_2h) + \frac{1}{2}\sin^2(k_1h-k_2h)\right)\right) \end{aligned} \tag{B.10}$$

Iso-contours of $\left(k_e^2h^2\right)/\pi^2$ of the second and fourth-order cross and square schemes are presented in Fig. B.3. It can be observed from Fig. B.3 and also from Eqs. (B.7) to (B.10) that the square schemes are strictly equivalent to their cross counterparts for all the wavenumbers such that $k_1 = 0$ or $k_2 = 0$ (i.e. for the modes aligned with the grid) and that a correction is added for the other directions. As expected, a square scheme is much more isotropic than a cross scheme. Unlike the former, the latter treats the modes that are not aligned with the grid differently. It, however, turns out that the error made on the amplitude of these modes is lower than the error made on the aligned modes, and is minimum for the modes $k_1 = k_2$. As a consequence, the error made on the amplitude using the square scheme is also equal or greater than when using the cross scheme.

First order derivatives

Similar developments have been made for the second and fourth-order finite difference schemes allowing to approximate first order derivatives. They also

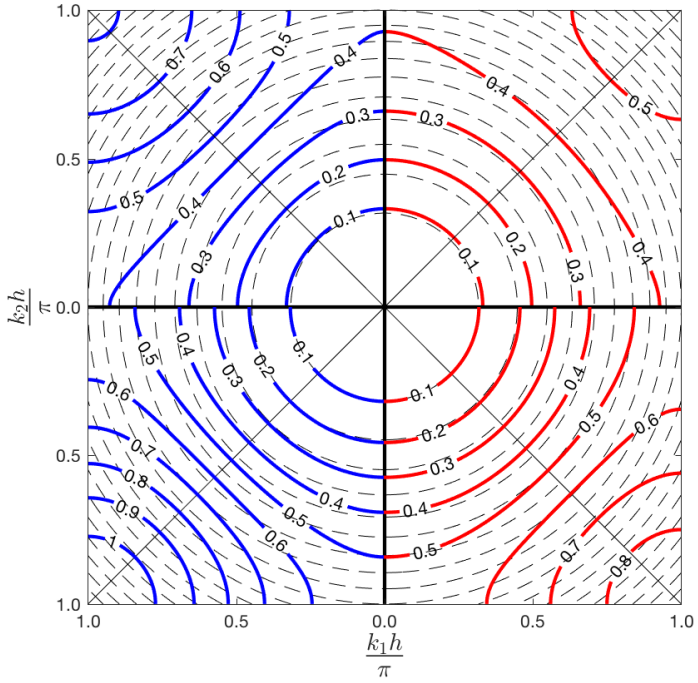


Figure B.3: 2-D diffusion problem: Iso-contours of exact $(kh)^2/\pi^2 = \left((k_1 h)^2 + (k_2 h)^2\right)/\pi^2$ (dashed, by step of 0.1) and of the effective $\left(k_e^2 h^2\right)/\pi^2$ (solid): second order square scheme (upper right quadrant, solid red) and cross scheme (upper left quadrant, solid blue), fourth order cross scheme (lower left quadrant, solid blue) and square scheme (lower right quadrant, solid red).

have the property to be less sensitive to the grid orientation. The schemes are here presented in the x-direction without loss of generality.

We here recall the second-order isotropic scheme or "square" scheme of [91]. The leading order term of the truncation error is required to be a Laplacian

$$\left.\frac{\partial_h f}{\partial_h x}\right|_{i,j} = \left.\frac{\partial f}{\partial x}\right|_{i,j} + Ch^2 \left.\nabla^2\left(\frac{\partial f}{\partial x}\right)\right|_{i,j} + \mathcal{O}\left(h^4\right). \tag{B.11}$$

Using Taylor series, the isotropic scheme is found to be:

$$\begin{aligned}\left.\frac{\partial_h f}{\partial_h x}\right|_{i,j} = \frac{1}{2h}\Big(& \frac{1}{6}\left(f_{i+1,j+1} - f_{i-1,j+1}\right) \\ & + \frac{4}{6}\left(f_{i+1,j} - f_{i-1,j}\right) \\ & + \frac{1}{6}\left(f_{i+1,j-1} - f_{i-1,j-1}\right) \Big)\end{aligned} \tag{B.12}$$

and the leading order term of the truncation error is given by $\left(h^2/6\right)\,\nabla^2\left(\partial f/\partial x\right)\Big|_{i,j}$. Figure B.4 provides the stencil and the coefficients of the resulting second-order "isotropic" scheme as well as those of the classical centered scheme.

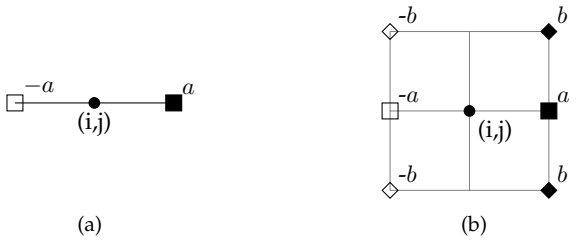


Figure B.4: Illustration of 2-D second order schemes used to evaluate a first order derivative in the x-direction: the classical centered scheme (B.4(a)), and the "square" scheme (B.4(b)). The coefficient a for the classical scheme is $a = 1/2$. The coefficients for the square scheme are $a = 1/3$ and $b = 1/12$.

For the fourth-order isotropic scheme, a similar procedure on a 5x5 stencil is carried out. The scheme is supposed to have a similar form as Eq. (B.12) under the constraint that the leading order term of the truncation error is now a bi-Laplacian of the first order derivative. This leads to the scheme depicted in Fig. B.5. The truncation error is then given by $(-h^4/30)\,\nabla^2\left(\nabla^2\left(\frac{\partial f}{\partial x}\right)\right)\Big|_{i,j} + \mathcal{O}\left(h^6\right)$. The fact that the points $(i \pm 1, j \pm 2)$ and $(i \pm 2, j \pm 1)$ are left unused is a choice. A family of schemes using the 5x5 stencil can be derived. However, as with the fourth-order square Laplacian scheme, it can be shown that the constant in the leading order term of the truncation error remains unchanged for all the members of the family. Other fourth-order isotropic schemes can be found in the literature. In [93], high-order spatial and isotropic finite difference approximations of the 2-D and 3-D gradient operator were presented. In their methodology, the isotropic order I can be controlled for a given spatial order S with $I \geq S$; this capability is also obtained in [131] although the methodology differs.

A modal analysis was performed on the following two-dimensional advection equation in order to assess the performance of the isotropic schemes

$$\frac{\partial u}{\partial t} + c\cos\alpha\frac{\partial u}{\partial x} + c\sin\alpha\frac{\partial u}{\partial y} = 0 \tag{B.13}$$

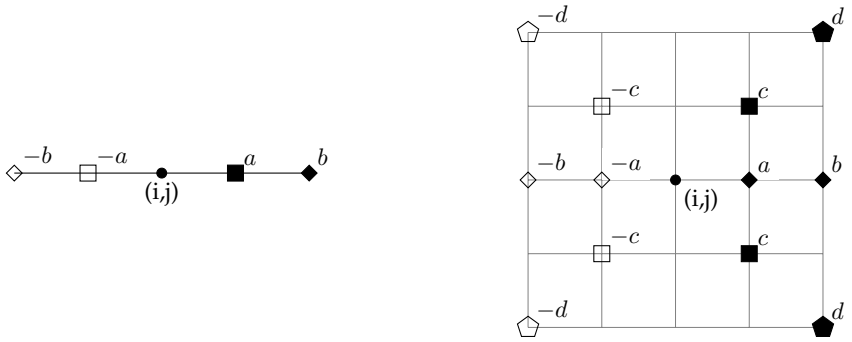


Figure B.5: Illustration of 2-D fourth order schemes used to evaluate a first order derivative in the x-direction: the classical centered scheme (B.5(a)), and the "square" scheme (B.5(b)). The coefficients for classical scheme are $a = 8/12$ and $b = -1/12$. The coefficients for the square scheme are $a = 8/15$, $b = -1/15$, $c = 1/15$, $d = -1/120$.

with c a uniform advection velocity and α the angle of propagation. A similar analysis was performed to study other FD schemes (e.g. Sescu [141], Lele [94] and the review of Colonius and Lele [35] where standard explicit FD schemes, compact FD schemes and optimized FD schemes are compared). Consider a mode $u(x, y, t) = A(t) \exp\left(i\left(k_1 x + k_2 y\right)\right)$ with $k_1 = k\cos\alpha$ and $k_2 = k\sin\alpha$, the wave numbers in the x- and y-direction respectively and k the wavenumber magnitude ($k^2 \triangleq k_1^2 + k_2^2$). The exact solution is given by

$$u(x, y, t) = A(0) e^{ik\left(cos(\alpha)x + sin(\alpha)y - ct\right)}. \tag{B.14}$$

The semi-discretization of Eq. (B.13) yields the following expression for the numerical solution

$$u_{num}(x_i, y_j, t) = A(0) e^{ik\left(cos(\alpha)x_i + sin(\alpha)y_j - \frac{k^* h}{kh} ct\right)} \tag{B.15}$$

with k^* the modified wavenumber associated to the numerical scheme. The expressions of k^*h/kh for the considered schemes are given hereunder.

Second-order cross scheme:

$$\frac{k^* h}{kh} = \cos\alpha \left(\frac{\sin(k_1 h)}{kh}\right) + \sin\alpha \left(\frac{\sin(k_2 h)}{kh}\right) \tag{B.16}$$

Second-order isotropic scheme:

$$\frac{k^*h}{kh} = \cos\alpha \left[\frac{2}{3}\frac{\sin(k_1h)}{kh} + \frac{1}{3}\frac{\sin(k_1h)\cos(k_2h)}{kh}\right] + \sin\alpha \left[\frac{2}{3}\frac{\sin(k_2h)}{kh} + \frac{1}{3}\frac{\sin(k_2h)\cos(k_1h)}{kh}\right] \tag{B.17}$$

Fourth-order cross scheme:

$$\frac{k^*h}{kh} = \cos\alpha \left[\frac{4}{3}\frac{\sin(k_1h)}{kh} - \frac{1}{3}\frac{\sin(2k_1h)}{2kh}\right] + \sin\alpha \left[\frac{4}{3}\frac{\sin(k_2h)}{kh} - \frac{1}{3}\frac{\sin(2k_2h)}{2kh}\right] \tag{B.18}$$

Fourth-order isotropic scheme:

$$\begin{aligned}\frac{k^*h}{kh} = \cos\alpha &\left[\frac{16}{15}\frac{\sin(k_1h)}{kh} - \frac{4}{15}\frac{\sin(2k_1h)}{2kh} + \right. \\ &\left. \frac{4}{15}\frac{\sin(k_1h)\cos(k_2h)}{kh} - \frac{1}{15}\frac{\sin(2k_1h)\cos(2k_2h)}{2kh}\right] + \\ \sin\alpha &\left[\frac{16}{15}\frac{\sin(k_2h)}{kh} - \frac{4}{15}\frac{\sin(2k_2h)}{2kh} + \right. \\ &\left. \frac{4}{15}\frac{\sin(k_2h)\cos(k_1h)}{kh} - \frac{1}{15}\frac{\sin(2k_2h)\cos(2k_1h)}{2kh}\right]\end{aligned} \tag{B.19}$$

Figure B.6 represents the modified wavenumber as a function of the angle of propagation α for different value of kh – or equivalently for different number of points per wavelength – in a polar diagram as in [94, 35, 141]; the exact solution being the unit circle. The same conclusions as those made here above for the Laplacian schemes can be drawn from Fig. B.6.

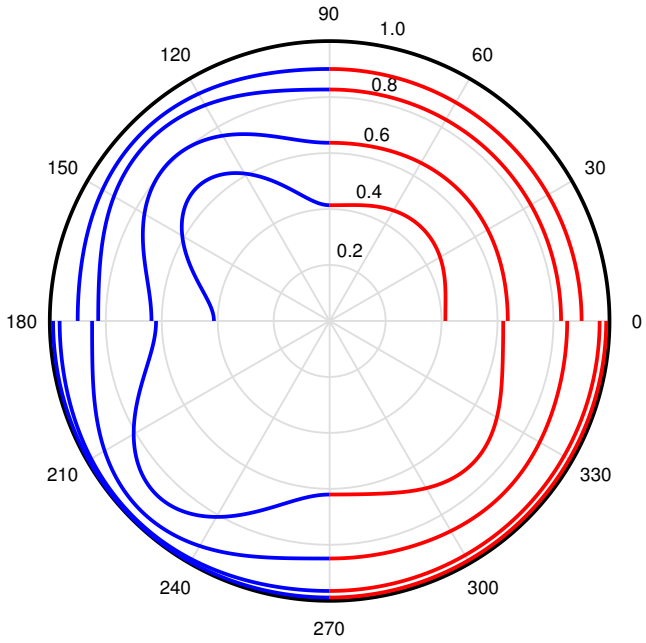


Figure B.6: 2-D advection problem: Polar diagram of the ratio between the modified wavenumbers and the wavenumber, $(k^*h)/kh$ as a function of the propagation angle α for different numbers of points per wavelength (3, 4, 6 and 8); second order square scheme (upper right quadrant, solid red) and cross scheme (upper left quadrant, solid blue), fourth order cross scheme (lower left quadrant, solid blue) and square scheme (lower right quadrant, solid red). The exact solution is the unit circle.

APPENDIX C

NUMERICAL DISPERSION AND DISSIPATION ERRORS OF A 1-D REDISTRIBUTED AND FILTERED LAGRANGIAN METHOD

This section is devoted to the comparison of the performance of the redistribution scheme of Eldredge et al. [50] and the performance of the M_6^* redistribution kernel designed in [11] when it is used in combination with a discrete filter. Their numerical dispersion and dissipation errors will be compared. To that purpose, a linear advection equation is considered and the study is performed in Fourier space. The methodology derived and detailed in [102] (see also references therein) is here directly applied to study the performance of the redistribution scheme designed by Eldredge et al. [50]. A straightforward addition to this methodology allows to study the performance of a redistribution scheme combined with a discrete filter. The methodology could also be applied to the case of a continuous filter. It would only require an additional sampling of the filter in real space. The derivation of the numerical dissipation and dispersion error is described in Section C.1. The performance study is presented in Section C.2.

C.1 Contribution of the filtering to the numerical dissipation and dispersion errors

In this section, the main results derived in Marichal's PhD thesis [102] are recalled and extended to the case of a 1-D redistributed and filtered Lagrangian method. In [102], a one-dimensional linear advection problem in a periodic domain is considered

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{C.1}$$

with c a uniform advection velocity. In infinite or periodic domain, the solution of this equation is

$$u(x,t) = U_0(x - ct) \tag{C.2}$$

where $U_0(x) \triangleq u(x,0)$ is the initial condition. In this study, a periodic domain is considered, $x \in \Omega \triangleq [0, L]$. The Lagrangian discretization of this problem using N particles is then given by

$$\frac{dx_p}{dt} = c\,, \qquad x_p(0) = ph \tag{C.3}$$

$$\frac{du_p}{dt} = 0\,, \qquad u_p(0) = u(ph, 0) \tag{C.4}$$

with $h = L/N$, the mesh spacing, $x_p(t)$ $(p = 0, ..., N-1)$, the position of the particles, and $u_p(t) \triangleq u(x_p(t), t)$, the function value carried by the particle p. As stated in [102], this discretization is equivalent to the method of characteristics and the exact solution can be obtained using an explicit Euler integration:

$$\begin{aligned} x_p^{n+1} &= x_p^n + c\Delta t \\ u_p^{n+1} &= u_p^n \end{aligned}$$

where $x_p^n \triangleq x_p(t^n), u_p^n \triangleq u_p(t^n)$ with Δt the time step and n the time step index ($t^{n+1} \triangleq t^n + \Delta t$).

The particles are redistributed every n_r time steps. The set of distorted particles x_q^n carrying the function value u_q^n is then replaced by a set of new particles lying at the grid nodes $x_p^{n,r} \triangleq ph$ and whose function value is computed using a redistribution kernel. Without loss of generality, $n_r = 1$ and thus $\Delta t^r \triangleq n_r \Delta t = \Delta t$ was used as in [102]. The redistribution period was then related to the number of cells crossed by the solution before being redistributed through a CFL-like number defined as

$$\beta \triangleq \frac{c\Delta t}{h} \tag{C.5}$$

The sequence of operations studied in [102] is the following:

1. advection of the particles from $x_q^n = qh$ to $x_q^{n+1} = (\beta + q)h$

2. redistribution of the particles: the redistributed particle positions are denoted as $x_p^{n+1,r} = ph$ and the function value they carry is computed using the chosen and periodized redistribution scheme:

$$u_p^{n+1,r} = \sum_{q=0}^{N-1} u_q^{n+1} v_{per}\left(x_p^{n+1,r} - x_q^{n+1}\right) \tag{C.6}$$

   with $v_{per}(x) \triangleq \sum_{m=-\infty}^{+\infty} v(x - mL)$ the periodized redistribution kernel based on the interpolation scheme $v(x) \triangleq w(x/h)$.

In the present extension, an additional filtering operation is carried out when a classical redistribution kernel is used (a M_6^* redistribution kernel, for example). The remeshed particles are then filtered every $n_f = \alpha n_r$ time steps. Conversely, when using the kernel derived in [50], some degree of smoothing is already included in the redistribution kernel itself and some filtering is then performed during the redistribution step (step 2 here above). The filtering frequency is thus equal to redistribution frequency ($n_f = n_r$). As a consequence, the filtering frequency will also be chosen such that $n_f = n_r$ in the following; we stress that the redistribution and filtering operation can be dissociated when using a redistribution kernel without integrated smoothing.

3. (optional) filtering operation: the function value $u_p^{n+1,r}$ carried by the redistributed particles is filtered using a periodized filtering operator:

$$\overline{u_p^{n+1,r}} = \sum_{q=0}^{N-1} u_q^{n+1,r} f_{per}\left(x_p^{n+1,r} - x_q^{n+1,r}\right) \tag{C.7}$$

   with $f_{per}(x) \triangleq \sum_{m=-\infty}^{+\infty} v(x - mL)$ the periodized filtering operator based on the filtering operator $f(x) \triangleq g(x/\bar{\Delta})$ with $\bar{\Delta}$ the filter width.

The following notation was then introduced in [102]:

- $f[\cdot]$ stands for a discrete function and has an integer as the argument
- $f(\cdot)$ stands for a continuous function and has a real value as the argument

The function values $u^0[p] \triangleq u_p^0$, $u^{1,r}[p] \triangleq u_p^{1,r}$ and its filtered counterpart $\overline{u^{1,r}}[p] \triangleq \overline{u_p^{1,r}}$ are then discrete functions of the particle index p lying on the

grid $x_p \triangleq ph$. Following [102], Equation (C.6) is then rewritten as a discrete convolution on the grid x_p:

$$u^{1,r}[p] = \sum_{q=0}^{N-1} u^0[q] v^{\beta}_{per}[p-q] \tag{C.8}$$

with $v^{\beta}_{per}[p] \triangleq v_{per}((p-\beta)h)$ the discretized and periodized redistribution scheme that has been shifted in space by a distance βh beforehand. The Discrete-Time Fourier Series (DTFS) of this convolution is then

$$\tilde{U}^{1,r}[l] = N\tilde{V}^{\beta}_{per}[l]\tilde{U}^0[l] = \tilde{\rho}^r[l]\tilde{U}^0[l] \qquad l = 0, 1, \ldots, N-1 \tag{C.9}$$

with $\tilde{\rho}^r[l] \triangleq N\tilde{V}^{\beta}_{per}[l]$, the amplification factor due to the advection and the redistribution.

Using this notation, Eq. (C.7) can be rewritten as a discrete convolution on the grid x_p as well:

$$\widetilde{\overline{U}}^{1,r}[l] = N\tilde{F}_{per}[l]\tilde{U}^{1,r}[l] = \tilde{\rho}^f[l]\tilde{\rho}^r[l]\tilde{U}^0[l] = \tilde{\rho}^{r,f}[l]\tilde{U}^0[l]$$
$$l = 0, 1, \ldots, N-1 \tag{C.10}$$

with $\tilde{\rho}^f[l] \triangleq N\tilde{F}_{per}[l]$ the amplification factor due to the filtering operation. In the general (i.e. continuous) case, the operations required to obtain the $\tilde{F}_{per}[l]$ as a function of the Fourier Transform of the continuous filter, $\hat{F}(k) \triangleq \text{FT}\{f(x)\} = \bar{\Delta}\mathcal{G}(k\bar{\Delta})$, are identical to those performed in [102] to obtain $\tilde{V}^{\beta}_{per}[l]$ as a function of $\hat{V}(k) \triangleq \text{FT}\{v(x)\}$ (the shift operation is however unnecessary and removed). When using a discrete filter as in the rest of this work, the DTFT (Discrete-Time Fourier Transform) of the filter, $\tilde{F}(K)$, can be directly used and periodized. The periodization of the filter f at a rate N leads to the sampling at a rate $\Delta K = 2\pi/N$ or $K = l\Delta K = 2\pi l/N$ of the spectrum $\tilde{F}(K)$. The DFTS of $\tilde{F}_{per}[l]$ is then related to the periodized DTFT of the discrete filter by

$$\tilde{F}_{per}[l] = \frac{1}{N}\tilde{F}\left(\frac{2\pi l}{N}\right) \tag{C.11}$$

and the amplification factor of the filtering operation is, as anticipated,

$$\tilde{\rho}^f[l] = \tilde{F}\left(\frac{2\pi l}{N}\right) \qquad l = 0, 1, \ldots, N-1 \tag{C.12}$$

while the amplification factor due to the advection and the redistribution operation is (see [102]),

$$\tilde{\rho}^r[l] = \exp\left(-\frac{j2\pi l\beta}{N}\right) \sum_{m=-\infty}^{+\infty} \exp(j2\pi\beta m)\hat{W}\left(2\pi\left(\frac{l}{N} - m\right)\right) \tag{C.13}$$

with $\hat{W}(k)$ the Fourier transform of the interpolation kernel $w(\xi)$. The exact amplification factor (i.e. the amplification factor due to the advection) is included in Eq. (C.13) and is

$$\tilde{\rho}[l] = \exp(-jk_l c\Delta t) = \exp(-jk_l \beta h) = \exp\left(-\frac{j2\pi l\beta}{N}\right) \tag{C.14}$$

with $k_l = 2\pi l/L$ the physical wavenumber. Equation (C.13) can thus be rewritten as $\tilde{\rho}^r[l] = \tilde{\rho}[l]\tilde{\gamma}^r[l]$. The exact amplication factor is such that $|\tilde{\rho}[l]| = 1$ and its phase is given by

$$\phi[l] \triangleq \arg(\tilde{\rho}[l]) = -2\pi l\beta/N = -k_l h\beta \tag{C.15}$$

The amplification factor amplitude of the whole sequence of operations is thus given by

$$|\tilde{\rho}^{r,f}[l]| = |\tilde{\gamma}^r[l]||\tilde{\rho}^f[l]| \tag{C.16}$$

while its phase is, by definition,

$$\phi^{r,f}[l] = \phi^r[l] + \phi^f[l] = \phi[l] + \arg(\tilde{\gamma}^r[l]) + \arg(\tilde{\rho}^f[l]) \tag{C.17}$$

with $\phi^f[l] = 0$ for a symmetric filter as considered hereunder.

Following [102], the dispersion error is rewritten as a modified wavenumber k^*:

$$\begin{aligned}
\overline{u}^r(x,\Delta t) &= \sum_{l=-N/2+1}^{N/2} \widetilde{\overline{U}}^{1,r}[l]\exp(jk_l x) \\
&= \sum_{l=-N/2+1}^{N/2} \tilde{U}^0[l]|\tilde{\rho}^f[l]||\tilde{\rho}^r[l]|\exp\left(j\left(\phi^r[l]+\phi^l[l]\right)\right)\exp(jk_l x) \\
&= \sum_{l=-N/2+1}^{N/2} \tilde{U}^0[l]|\tilde{\rho}^f[l]||\tilde{\rho}^r[l]|\exp\left(j\left(k_l x - k_l^* c\Delta t\right)\right)
\end{aligned}$$

The dispersion error is then also given by

$$k_l^* h = -\frac{\left(\phi^r[l]+\phi^f[l]\right)}{\beta} \tag{C.18}$$

The dissipation error is also recast as an effective diffusion coefficient ϵ^*, which can obviouly be decomposed in two contributions; one due the redi-

stribution and the other one due to the filtering:

$$\begin{aligned} |\tilde{\rho}^f[l]||\tilde{\rho}^r[l]| &\triangleq \exp(-\epsilon^* k_l^2 \Delta t) \\ \frac{\epsilon_l^*}{ch} &= -\frac{\log\left(|\tilde{\rho}^f[l]||\tilde{\rho}^r[l]|\right)}{\beta(k_l h)^2} \\ &= -\left(\frac{\epsilon_l^{*,f}}{ch} + \frac{\epsilon_l^{*,r}}{ch}\right) \end{aligned} \tag{C.19}$$

As stated in [102], every redistributed, and here filtered, mode $u_l^r(x, \Delta t) \triangleq \widetilde{\overline{U}}^{1,r}[l] \exp(jk_l x)$ can be seen as the solution of a modified equation corresponding to a advection-diffusion problem:

$$\frac{\partial u}{\partial t} + c\left(\frac{k_l^* h}{k_l h}\right)\frac{\partial u}{\partial x} = \epsilon_l^* \frac{\partial^2 u}{\partial x^2}. \tag{C.20}$$

The Fourier transform of various redistribution schemes (M_1, M_2, M_3 and M_4' kernels) are recalled in [102]. We here provide the Fourier transform of the M_6^* redistribution scheme and of the E_6 scheme (see Eq. (2.27) in Chapter 2)

$$FT\{M_6^*\}(k) = -\frac{2(\cos(k)-1)^2\left((100-7k^2)(\cos(k)-1)+48k\sin(k)\right)}{k^6} \tag{C.21}$$

$$FT\{W_E\}(k) = \frac{1}{32}\left[\left(\frac{\sigma}{h}k\right)^4 + 8\left(\frac{\sigma}{h}k\right)^2 + 32\right]\exp\left(-\frac{1}{4}\left(\frac{\sigma}{h}k\right)^2\right) \tag{C.22}$$

We also recall the Fourier transform of the explicit discrete filter of order $2n$ (Eq. (2.31)), n being the number of iterations performed,

$$\mathcal{G}^{(n)}(kh) = 1 - \left(\sin^2\left(kh/2\right)\right)^n \tag{C.23}$$

C.2 Performance of the E_6 scheme and of the combination M_6^* scheme - discrete filter

Figures C.1(a) and C.1(b) compare the modified wavenumber k^*h for the sixth-order interpolation kernel introduced by Eldredge (here denoted as E_6) and the M_6^* redistribution scheme[1] . It can be observed that the E_6 scheme exhibits less dispersion error than the M_6^* scheme for all values of the parameter β. As

[1]Let us recall that the index in the denomination E_6 here refers to the order of the redistribution kernel while the index in M_6^* refers to the width of the kernel.

already stated, this observation is also true when performing a redistribution-filtering step using the M_6^* scheme combined with a filtering operator.

Figure C.2 show the amplitude of the amplification factor for both schemes. By construction, the E_6 scheme has a built-in diffusive or smoothing behavior. An interesting property of the E_6 scheme can be identified by inspecting Fig. C.1 and Fig. C.2: the dispersed modes (starting from $\frac{kh}{\pi} \simeq 0.8$) are also almost entirely dissipated. The E_6 scheme thus operates with virtually no dispersion of the modes $\frac{kh}{\pi} \lesssim 0.8$. As already stated by Eldredge et al. [50], the remeshing-filtering frequency must be increased when the mesh is refined, as the smoothing decreases. This comment is illustrated in Figure C.2 as the axis is the product of the wavenumber, k, with the mesh spacing, h.

As explained in Chapter 2, an alternative to the E_6 scheme is to use the M_6^* scheme in combination with an iterated explicit discrete filter. Using the developments made in Section C.1, the number of iterations can be chosen so as to provide a redistribution-filtering operation having similar smoothing properties as the E_6 scheme. Figure C.3 shows that the explicit discrete filter iterated four times (and thus of order 8) fulfill that function[2]. As explained in Section 2.2.4, this combination offers some advantages:

- the full procedure of remeshing and filtering is here conservative, which is not formally the case when using the E_6 scheme with a finite-width stencil. This should however not be seen as an eliminatory comment as the departure rapidly decreases with the stencil width.
- the M_6^* scheme used in a hybrid VPM method for incompressible flows was shown in [156] to provide results in quantitive agreement with those provided by a pseudo-spectral method.
- the redistribution and the filtering are separate operations. They can thus be applied at a frequency tailored to meet their needs. This offers more flexibility.

These advantages are however associated to a higher computational cost as the filtering needs to be iterated.

In the light of the above, and of the excellent dispersion properties of the E_6 scheme in particular, the choice made in this work to use the M_6^* scheme

[2]The filter of order 6 is even closer to the values obtained with the E_6 scheme but we here elect to use a filter being a bit less diffusive.

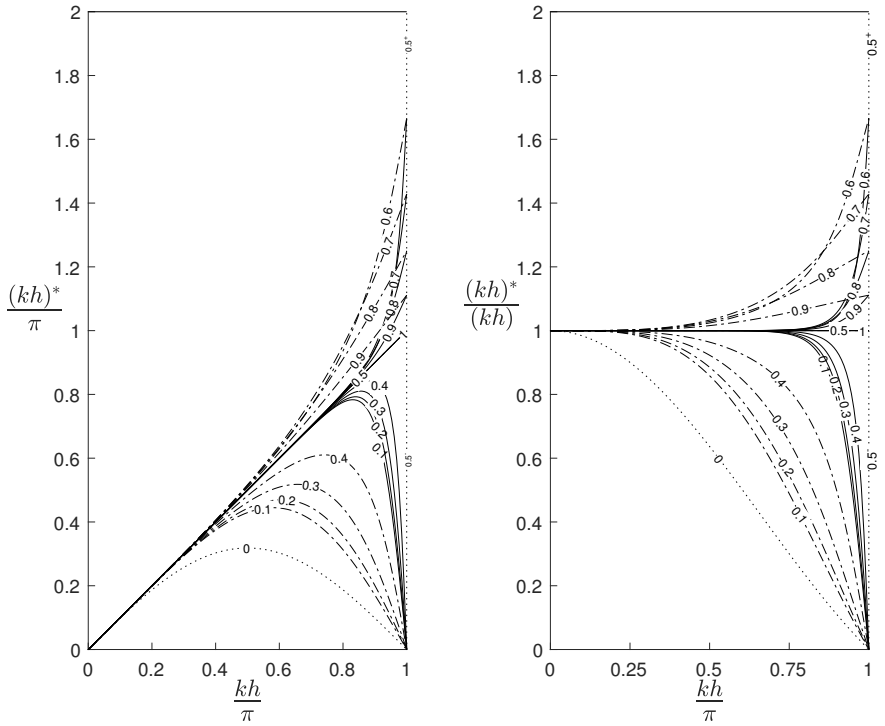


Figure C.1: Numerical dispersion error for the E_6 redistribution scheme [50] (solid lines) and for the M_6^* redistribution scheme (dashed lines) for different values of the parameter β.

combined with a discrete filter of order $2n$, with $n = 3$ or 4), rather than the E_6 scheme, should not be seen as definite.

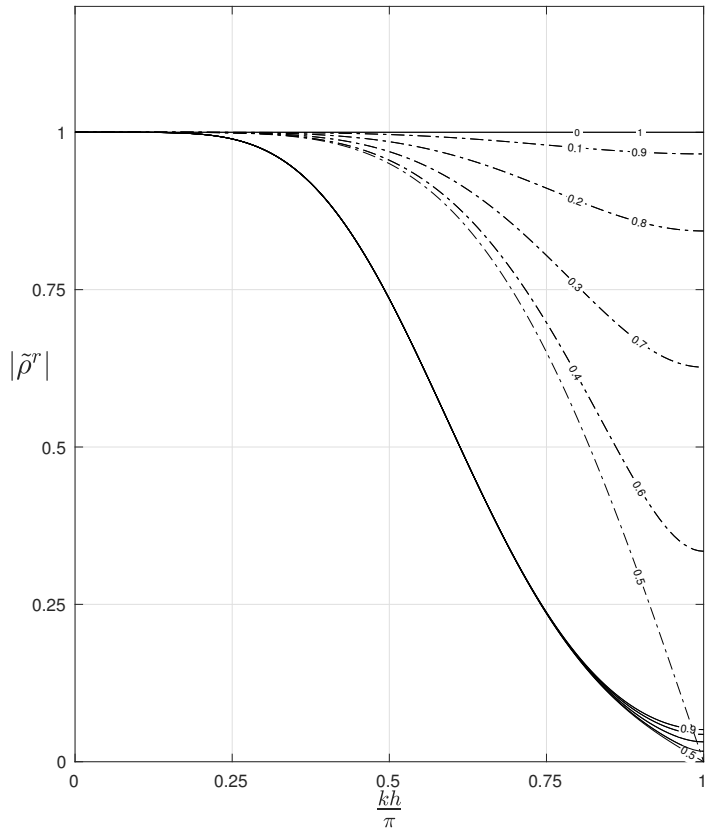


Figure C.2: Numerical diffusion introduced by the E_6 redistribution scheme [50] (solid lines) and by the M_6^* redistribution scheme (dashed lines).

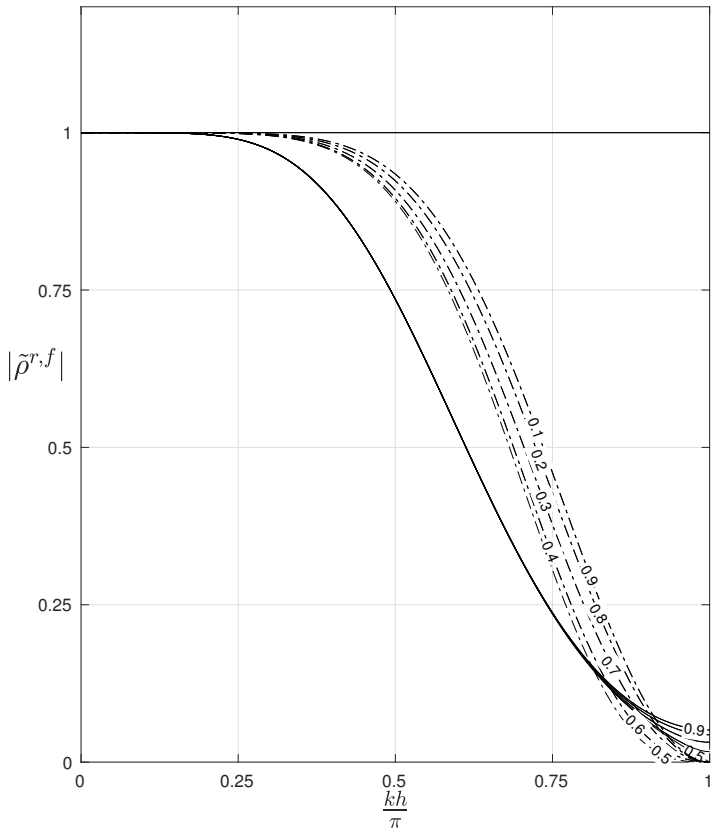


Figure C.3: Numerical diffusion introduced by the E_6 redistribution scheme [50] (solid lines) and by the M_6^* redistribution scheme combined with the explicit discrete filter of order 8 (dashed lines).

APPENDIX D

ANALYTICAL EXPRESSION FOR THE DIVERGENCE OF THE VELOCITY FIELD FOR THE ACOUSTIC PULSE PROBLEM

In [150], Tam and Webb derived analytical solutions of the Linearized Euler Equations applied to the problem where wave disturbances are generated by initial excitations in the presence of an uniform mean flow of Mach number M. We here concentrate on the problem of acoustic waves generated by a single initial Gaussian pressure distribution as considered in Section 2.3.1. The initial condition is given by Eq. (2.33). In that case, the dimensionless analytical solution derived by Tam and Webb read as

$$u\left(x,y,t\right) = \frac{\epsilon_1\left(x-Mt\right)}{2\alpha_1\eta}\int_0^\infty e^{-\frac{\xi^2}{4\alpha_1}}\sin\left(\xi t\right)J_1\left(\xi\eta\right)\xi d\xi \tag{D.1}$$

$$v\left(x,y,t\right) = \frac{\epsilon_1 y}{2\alpha_1\eta}\int_0^\infty e^{-\frac{\xi^2}{4\alpha_1}}\sin\left(\xi t\right)J_1\left(\xi\eta\right)\xi d\xi \tag{D.2}$$

$$p(x,y,t) = \rho(x,y,t) = \frac{\epsilon_1}{2\alpha_1}\int_0^\infty e^{-\frac{\xi^2}{4\alpha_1}}\cos\left(\xi t\right)J_0\left(\xi\eta\right)\xi d\xi \tag{D.3}$$

where $\eta = \left(\left(x-Mt\right)^2+y^2\right)^{1/2}$ and where J_0 and J_1 are the Bessel functions of order zero and one, respectively. The parameter α_1 is defined as $\alpha_1 = \frac{log(2)}{b^2}$, with b the half-width of the Gaussian. The purpose of this appendix is to

provide the developments leading to the expression for the divergence of the velocity field. These developments will rely on the following property of the Bessel functions:

$$\frac{d}{dx}\left(x^n J_n(x)\right) = x^n J_{n-1}(x) \tag{D.4}$$

By definition of the divergence of the velocity field, the first step consists in evaluating the spatial derivative of Eq. (D.1) with respect to x. Let's first group the terms that are independent of x,

$$a = \frac{\epsilon_1}{2\alpha_1} \tag{D.5}$$

$$b(\xi) = e^{-\frac{\xi^2}{4\alpha_1}} \sin(\xi t) \tag{D.6}$$

Equation D.1 then reads,

$$u(x,y,t) = a\left(\frac{x-Mt}{\eta}\right)\int_0^\infty b(\xi) J_1(\xi\eta)\xi d\xi \tag{D.7}$$

$$= a\left(\frac{x-Mt}{\eta^2}\right)\int_0^\infty b(\xi) J_1(\xi\eta)(\xi\eta) d\xi \tag{D.8}$$

$$= \underbrace{a\left(\frac{x-Mt}{(x-Mt)^2+y^2}\right)}_{h(x,t)} \underbrace{\int_0^\infty b(\xi)\, \underbrace{J_1(\xi\eta)(\xi\eta)}_{f(\xi\eta)}\, d\xi}_{g(x,t,\xi)} \tag{D.9}$$

We then obtain,

$$\frac{\partial h}{\partial x} = a\left(\frac{(x-Mt)^2 + y^2 - 2(x-Mt)^2}{((x-Mt)^2+y^2)^2}\right) \tag{D.10}$$

$$= \frac{\epsilon_1}{2\alpha_1}\left(\frac{y^2-(x-Mt)^2}{\eta^4}\right) \tag{D.11}$$

Using Eq. (D.4) and

$$\frac{\partial f}{\partial x} = \frac{\partial(\xi\eta)}{\partial x}\frac{\partial f}{\partial(\xi\eta)} = \frac{\partial}{\partial x}\left(\xi\sqrt{(x-Mt)^2+y^2}\right)\frac{\partial f}{\partial(\xi\eta)} \tag{D.12}$$

$$= \xi\left(\frac{x-Mt}{\eta}\right)\frac{\partial f}{\partial(\xi\eta)} \tag{D.13}$$

it follows that

$$\frac{\partial g}{\partial x} = \int_0^\infty b(\xi)\frac{\partial f}{\partial x}d\xi \tag{D.14}$$

$$= \left(\frac{x-Mt}{\eta}\right)\int_0^\infty b(\xi)\frac{\partial f}{\partial(\xi\eta)}\xi d\xi \tag{D.15}$$

$$= \left(\frac{x-Mt}{\eta}\right)\int_0^\infty b(\xi)\underbrace{\frac{\partial}{\partial(\xi\eta)}\left((\xi\eta)J_1(\xi\eta)\right)}_{(\xi\eta)J_0(\xi\eta)}\xi d\xi \tag{D.16}$$

$$= (x-Mt)\int_0^\infty b(\xi)J_0(\xi\eta)\xi^2 d\xi \tag{D.17}$$

Using the product rule, the derivative of u with respect to x is given by

$$\frac{\partial u}{\partial x} = \frac{\partial h}{\partial x}g + h\frac{\partial g}{\partial x} \tag{D.18}$$

$$= \frac{\epsilon_1}{2\alpha_1}\left(\frac{y^2-(x-Mt)^2}{\eta^3}\right)\int_0^\infty b(\xi)J_1(\xi\eta)\xi d\xi + \frac{\epsilon_1}{2\alpha_1}\left(\frac{x-Mt}{\eta}\right)^2\int_0^\infty b(\xi)J_0(\xi\eta)\xi^2 d\xi \tag{D.19}$$

Similar developments are made to evaluate the derivative with respect to y of the vertical component of the velocity field, Eq. (D.2), which leads to

$$\frac{\partial v}{\partial y} = \frac{\epsilon_1}{2\alpha_1}\left(\frac{(x-Mt)^2-y^2}{\eta^3}\right)\int_0^\infty b(\xi)J_1(\xi\eta)\xi d\xi + \frac{\epsilon_1}{2\alpha_1}\left(\frac{y}{\eta}\right)^2\int_0^\infty b(\xi)J_0(\xi\eta)\xi^2 d\xi \tag{D.20}$$

When evaluating the divergence of the velocity field, the first term of Eq. (D.19) and Eq. (D.20) cancel, which leads to:

$$\theta = \frac{\epsilon_1}{2\alpha_1}\int_0^\infty e^{-\frac{\xi^2}{4\alpha_1}}\sin(\xi t)\,J_0(\xi\eta)\xi^2 d\xi \tag{D.21}$$

This integral was numerically evaluated in order to provide reference values for the rms. and maximum value of the dilatation field in Section 2.3.1.

APPENDIX E

ENFORCEMENT OF GLOBAL CONSERVATION ON THE CORRECTION

A conservative correction procedure would be such that,

$$\sum_{p}^{N_p^E} \alpha_p = \sum_{p}^{N_p^E} \bar{f}_E^p S_E^p \tag{E.1}$$

where

$$\sum_{p}^{N_p^E} \alpha_p \triangleq \sum_{p}^{N_p^E} \bar{f}_p S_p \tag{E.2}$$

and

$$\sum_{p}^{N_p^E} \bar{f}_E^p S_E^p \triangleq \sum_{p}^{N_p^E} \left(\sum_{se}^{n_{se}^p} \int_{se} f_E dS \right) \tag{E.3}$$

where n_{se}^p is the number of subelements se whose integrals contributes to the strength of the particle p, the surface of those subelements being $S_E^p \triangleq \sum_{se}^{n_{se}^p} \int_{se} dS$, and N_p^E is the number of particles in the Eulerian subdomain. In other words, the integral of the Eulerian solution vector on the Lagrangian particles should be identical to the integral of the solution vector on the Eulerian mesh. It is however not ensured in the correction procedure described in Section 4.4.2. In [115], Palha et al. also used a correction to enforce the conservation of the total circulation in the Eulerian subdomain in the context of the

coupling of two incompressible solvers. They proposed to use an "additive correction" of the type

$$\alpha_{p,cons.} = \alpha_p + \Delta\alpha_p = \alpha_p + \frac{1}{N_p^I}\left(\int_{\Omega_I} f_E dS - \sum_p^{N_p^I} \alpha_p\right) \tag{E.4}$$

with $\Omega_I \subset \Omega_E$, a subset of the Eulerian subdomain excluding a narrow band around the body and a band close to the Eulerian exterior boundary, and N_p^I the number of particles in this subset. However, this technique introduces some circulation everywhere in the Eulerian subdomain, which is not physical.

This undesirable behavior can be avoided by working with a correction that is proportional to the deviation of α_p with respect to its freestream value, $\alpha_p' = \alpha_p - \alpha_{p,\infty}$. The following correction on the particles in the Eulerian subdomain Ω_E could be used to enforce the conservation,

$$\alpha_{p,cons.}^{'} = \alpha_p' + \frac{(\alpha_p')^2}{\left(\sum_p^{N_p^E} (\alpha_p')^2\right)}\left(\sum_p^{N_p^E} \bar{f}_E^p S_E^p - \sum_p^{N_p^E} \alpha_p\right) \tag{E.5}$$

and thus,

$$\alpha_{p,cons.} = \alpha_{p,\infty} + \alpha_{p,cons.}^{'} \tag{E.6}$$

It can be easily shown that this correction is equivalent to the following constrained minimization problem:

$$\min \quad \sum_p^{N_p^E} \left(\frac{\alpha_{p,cons.}^{'} - \alpha_p'}{\alpha_p'}\right)^2 \tag{E.7}$$

$$\text{subject to} \quad \sum_p^{N_p^E} \alpha_{p,cons.} - \sum_p^{N_p^E} \bar{f}_E^p S_E^p = 0 \tag{E.8}$$

By working with the deviation $\alpha_p^{'}$ rather than with α_p, the formulation has the advantage of avoiding to correct particles in the Eulerian subdomain that carries the freestream intensities. Another admissible possibility would have been a "multiplicative correction" on the deviation from the freestream state, i.e,

$$\alpha_{p,cons.}^{'} = \beta\alpha_p^{'} \tag{E.9}$$

and,

$$\alpha_{p,cons.} = \alpha_{p,\infty} + \alpha_{p,cons.}^{'} \tag{E.10}$$

the coefficient β being computed such that

$$\sum_{p}^{N_p^E}(\alpha_{p,\infty} + \beta\alpha_p^{'}) = \sum_{p}^{N_p^E} \bar{f}_E^p S_E^p \tag{E.11}$$

leading to,

$$\beta = \frac{\left(\sum_p^{N_p^E} \bar{f}_E^p S_E^p - N_p^E \alpha_{p,\infty}\right)}{\sum_p^{N_p^E} \alpha_p^{'}} \tag{E.12}$$

Unfortunately, none of the variants could enforce the conservation, Eq. (E.1), without noticeably interfering with the local behavior of the corresponding field.

BIBLIOGRAPHY

[1] Ann S. Almgren, Thomas Buttke, and Phillip Colella. A fast Adaptive Vortex Method in Three Dimensions. *J. Comput. Phys.*, 113(2):177–200, aug 1994.

[2] Christopher R Anderson. A method of local corrections for computing the velocity field due to a distribution of vortex blobs. *Journal of Computational Physics*, 62(1):111–123, 1986.

[3] John D. Anderson, Jr. *Fundamentals of aerodynamics*. Mcgraw-Hill, fifth si edition edition, 2011.

[4] P. Angot. Analysis of singular perturbations on the brinkman problem for fictitious domain models of viscous flows. *Mathematical Methods in the Applied Sciences*, 22(16):1395–1412, 1999.

[5] P. Angot, C. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81:497–520, 1999 1999.

[6] E. Arquis and J. P. Caltagirone. On the hydrodynamical boundary-conditions along a fluid layer porous-medium interface - Application to the case of free convection. *Comptes Rendus de l Academie des Sciences Serie II*, 299(1):1–4, 1984.

[7] Stéphane Backaert, Philippe Chatelain, and Grégoire Winckelmans. Vortex Particle-Mesh with Immersed Lifting Lines for Aerospace and Wind Engineering. In Jens Honoré Walther, editor, *Symposium on Particle Methods in Fluid Dynamics*, volume 18 of *Procedia IUTAM*, pages 1–7, DTU Copenhagen, October 2012. IUTAM.

[8] Stéphane Backaert, Philippe Chatelain, Grégoire Winckelmans, and Ivan De Visscher. Vortex Particle-Mesh simulations of atmospheric turbulence effects on wind turbine blade loading and wake dynamics.

In Michael Hölling, Joachim Peinke, and Stefan Ivanell, editors, *EUROMECH Colloquium 528 (Joint colloquium with ERCOFTAC and EAWE) ``Wind energy and the impact of turbulence on the conversion process''*, volume 2 of *Research Topics in Wind Energy*, Oldenburg, Germany, Feb. 22-24 2014. Springer-Verlag, Berlin Heidelberg.

[9] Gregory T. Balls and Phillip Colella. A finite Difference Domain Decomposition Method Using Local Corrections for the Solution of Poisson's Equation. *Journal of Computational Physics*, 180(1):25 – 53, 2002.

[10] M. Bar-Lev and H. T. Yang. Initial flow field over an impulsively started circular cylinder. *Journal of Fluid Mechanics*, 72:625–647, 12 1975.

[11] M. Bergdorf. *Multiresolution Particle Methods for the Simulation of Growth and Flow*. PhD thesis, ETH, Zurich, 2007.

[12] Birgitta Berglund and Thomas Lindvall. Community noise. Technical report, Center for Sensory Research Stockholm, Stockholm, Sweden, 1995.

[13] Daniel J. Bodony. Analysis of sponge zones for computational fluid mechanics. *Journal of Computational Physics*, 212(2):681 – 702, 2006.

[14] Bailly C Bogey C. Three-dimensional non-reflective boundary conditions for acoustic simulations: far field formulation and validation test cases. *Acta Acustica united with Acustica*, 88:463–471, 2002.

[15] O. Boiron, G. Chiavassa, and R. Donat. A high-resolution penalization method for large mach number flows in the presence of obstacles. *Computers & Fluids*, 38(3):703 – 714, 2009.

[16] O. Botella and R. Peyret. Benchmark spectral results on the lid-driven cavity flow. *Computers & Fluids*, 27(4):421–433, 5 1998.

[17] Jean-François Brouckaert, François Mirville, Kevin Phuah, and Peter Taferner. Clean sky research and demonstration programmes for next-generation aircraft engines. 122(1254):1163–1175, 2018.

[18] Eric Brown-Dymkoski, Nurlybek Kasimov, and Oleg V. Vasilyev. A characteristic based volume penalization method for general evolution problems applied to compressible viscous flows. *Journal of Computational Physics*, 262:344 – 357, 2014.

[19] Jean-Sébastien Cagnone, Koen Hillewaert, and Nicolas Poletz. A discontinuous galerkin method for multiphysics welding simulations. *Key Engineering Materials*, 611-612:1319–1326, 2014.

[20] Daniel Canuto and Kunihiko Taira. Two-dimensional compressible viscous flow around a circular cylinder. *Journal of Fluid Mechanics*, 785:349–371, 2015.

[21] Denis-Gabriel Caprace, Simon Buffin, Matthieu Duponcheel, Philippe Chatelain, and Grégoire Winckelmans. Large Eddy Simulation of Advancing Rotor for Near to Far Wake Assessment. In *43rd European Rotorcraft Forum*, 2017.

[22] C. Carton de Wiart and K. Hillewaert. *Development and Validation of a Massively Parallel High-Order Solver for DNS and LES of Industrial Flows*, pages 251–292. Springer International Publishing, Cham, 2015.

[23] C. Carton de Wiart, K. Hillewaert, M. Duponcheel, and G. Winckelmans. Assessment of a discontinuous Galerkin method for the simulation of vortical flows at high Reynolds number. *International Journal for Numerical Methods in Fluids*, 74(7):469–493, 2014.

[24] Corentin Carton de Wiart. *Towards a Discontinuous Galerkin solver for scale-resolving simulations of moderate Reynolds number flows, and application to industrial cases*. PhD thesis, Université catholique de Louvain, 2014.

[25] Corentin Carton de Wiart, Koen Hillewaert, Etienne Lorriaux, and Guillaume Verheylewegen. Development of a discontinuous galerkin solver for high quality wall-resolved/modelled dns and les of practical turbomachinery flows on fully unstructured meshes. In *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, volume 2B: Turbomachinery, pages V02BT39A035–, 2015.

[26] Philippe Chatelain, Stéphane Backaert, Grégoire Winckelmans, and Stefan Kern. Large Eddy Simulation of wind turbine wakes. In Wolfgang Rodi, editor, *Proceedings of The 9th International Symposium on Engineering Turbulence Modelling and Measurements (ETMM-9), June 6–8, 2012, Thessaloniki, Greece*, volume 91 of *Flow, Turbulence and Combustion*, pages 587–605. ERCOFTAC, Springer, Jun 2013.

[27] Philippe Chatelain, Alessandro Curioni, Michael Bergdorf, Diego Rossinelli, Wanda Andreoni, and Petros Koumoutsakos. Billion vortex

particle Direct Numerical Simulations of aircraft wakes. *Computer Methods in Applied Mechanics and Engineering*, 197(13):1296–1304, Feb 2008.

[28] Philippe Chatelain and Petros Koumoutsakos. A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions. *Journal of Computational Physics*, 229(7):2425–2431, 4 2010.

[29] Philippe Chatelain and Anthony Leonard. Isotropic compact interpolation schemes for particle methods. *Journal of Computational Physics*, 227(6):3244 – 3259, 2008.

[30] I.P Christiansen. Numerical simulation of hydrodynamics by the method of point vortices. *Journal of Computational Physics*, 13(3):363 – 379, 1973.

[31] R. Cocle, G. Winckelmans, and G. Daeninck. Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations. *Journal of Computational Physics*, 227(21):9091–9120, Nov 2008.

[32] Lothar Collatz. *The Numerical Treatment of Differential Equations*, volume 60 of *Grundlehren der mathematischen Wissenschaften*. Springer-Verlag Berlin Heidelberg, 1960.

[33] T. Colonius, S. K. Lele, and P. Moin. Boundary Conditions for Direct Computation of Aerodynamic Sound Generation. *AIAA Journal*, 31(9):1574–1582, 1993.

[34] Tim Colonius. Numerically Nonreflecting Boundary and Interface Conditions for Compressible Flow and Aeroacoustic Computations. *AIAA Journal*, 35(7):1126–1133, 2015/06/04 1997.

[35] Tim Colonius and Sanjiva K. Lele. Computational aeroacoustics: progress on nonlinear problems of sound generation. *Progress in Aerospace Sciences*, 40(6):345 – 416, 2004.

[36] M. Coquerelle and G. H. Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *Journal of Computational Physics*, 227(21):9121–9137, 2008.

[37] G.-H. Cottet. *Vortex dynamics and vortex methods*, volume 28, chapter Particle-grid domain decomposition methods for the Navier-Stokes equations in exterior domains. Lectures in Applied Mathematics, 1991.

[38] G.-H. Cottet. Artificial viscosity models for vortex and particle methods. *J. Comput. Phys.*, 127(2):299–308, 1996.

[39] G.-H. Cottet, D. Jiroveanu, and B. Michaux. Vorticity dynamics and turbulence models for large-eddy simulations. *Modélisation Mathematique et Analyse Numérique*, 2002.

[40] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods, Theory and Practice*. Cambridge University Press, 2000.

[41] G.-H. Cottet and P. Poncet. Advances in direct numerical simulations of 3D wall-bounded flows by Vortex-in-Cell methods. *J. Comput. Phys.*, 193(1):136–158, 2004.

[42] Cottet, G.-H., Etancelin, J.-M., Perignon, F., and Picard, C. High order semi-Lagrangian particle methods for transport equations: numerical analysis and implementation issues. *ESAIM: M2AN*, 48(4):1029–1060, 2014.

[43] Michael J. Czech and Russell H. Thomas. Open rotor aeroacoustic installation effects for conventional and unconventional airframes. In *19th AIAA/CEAS Aeroacoustics Conference*, Berlin; Germany, May 2013.

[44] G. Daeninck. *Developments in hybrid approaches. Vortex method with known separation location. Vortex method with near-wall Eulerian solver. RANS-LES coupling*. PhD thesis, UniversitÈ catholique de Louvain, 2006.

[45] A.C. de Bruin and G. Schrauf. Wake vortex results from the awiator project. Technical report, National Aerospace Laboratory NLR, 2007.

[46] Nicolas Docquier, Antoine Poncelet, and Paul Fisette. ROboTraN: a powerful symbolic gnerator of multibody models. *Mechanical Sciences*, 4(1):199–219, 2013.

[47] J. D. Eldredge, A. Leonard, and T. Colonius. A general deterministic treatment of derivatives in particle methods. *J. Comput. Phys.*, 180:686–709, 2002.

[48] Jeff D. Eldredge. *A Dilating Vortex Particle Method for Compressible Flow with Application to Aeroacoustics*. PhD thesis, California Institute of Technology, Pasadena, California, 2002.

[49] Jeff D. Eldredge. The dynamics and acoustics of viscous two-dimensional leapfrogging vortices. *Journal of Sound and Vibration*, 301(1–2):74 – 92, 2007.

[50] Jeff D. Eldredge, Tim Colonius, and Anthony Leonard. A vortex Particle Method for Two-Dimensional Compressible Flow. *Journal of Computational Physics*, 179(2):371–399, jul 2002.

[51] Bjorn Engquist and Andrew Majda. Absorbing Boundary Conditions for the Numerical Simulation of Waves. *Mathematics of Computation*, 31(139):629–651, 7 1977.

[52] Ariane Frère. *Towards wall-modeled Large-Eddy Simulations of high Reynolds number airfoils using a discontinuous Galerkin method*. PhD thesis, Université catholique de Louvain, 2018.

[53] J. B. Freund. Proposed Inflow/Outflow Boundary Condition for Direct Computation of Aerodynamic Sound. *AIAA Journal*, 35(4):740–742, 2016/07/19 1997.

[54] Matteo Frigo and Steven˜ G. Johnson. The Design and Implementation of FFTW3. *Proceedings of the IEEE*, 93(2):216–231. Special issue on ``Program Generation, Optimization, and Platform Adaptation''.

[55] G. Gassner and D. Kopriva. A comparison of the dispersion and dissipation errors of gauss and gauss–lobatto discontinuous galerkin spectral element methods. *SIAM Journal on Scientific Computing*, 33(5):2560–2579, 2018/10/12 2011.

[56] Mattia Gazzola, Philippe Chatelain, Wim M. van Rees, and Petros Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, 230(19):7093–7114, 8 2011.

[57] U Ghia, K. N Ghia, and C. T Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, 48(3):387–411, 1982.

[58] T. Gillis, Y. Marichal, G. Winckelmans, and P. Chatelain. A 2D immersed interface vortex particle-mesh method. *Submitted*, 2019.

[59] T. Gillis, G. Winckelmans, and P. Chatelain. An efficient iterative penalization method using recycled Krylov subspaces and its application to impulsively started flows. *Journal of Computational Physics*, 347:490 – 505, 2017.

[60] T. Gillis, G. Winckelmans, and P. Chatelain. Fast immersed interface Poisson solver for 3D unbounded problems around arbitrary geometries. *Journal of Computational Physics*, 354:403 – 416, 2018.

[61] A. Gillman and P.G. Martinsson. Fast and accurate numerical methods for solving elliptic difference equations defined on lattices. *Journal of Computational Physics*, 229(24):9026 – 9041, 2010.

[62] A. Gillman and P.G. Martinsson. A fast solver for Poisson problems on infinite regular lattices. *Journal of Computational and Applied Mathematics*, 258:42 – 56, 2014.

[63] Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. *ACM Comput. Surv.*, 51(3):49:1–49:33, May 2018.

[64] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[65] J. L. Guermond, S. Huberson, and W. Z. Shen. Simulation of 2D external Viscous Flows by Means of a Domain Decomposition Method. *Journal of Computational Physics*, 108(2):343–352, 1993.

[66] Roy D. Hager and Deborah Vrabel. Advanced turboprop project. Technical report, NASA Lewis Research Center; Cleveland, OH, United States, Jan 1988.

[67] Mads Molholm Hejlesen, Petros Koumoutsakos, Anthony Leonard, and Jens Honoré Walther. Iterative Brinkman penalization for remeshed vortex methods. *Journal of Computational Physics*, 280:547–562, 2015.

[68] Mads Mølholm Hejlesen, Johannes Tophøj Rasmussen, Philippe Chatelain, and Jens Honore Walther. A high order solver for the unbounded Poisson equation. *Journal of Computational Physics*, 252:458–467, 2013.

[69] Mads Mølholm Hejlesen and Jens Honoré Walther. A multiresolution method for solving the Poisson equation using high order regularization. *Journal of Computational Physics*, 326:188 – 196, 2016.

[70] Jan S. Hesthaven and Tim Warburton. *Nodal Discontinuous Galerkin Methods*. Number XIV,502 in 54. Springer-Verlag New York, 1 edition, 2008.

[71] Koen Hillewaert. *Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries*. PhD thesis, Université catholique de Louvain, 2013.

[72] Koen Hillewaert and Corentin Carton de Wiart. The discontinuous galerkin method as an enabling technology for dns and les of industrial

aeronautical applications. In Rolf Radespiel, Reinhard Niehuis, Norbert Kroll, and Kathrin Behrends, editors, *Advances in Simulation of Wing and Nacelle Stall*, pages 75–96, Cham, 2016. Springer International Publishing.

[73] Koen Hillewaert, Corentin Carton de Wiart, Guillaume Verheylewegen, and Tony Arts. Assessment of a high-order discontinuous galerkin method for the direct numerical simulation of transition at low-reynolds number in the t106c high-lift low pressure turbine cascade. (45615):V02BT39A034–, 2014.

[74] R. Hockney and J. Eastwood. *Computer Simulation using Particles*. Taylor & Francis, Inc. Bristol, PA, USA, 1988.

[75] F. Holzäpfel and al. Aircraft wake vortex state-of-the-art & research needs. Technical report, WakeNet3-Europe, Nov 2015.

[76] Mike Hoogstraten, Hendrikus G. Visser, Dennis Hart, Vincent Treve, and Frederic Rooseleer. Improved understanding of en route wake-vortex encounters. *Journal of Aircraft*, 52(3):981–989, 2018/08/18 2014.

[77] Serge Huberson, Elie Rivoalen, and Spyros Voutsinas. Vortex particle methods in aeroacoustic calculations. *Journal of Computational Physics*, 227(21):9216–9240, 11 2008.

[78] Moshe Israeli and Steven A. Orszag. Approximation of radiation boundary conditions. *Journal of Computational Physics*, 41(1):115 – 135, 1981.

[79] R. A. James. Solution of Poisson's equation for isolated source distributions. *Journal of Computational Physics*, 25(2):71–93, 1977.

[80] H. Jeanmart and G. Winckelmans. Investigation of eddy-viscosity models modified using discrete filters: A simplified regularized variational multiscale model and an enhanced field model. *Physics of Fluids*, 19(5):055110, May 2007.

[81] Wan-Ho Jeon, Seung-Jo Baek, and Chang-Joon Kim. Analysis of the aeroacoustic characteristics of the centrifugal fan in a vacuum cleaner. *Journal of Sound and Vibration*, 268(5):1025–1035, 2003.

[82] Guang-Shan Jiang and Chi-Wang Shu. Efficient Implementation of Weighted Eno Schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[83] Nicholas K.-R. Kevlahan and Jean-Michel Ghidaglia. Computation of turubulent flow past an array of cylinders using a spectral method with Brinkman penalization. *European Journal of Mechanics - B/Fluids*, 20(3):333–350, May-June 2001.

[84] Omar M. Knio, Luc Collorec, and Daniel Juvé. Numerical Study of Sound Emission by 2D regular and Chaotic Vortex Configurations. *Journal of Computational Physics*, 116(2):226–246, 1995.

[85] P. Koumoutsakos. Inviscid axisymmetrization of an elliptical vortex. *Journal of Computational Physics*, 138(2):821–857, 1997.

[86] P. Koumoutsakos. Multiscale flow simulations using particles. *Annual Review of Fluid Mechanics*, 37:457–487, 2005.

[87] P. Koumoutsakos and A. Leonard. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid. Mech.*, 296(1):1–38, 1995.

[88] P. Koumoutsakos, A. Leonard, and F. Pépin. Boundary Conditions for Viscous Vortex Methods. *Journal of Computational Physics*, 113(1):52 – 61, 1994.

[89] Ulrich Kowarsch, Constantin Öhrle, Manuel Keßler, and Ewald Krämer. Aeroacoustic simulation of a complete h145 helicopter in descent flight. *Journal of the American Helicopter Society*, 61(4):1–13, 2016-10-01T00:00:00.

[90] H. C. Kuhlmann, M. Wanschura, and H. J. Rath. Flow in two-sided lid-driven cavities: non-uniqueness, instabilities, and cellular structures. *Journal of Fluid Mechanics*, 336:267–299, 4 1997.

[91] Anand Kumar. Isotropic finite-differences. *Journal of Computational Physics*, 201(1):109 – 118, 2004.

[92] K. Lackner. Computation of ideal Mhd equilibria. *Computer Physics Communications*, 12(1):33 – 44, 1976.

[93] Sébastien Leclaire, Maud El-Hachem, Jean-Yves Trépanier, and Marcelo Reggio. High Order Spatial Generalization of 2D and 3D isotropic Discrete Gradient Operators with Fast Evaluation on Gpus. *Journal of Scientific Computing*, 59(3):545–573, 2013.

[94] S. K. Lele. Compact finite-difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16–42, Nov 1992.

[95] R. J. Leveque and Z. L. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, Aug 1994.

[96] Thomas Leweke and al. Publishable final activity report. Technical report, FAR-Wake, Fundamental Research on Aircraft Wake Phenomena;, 2008.

[97] Sebastian Liska and Tim Colonius. A parallel fast multipole method for elliptic difference equations. *Journal of Computational Physics*, 278:76 – 91, 2014.

[98] Sebastian Liska and Tim Colonius. A fast lattice Green's function method for solving viscous incompressible flows on unbounded domains. *Journal of Computational Physics*, 316:360 – 384, 2016.

[99] Qianlong Liu and Oleg V. Vasilyev. A Brinkman penalization method for compressible flows in complex geometries. *Journal of Computational Physics*, 227(2):946–966, 2007.

[100] T. Lonfils. *Vortex particle-mesh method with combined immersed boundary and mesh refinement techniques. Application to bluff-body and wake vortex flows*. PhD thesis, Université catholique de Louvain, 2011.

[101] A. Mani. On the reflectivity of sponge zones in compressible flow simulations. In *Annual Research Briefs*. Center for Turbulence Research, 2010.

[102] Yves Marichal. *An immersed interface vortex particle-mesh method*. PhD thesis, Université catholique de Louvain, Sep 2014.

[103] Yves Marichal, Philippe Chatelain, and Grégoire Winckelmans. An immersed interface solver for the 2-D unbounded Poisson equation and its application to potential flow. *Computers & Fluids*, 96:76–86, 2014.

[104] Yves Marichal, Philippe Chatelain, and Grégoire Winckelmans. Immersed interface interpolation schemes for particle–mesh methods. *Journal of Computational Physics*, 326:947 – 972, 2016.

[105] Chloe Mimeau, Federico Gallizio, Georges-Henri Cottet, and Iraj Mortazavi. Vortex penalization method for bluff body flows. *International Journal for Numerical Methods in Fluids*, 79(2):55–83, 2015. fld.4038.

[106] Brian E. Mitchell, Sanjiva K. Lele, and Parviz Moin. Direct computation of the sound from a compressible co-rotating vortex pair. *Journal of Fluid Mechanics*, 285:181–202, 2 1995.

[107] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005.

[108] W. Möhring. On vortex sound at low Mach number. *Journal of Fluid Mechanics*, 85:685–691, 4 1978.

[109] J.J Monaghan. Extrapolating B splines for interpolation. *Journal of Computational Physics*, 60(2):253 – 262, 1985.

[110] F. Noca, D. Shiels, and D. Jeon. Measuring instantaneous fluid dynamic forces on bodies, using only velocity fields and their derivatives. *Journal of Fluids and Structures*, 11(3):345–350, Apr 1997.

[111] F. Noca, D. Shiels, and D. Jeon. A comparison of methods for evaluating time-dependent fluid dynamic forces on bodies, using only velocity fields and their derivatives. *Journal of Fluids and Structures*, 13(5):551–578, Jul 1999.

[112] Flavio Noca. On the evaluation of instantaneous fluid-dynamic forces on a bluff body. Report FM96-5, GALCIT, 1996.

[113] M. L. Ould-Salihi, G.-H. Cottet, and M. El Hamraoui. Blending Finite-Difference and Vortex Methods for Incompressible Flow Computations. *SIAM J. Sci. Comput.*, 22(5):1655–1674, may 2000.

[114] Gregory S. Oxley. *A 2-D Hybrid Euler-Compressible Vortex Particle Method For Transonic Rotorcraft Flows*. PhD thesis, Department of Mechanical and Aerospace Engineering, Carleton University, Ottawa, Ontario, Canada, July 2009.

[115] A. Pahla, L. Manickathan, C.S. Ferreira, and G. van Bussel. A hybrid Eulerian-Lagrangian flow solver. *Submitted to Elsevier*, 6 2015. arXiv:1505.03368 [math.NA].

[116] G. Papadakis, D. Manolas, V. A. Riziotis, and S. G. Voutsinas. A novel hybrid method for helicopter cost effective aeroelastic simulations. In *43rd European Rotorcraft Forum, ERF 2017*, volume 1, pages 53–66, 2017.

[117] Giorgos Papadakis. *Development of a hybrid compressible vortex particle method and application to external problems including helicopter flows*. PhD thesis, NATIONAL TECHNICAL UNIVERSITY OF ATHENS, December 2014.

[118] Giorgos Papadakis and Spyros G Voutsinas. In view of accelerating Cfd simulations through coupling with vortex particle approximations. *Journal of Physics: Conference Series*, 524(1):012126, 2014.

[119] Philippe Parmentier, Philippe Chatelain, and Grégoire Winckelmans. A fourth order vortex particle-mesh method for compressible flows. In *7th International Conference on Vortex Flows and Vortex Models (ICVFM)*, Rostock, Germany, September 2016.

[120] Philippe Parmentier, Grégoire Winckelmans, and Philippe Chatelain. A vortex Particle-Mesh method for subsonic compressible flows. *Journal of Computational Physics*, 354:692 – 716, 2018.

[121] Philippe Parmentier, Grégoire Winckelmans, Philippe Chatelain, et al. Development and validation of a 2-D compressible vortex particle-mesh method. volume 59. American Physical Society, 2014.

[122] Philippe Parmentier, Grégoire Winckelmans, Philippe Chatelain, and Koen Hillewaert. Coupling of a compressible vortex particle-mesh method with a near-body compressible discontinuous Galerkin solver. volume 59. American Physical Society, 2015.

[123] Michael Patra and Mikko Karttunen. Stencils with isotropic discretization error for differential operators. *Numerical Methods for Partial Differential Equations*, 22(4):936–953, 2006.

[124] Di Peng and James W. Gregory. Vortex dynamics during blade-vortex interactions. *Physics of Fluids*, 27(5):053104, 2015.

[125] Per-Olof Persson. *Shock Capturing for High-Order Discontinuous Galerkin Simulation of Transient Flow Problems*. American Institute of Aeronautics and Astronautics, 2018/10/19 2013.

[126] Per-Olof Persson and Jaime Peraire. *Sub-Cell Shock Capturing for Discontinuous Galerkin Methods*. American Institute of Aeronautics and Astronautics, 2018/10/19 2006.

[127] A. Piquet, O. Roussel, and A. Hadjadj. A comparative study of brinkman penalization and direct-forcing immersed boundary methods for compressible viscous flows. *Computers & Fluids*, 136:272 – 284, 2016.

[128] P. Ploumhans and G. S. Winckelmans. Vortex Methods for High-Resolution Simulations of Viscous Flow Past Bluff Bodies of General Geometry. *Journal of Computational Physics*, 165(2):354–406, 2000.

[129] P. Ploumhans, G.S. Winckelmans, J.K. Salmon, A. Leonard, and M.S. Warren. Vortex methods for direct numerical simulation of three-dimensional bluff body flows: Application to the sphere at re=300, 500, and 1000. *Journal of Computational Physics*, 178(2):427 – 463, 2002.

[130] Pothou, K. P., Voutsinas, S. G., Huberson, S. G., and Knio, O. M. Application of 3D particle method to the prediction of aerodynamic sound. *ESAIM: Proc.*, 1:349–362, 1996.

[131] Rashmi Ramadugu, Sumesh P. Thampi, Ronojoy Adhikari, Sauro Succi, and Santosh Ansumali. Lattice differential operators for computational physics. *EPL (Europhysics Letters)*, 101(5):50006, 2013.

[132] Johannes Tophøj Rasmussen, Georges-Henri Cottet, and Jens Honoré Walther. A multiresolution remeshed Vortex-In-Cell algorithm using patches. *Journal of Computational Physics*, 230:6742–6755, 2011.

[133] M. Rasquin and K. Hillewaert. Co-processing and post-processing of high order polynomial solutions on massively parallel systems. In *7th ECCOMAS ECFD conference*, Glasgow, United Kingdom, June 11-15 2018.

[134] Jean-François Remacle, Joseph E Flaherty, and Mark S Shephard. An adaptive discontinuous galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM review*, 45(1):53–72, 2003.

[135] Diego Rossinelli, Michael Bergdorf, Georges-Henri Cottet, and Petros Koumoutsakos. GPu Accelerated Simulations of Bluff Body Flows Using Vortex Particle Methods. *J. Comput. Phys.*, 229(9):3316–3333, may 2010.

[136] Diego Rossinelli, Babak Hejazialhosseini, Wim van Rees, Mattia Gazzola, Michael Bergdorf, and Petros Koumoutsakos. MRag-I2d: Multi-resolution adapted grids for remeshed vortex methods on multicore architectures. *Journal of Computational Physics*, 288:1–18, 5 2015.

[137] P. Schrooyen, L. Arbaoui, J.-S. Cagnone, N. Poletz, and K. Hillewaert. A high-order extended discontinuous galerkin method to treat hydrodynamics problems. In *ECCM-ECFD Conference*, Glasgow, United Kingdom, June 11-15 2018.

[138] P. Schrooyen, J.-S. Cagnone, N. Poletz, and K. Hillewaert. High-order extended discontinuous galerkin method for moving immersed interface problems, extended discretization methods. Umea, Sweden, June 19-21 2017.

[139] Pierre Schrooyen. *A Numerical Simulation Tool for the Aerothermal Flow Through an Ablative Thermal ProtectionSystem for Atmospheric Entry*. PhD thesis, Université catholique de Louvain, 2015.

[140] Pierre Schrooyen, Koen Hillewaert, Thierry E. Magin, and Philippe Chatelain. Fully implicit discontinuous galerkin solver to study surface and volume ablation competition in atmospheric entry flows. *International Journal of Heat and Mass Transfer*, 103:108–124, 2016.

[141] Adrian Sescu. Numerical anisotropy in finite differencing. *Advances in Difference Equations*, 2015(1):1–17, 2015.

[142] P. Shah, B. Rovagnati, F. Mashayek, and G. B. Jacobs. Subsonic compressible flow in two-sided lid-driven cavity. Part I: Equal walls temperatures. *International Journal of Heat and Mass Transfer*, 50(21–22):4206–4218, 10 2007.

[143] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In Alfio Quarteroni, editor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures given at the 2nd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) held in Cetraro, Italy, June 23–28, 1997*, pages 325–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[144] Chi-Wang Shu and Stanley Osher. Efficient Implementation of Essentially Non-oscillatory Shock-capturing Schemes. *J. Comput. Phys.*, 77(2):439–471, aug 1988.

[145] Clean Sky. Sustainable and green engines (sage). http://www.cleansky.eu/sustainable-and-green-engines-sage.

[146] H. Spietz. *Vortex-particle Mesh Method for Large Eddy Simulation of Bluff Body Aerodynamics*. PhD thesis, Technical University of Denmark, 2018.

[147] Mark Stock, Adrin Gharakhani, and Christopher Stone. *Modeling Rotor Wakes with a Hybrid OVERFLOW-Vortex Method on a GPU Cluster*. American Institute of Aeronautics and Astronautics, 2015/08/11 2010.

[148] A. Stuermer, J. Yin, and R. Akkermans. Progress in aerodynamic and aeroacoustic integration of CROR propulsion systems. *The Aeronautical Journal (1968)*, 118(1208):1137–1158, 2014.

[149] C. K. W. Tam and J. C. Hardin (Eds.). Second computational aeroacoustics (caa) workshop on benchmark problems. NASA Conference Publication 3352, NASA, 1997.

[150] Christopher K.W. Tam and Jay C. Webb. Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics. *Journal of Computational Physics*, 107(2):262 – 281, 1993.

[151] C.K.W. TAM and Z. DONG. Radiation and outflow boundary conditions for direct computation of acoustic and flow disturbances in a nonuniform mean flow. *Journal of Computational Acoustics*, 04(02):175–201, 1996.

[152] Sumesh P. Thampi, Santosh Ansumali, R. Adhikari, and Sauro Succi. Isotropic discrete Laplacian operators from lattice hydrodynamics. *Journal of Computational Physics*, 234:1 – 7, 2013.

[153] F. Thirifay. *Simulation of non-reactive and reactive shear flows using lagrangian particle methods*. PhD thesis, UniversitÈ catholique de Louvain, 2006.

[154] F. Thirifay and G. Winckelmans. Development of a Lagrangian method for combustion and application to the planar methane air jet diffusion flame. *Journal of Turbulence*, 3:059, Dec 2002.

[155] Milton. Van Dyke. *Perturbation methods in fluid mechanics, by Milton Van Dyke*. Academic Press New York, 1964.

[156] Wim M. van Rees, Anthony Leonard, Dale I. Pullin, and Petros Koumoutsakos. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *Journal of Computational Physics*, 230(8):2794–2805, 2011.

[157] Dale E. Van Zante. Progress in open rotor research: A u.s. perspective. (56628):V001T01A003–, 2015.

[158] Roelof Vos and Saeed Farokhi. *Introduction to Transonic Aerodynamics*. Springer Netherlands, Dordrecht, 2015.

[159] G. Winckelmans, R. Cocle, L. Dufresne, and R. Capart. Vortex methods and their application to trailing wake vortex simulations. *C. R. Phys.*, 6(4-5):467–486, 2005.

[160] G. S. Winckelmans. *Encyclopedia of Computational Mechanics*, chapter Vortex Methods, pages 129–153. John Wiley & Sons, Ltd, 2004.

[161] Grégoire Winckelmans. Recent developments in Vortex Particle-Mesh (Vpm) methods, and applications. In *7th International Conference on Vortex Flows and Vortex Models, Rostock*, Sep 2016.

[162] Rio Yokota and L. A. Barba. FMm-based vortex method for simulation of isotropic turbulence on Gpus, compared with a spectral method. *Computers & Fluids*, 80:17–27, 7 2013.

[163] Rio Yokota, L.A. Barba, Tetsu Narumi, and Kenji Yasuoka. Petascale turbulence simulation using a highly parallel fast multipole method on Gpus. *Computer Physics Communications*, 184(3):445 – 455, 2013.

[164] Shuhai Zhang, Hu Li, Xuliang Liu, Hanxin Zhang, and Chi-Wang Shu. Classification and sound generation of two-dimensional interaction of two Taylor vortices. *Physics of Fluids*, 25(5), 2013.