Université catholique de Louvain
Ecole polytechnique de Louvain
Institut de Mécanique, Matériaux et Génie Civil

An immersed interface vortex particle-mesh method

Yves Marichal

Composition du jury:

Pr. I. Doghri (UCL, président)
Pr. G. Winckelmans (UCL, promoteur)
Pr. P. Chatelain (UCL, promoteur)
Dr. M. Duponcheel (UCL)
Dr. K. Hillewaert (Cenaero)
Pr. P. Poncet (UPPA)

Thèse présentée en vue de
l'obtention du grade de docteur
en sciences de l'ingénieur

Louvain-la-Neuve,
Septembre 2014

A la mémoire de Maman et Dominique.

Acknowledgments

Je ne pourrais entamer ces remerciements sans exprimer ma sincère reconnaissance envers mes promoteurs, Grégoire Winckelmans et Philippe Chatelain. Je leur adresse toute ma gratitude pour la passion qu'ils m'ont communiquée et pour le soutien qu'ils m'ont apporté dans les bons et les mauvais moments. Mais aussi, et surtout, je les remercie d'avoir cru en moi, à défaut de le faire moi-même par moments. Je remercie Grégoire tout particulièrement pour sa rigueur, son optimisme, son investissement et ... son extrême efficacité dans la remise de ses corrections. Je remercie Philippe spécialement pour son enthousiasme, son imagination débordante et pour sa disponibilité quasi permanente.

En termes d'encadrement, je souhaite également citer Matthieu Duponcheel, au vu de son aide et de tout le temps qu'il m'a consacré. Merci à Matthieu pour sa générosité, son désir d'aider les gens, ses conseils, sa clairvoyance et pour sa persévérance dans la recherche de solutions.

Je désire remercier Koen Hillewaert, Philippe Poncet, ainsi que Matthieu d'avoir accepté de faire partie de mon jury et d'avoir pris le temps de relire ce travail. J'ai apprécié leurs remarques pertinentes lors de ma défense et le regard critique qu'ils y ont apporté. Merci aussi à Issam Doghri d'avoir accepté de présider le jury au pied levé. Par la même occasion, je voudrais remercier Philippe Geuzaine pour sa participation à mon comité d'accompagnement.

J'exprime ma gratitude au F.R.S.-FNRS (Fonds de la Recherche Scientifique), qui m'a permis de réaliser cette thèse en m'accordant une bourse d'aspirant FNRS. Je remercie Cenaero (Centre of Excellence in Aeronautical Research) et le C.I.S.M. (Centre de calcul intensif et de stockage de masse de l'UCL) pour la mise à disposition des ressources de calcul me permettant d'effectuer les simulations nécessaires à ma thèse.

Les années passées au sein du pôle TFL ont été mémorables grâce aux moments partagés ensemble. On ne s'ennuie pas au TFL : Saint-Eloi, séminaires

et conférences, courses à pied, matchs de foot et de badminton, journées de pôle et drinks rythment le chemin parfois trop studieux du thésard. Je ne peux malheureusement pas nommer de manière exhaustive l'ensemble des personnes avec lesquelles je partage d'excellents souvenirs, en termes de collaboration ou même d'amitié : Laurent B., Isabelle, Ivan, Matthieu, Christos, Hervé, Philippe P., Pierre, Tim, Francesco, Subir,...

Je tiens tout particulièrement à citer Nicolas B., Benjamin, Corentin, Haddy, Panos et François H., sans lesquels mon séjour dans le pôle n'aurait clairement pas eu la même saveur. Certains ont partagé mon bureau, certains y venaient souvent pour discuter et d'autres venaient essentiellement pour les drinks. Bien entendu, s'ajoutent à cela des tas d'activités en commun, comme les matchs au stade, les fameux cours GrasMech (avec les soirées qui les précédaient parfois la veille) ou les soirées entre amis. Par la même occasion, merci à Stefano qui lors d'un de ces moments privilégiés m'a permis de rencontrer la personne qui depuis lors partage ma vie...

Ma vie de doctorant aura aussi été marquée par les différents kots ou appartements, dans lesquels j'ai séjourné depuis le début de mes études (rue du Collège, Germanokot place des Wallons, résidence Pomerol et Bruxelles). J'ai ainsi eu la chance d'enrichir ma collection d'excellents souvenirs en compagnie de très bons amis: David, Christine, André, Frank, Moritz, Christoph, Dany, Pierre, Michael, Jennifer, Katharina, ... Das Kotleben war einfach genial, sowie die Soirées im Only, im Coq, in der GD, ... Ach ja, vielen Dank für's Kochen, Michael (man war Carbonara geil ...).

Merci aussi à tous les autres amis de longue date pour toutes les folies réalisées ensemble et toutes celles à venir: Pierre-Henri, Aurélie, Gregory, Pol-Axel, Pierre-Nicolas, Julien, Daniel, Michael, Rinus, ... Au plaisir de se revoir aux quatre coins de la Belgique (ou d'ailleurs). Merci PH pour nos multiples discussions ouvertes et sincères. Auf die künftigen Abenteuer und Erlebnisse in London, Australien oder wo auch immer.

Un tout grand merci à ma famille qui m'a accompagné tout le long de ce chemin. Un merci tout particulier à Papa et Valérie pour votre amour et votre soutien inconditionnel.

Finalement, je ne sais pas comment je pourrais assez exprimer ma gratitude envers Claire, qui a littéralement illuminé ma vie, qui m'a supporté dans les moments difficiles et qui a tant fait pour moi au jour le jour, afin de m'encourager quand le besoin en était. Je tiens aussi à remercier sa famille, avec une attention particulière à sa maman pour sa gentillesse et sa générosité.

Contents

Chapter 1

Introduction

Nowadays, the usage of numerical simulation has become standard in the domains of industrial research and design processes. The evolution and the increasingly widespread application of these techniques are mostly related to the very rapid development of computing technology, from the hardware and the software point of view.

Computer simulation applications range across numerous fields that require an accurate prediction and quantification of complex physical phenomena; these include fluid dynamics, chemical processes, quantum mechanics, solid mechanics, etc. In many cases, simulation can be considered as a valuable alternative to the costly development of prototypes (e.g. predicting the aerodynamic performance of a wind turbine by studying experimentally the flow past a small scale replica of the original model inside a wind tunnel). Besides, sometimes, it is the only way to gain some insight into physical phenomena that are otherwise not observable with experimental studies.

Furthermore, simulation is not limited to the domain of engineering, as one may find applications in biology, astrophysics, economics, geology, climatology, movie industry, game industry, etc. One common example is the weather forecast, which relies on very complex physical modeling.

The need for energy efficiency, the ever higher quality standards (in terms of comfort, design, etc.) and the increasing security requirements drive a demand for high-fidelity simulations that are able to accurately predict the behavior of physical systems. Yet, this accuracy requirement goes along with a considerable increase of computational needs.

In response to this request, new computational architectures have been developed in order to allow parallel computations on multiple processors. This has significantly extended the range of applications and has further led to the emergence of new specifically dedicated research fields, such as computational science and high-performance computing (HPC). Both domains are intrinsically multi disciplinary, as they make the link between physics, computer science, algorithmics and numerical analysis.

As an example of the extent of such computations, state-of-the-art simulations currently run simultaneously on hundreds of thousands cores, sometimes during several months. It is thus not difficult to realize that the associated computational cost, in terms of money and time, becomes more and more critical.

The above observations are particularly true for computational fluid dynamics (CFD). In fluid mechanics, any flow configuration can be characterized by the associated Reynolds number Re, defined as the ratio between the inertial forces and the viscous forces. The higher the Reynolds number, the more turbulent the flow becomes and the tinier the smallest flow structures/scales get. Hence, simulating a high Reynolds number flow requires more computational resources than a low Reynolds number flow, as the former contains more information.

It can be shown that the computational complexity for the simulation of an unbounded flow scales like Re^3, when all scales in time and space are captured (i.e. DNS, direct numerical simulation). The highest Reynolds number currently achievable by a time and space resolving simulation is of the order of $10^4 - 10^5$. Considering that industrial flows typically encountered in the domain of aerodynamics are characterized by $Re \simeq 10^7 - 10^8$ (e.g. the flow past an airplane), one can get an impression of the extent of the gap that remains to be bridged.

As a consequence, great efforts are made in order to improve the computational cost of currently available simulation techniques.

On the one hand, relying on turbulence modeling approaches such as RANS (Reynolds-Averaged Navier-Stokes), LES (large eddy simulation) or hybrid RANS/LES methods, allows further reducing the computational cost by truncating the range of scales that need to be captured. The choice of the turbulence modeling approach requires a trade-off between accuracy and the related cost.

On the other hand, another approach consists in developing simulation tools that are more specific, in the sense that they focus on very precise applications. This allows adapting the tool to the situation that is studied, by exploiting its specificities, and hence optimizing the related computational performance.

The present work considers the second approach and focuses on the framework of external incompressible aerodynamics. The simulation methodology is based on vortex methods, which are particularly well-suited for this type of flow, thanks to their numerical properties. As an example, accounting for an unbounded flow domain is handled quite naturally, compared to other approaches. Vortex methods are Lagrangian methods using a set of particles that are transported by the flow. These particles carry information about the vorticity (i.e. a measure of the local flow rotation speed), based on which the entire flow field can be reconstructed. In 1931, Rosenhead [108] was the first to use what is now known as a vortex method, while in the absence of any computing facility, he performed the computations by hand...

Vortex methods have evolved a lot since then, as many research efforts have been made in order to improve the efficiency and the versatility of the approach. Applications include aircraft wakes [131], bluff body flows [103, 38], wind turbine wakes [4], reactive flows [124], biological flows [19], biolocomotion [50],etc.

As for any incompressible flow solver, the main challenge consists in efficiently solving the underlying Poisson equation, as it represents the most expensive computational operation. A major turning point, which made the approach computationally attractive was the use of fast multipole methods [55, 6]. In this way, the number of operations required for solving Poisson equation was reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \log N)$ or even to $\mathcal{O}(N)$ (depending on the algorithm), where N is the number of particles.

Among other things, different techniques from the better known Eulerian methods have also been integrated and have led to the vortex particle-mesh (VPM) methods [23], combining the particle information with an underlying grid. The main advantage related to this development is that it offers the possibility to rely on even faster grid solvers for the Poisson equation [121, 49], compared to multipole methods. The combination of a grid solver and multipole methods further improves the efficiency, as was shown in [28].

In parallel, the presence of solid bodies inside the flow domain has been accounted for by using different techniques ranging from vorticity flux- and panel-based approaches [72, 102] to penalization methods [32, 109, 50]. Penal-

ization methods are very easy to implement but they lack accuracy near the wall, as the force regularization tends to smear the solution in the vicinity of the boundaries. Panel-based methods achieve a higher precision, as they provide a sharper treatment of the wall by relying on a boundary element method.

Yet, the application of previous approaches is limited to moderate Reynolds number flows. Indeed, in the framework of wall-bounded flows, vortex methods are computationally penalized with respect to unstructured Eulerian approaches when the Reynolds number increases, as the latter offer more flexibility in the near-wall treatment by using body-fitted meshes. This naturally suggests combining both methodologies into a hybrid Eulerian-Lagrangian solver. The near-wall flow is then accurately captured using the Eulerian solver and the solution elsewhere in the domain is provided by the Lagrangian solver, as vortex methods are designed to efficiently compute unbounded flows. The idea has been around for a long time now and it has been successfully applied to 2-D flows [98, 40].

The starting point for this thesis consists in generalizing the above hybrid approach to 3-D flows, as explained in the following:

- **Chapter 2** describes the development of a 3-D hybrid Eulerian-Lagrangian solver based on the vortex particle-mesh method developed in Cocle [28] and Lonfils [83], and on an unstructured finite volume solver from the OpenFOAM software library [1, 128]. First, a more extensive introduction to vortex particle mesh-methods is given, as it lays the basis for the remainder of this work, and secondly, the coupling approach is described. It is performed by using the overlapping domain technique from Daeninck [40]. As a consequence, the vortex method component of the hybrid approach must compute the solution in the entire flow domain, i.e. it also needs to (roughly) estimate the near-wall flow (using among other things a vortex panel method). The representation of the flow in that region is subsequently refined using the Eulerian solver, somewhat like a near-wall corrector step that would succeed the Lagrangian solver-based predictor step. The methodology is then illustrated on the flow past a sphere at $Re = 300$.

The application of the hybrid approach onto the above test case reveals the presence of spurious high-frequency oscillations in the drag coefficient, though. The questions about the exact reason for this behavior, as well as about its origin, remain unanswered at the time of writing. Yet, some observations

made in Lonfils [83] about the stand-alone VPM solver hint at a consistency problem between the vortex panel solver and the grid-based Poisson solver. This assertion is further supported by the analysis made in Chapter 2, about the drag induced by the vortex panels in the case of the hybrid approach: it can be observed that the panel-induced drag presents an oscillation pattern similar to that of the total drag signal. Even if these observations do not prove that the aforementioned inconsistency is at the root of the problem, they do indicate that the current treatment of solid walls in VPM methods, using vortex panels, is perfectible.

Considering the difficulties experienced while trying to remedy the spurious behavior of the Eulerian-Lagrangian solver, the choice has been made in this work to address the consistency issue by focusing on the way to account for solid walls in VPM methods, as will be detailed hereafter. However, it is not clear yet whether or not the suggested procedure would significantly improve the results of the hybrid methodology, as this remains to be verified.

In the light of previous discussion, a novel approach for the treatment of the wall in VPM methods is hence developed throughout this thesis, by using an immersed interface technique initially introduced by Leveque & Li [77]. The rationale for this approach is based on the observation that all spatial differential operations can be performed consistently on the grid by using similar stencil corrections accounting for the presence of the wall.

Basically, the finite difference schemes used for the Poisson equation and the discretization of the diffusion term are corrected when their stencil crosses the boundary. The associated corrections then allow maintaining the same accuracy at the wall as inside the volume. The possible discontinuities of the field (and of its spatial derivatives) are taken into account in a "sharp" and furthermore consistent manner, as all operations are performed on the grid by following the same philosophy. One of the features of the present approach resides in the fact that the corrections are purely "one-dimensional", as they are computed at the intersections of the grid lines with the interface. This allows performing the corrections independently of each other, one spatial direction at a time.

The challenge of this work consists in developing the numerical tools that are required for an immersed interface-enabled VPM solver. Therefore, all different operations of classical VPM methods are revisited and finally replaced by their immersed interface counterpart. The main ingredients are the Poisson solver, the computation of the diffusion term and the particle-mesh interpolation.

With the view of developing a novel technique to account for solid walls inside VPM methods, the present developments are made in two dimensions and serve as a proof of concept. The outline for the rest of this thesis is as follows:

- **Chapter 3** is essentially introductory for the following chapters and it serves two purposes. First, the vortex particle-mesh method is presented in 2-D, along with a detailed description of the different computational steps it requires. Second, the no-slip enforcement techniques in vortex methods are briefly reviewed. More specifically, the classical vorticity flux-based approach inspired from Lighthill's model [79] is thoroughly depicted, as this technique is later used in the immersed interface framework. This technique is unfortunately only first order accurate in time. Hence, by keeping the momentum of increasing the overall accuracy of VPM methods, some suggestions are made with a view to forthcoming work that will aim at improving the temporal accuracy of the no-slip enforcement procedure. The presented analysis shall be considered as an effort suggesting some tracks for future investigations.

  In order to keep the problem geometry simple, the numerical test case adopted for the assessment of the convergence rate consists of a dipole flow inside a cavity. This allows bypassing the errors due to arbitrary intersections between the wall and the grid, as would be introduced by immersed interface methods, and one may thus focus on the no-slip condition alone. As a consequence, this benchmark may also be considered as a "sand box" problem for the validation of future developments.

- **Chapter 4** presents a 2-D Poisson solver for the computation in an unbounded domain of a velocity field that satisfies a no-through flow condition at solid walls. This tool is intended to replace the panel-based Poisson solver from [83], as it provides a more consistent treatment of the walls. It is based on corrected finite differences, along the lines of the immersed interface approach that was introduced in [77]. The unbounded character of the solution is obtained by an iteration inspired from the James-Lackner algorithm [62, 74]. The approach is validated through grid convergence studies on the potential flow past one or multiple bodies, by either prescribing the circulation around the body or by enforcing the Kutta-Joukowsky condition (for airfoils). This chapter further introduces

the tools required in the following chapters for the development of an immersed interface vortex particle-mesh method.

- **Chapter 5** provides a numerical framework in order to compute the solution of the parabolic heat equation with a flux condition at the wall. Enforcing a Neumann condition is not straightforward, since the chosen immersed interface approach relies on one-dimensional stencil corrections along the grid lines, while the Neumann condition is intrinsically 2-D. A "compatible extrapolation scheme" accounting for the flux is presented. The stability of the discretization is studied, as well as the spatial accuracy of the scheme.

- **Chapter 6** focuses on the interpolation that is required between the particles and the grid. One distinguishes here the mesh-to-particle (M2P) interpolation and the particle-to-mesh interpolation (P2M). Basically, both approaches rely on high order interpolation kernels, conforming to what is classically done in VPM methods. Again, the main challenge consists in preserving the accuracy of the interpolation in the presence of the wall, similarly to the two previous chapters. The M2P interpolation requires the computation of grid ghost values and the P2M interpolation introduces ghost particles, whose intensity is obtained by a level set extension technique. The spatial accuracy is also studied.

- **Chapter 7** finally combines the tools from previous chapters into an immersed interface vortex particle-mesh solver. All spatial differential operations are consistently performed on the grid, whereas the time evolution of the particles and their displacement are computed in a Lagrangian fashion. The methodology is validated on the impulsively started flow past a cylinder at $Re = 550$ and $Re = 3000$ and the results are compared to references in the literature. The ability of the solver to handle sharp bodies, such as an airfoil, is further demonstrated by studying the flow past a NACA0021 airfoil at $Re = 500$. The vortex shedding formation for a cylinder at $Re = 100$ is also examined using the present approach.

- **Chapter 8** studies the numerical dispersion and dissipation errors that are introduced by the redistribution of the particles, in the 1-D case. Indeed, contrary to purely Lagrangian methods that enjoy negligible numerical dispersion and dissipation errors, the VPM methods become more

and more similar to classical grid methods, at least in terms of their numerical properties. In that sense, a thorough analysis of the effect of the interpolation scheme and of the redistribution frequency onto these errors is performed.

Some additional tools concerning the immersed interface techniques and the study of the numerical errors are provided in the appendices of this document. Furthermore, Appendix F reproduces a conference paper [86] that was written in the framework of the AIAA BANC-I workshop in Stockholm, 2010. The flow past two cylinders in tandem configuration is studied at $Re = 1.6 \cdot 10^5$, using an unstructured finite volume solver [53] and the delayed detached eddy simulation (DDES) approach from [117].

Chapter 2

Coupling a vortex particle-mesh method with a near-wall finite volume solver

Lesser-known than the widely-used finite volume (FV) methods in the computational fluid dynamics (CFD) community, vortex particle-mesh (VPM) methods are a subset of the so-called vortex methods. This type of method offers a valuable alternative to classical numerical approaches, especially for computational aerodynamics. The term "vortex methods" refers to a class of Lagrangian methods relying on a set of particles in order to compute the entire flow field. An overview of vortex methods is provided in Cottet & Koumoutsakos [34] and Winckelmans [132].

As will be explained in this chapter, providing a more accurate representation of the flow in the vicinity of solid walls using an auxiliary solver is the logical continuation in the development of vortex methods, especially at high Reynolds numbers.

Vortex methods are characterized by several features (low numerical dispersion and dissipation errors, a relaxed CFL stability condition) that are particularly interesting for the simulation of high Reynolds number *unbounded* vortical flows, such as wakes or jets. Yet, they suffer from one clear disadvan-

tage when solid walls are accounted for and that disadvantage is related to the nature of the near-wall flow and to that captured by the particles. Particles are inherently spatially isotropic in the sense that their dimension is identical in all directions. On the contrary, the near-wall flow, i.e. the boundary layer, is intrinsically anisotropic, as the velocity gradients in the direction perpendicular to the wall predominate. The higher the Reynolds number, the tighter the boundary layer and hence the more the anisotropy prevails.

Unstructured Eulerian methods, such as finite volume methods, typically use body-fitted meshes consisting of computational elements near the wall that are stretched in the directions parallel to the wall, which provides a natural treatment for the flow anisotropy. The boundary layer is hence more efficiently captured than it is by a vortex method that would require using a huge amount of particles near the wall so as to yield the same precision as a body-fitted approach. It can be shown, for wall-bounded flows, that the ratio between the number of points required when using a non-conforming grid (e.g. for immersed boundary/interface methods) and when using a body-fitted mesh scales like $Re^{1.0}$ in 2-D and like $Re^{1.5}$ in 3-D [92]. This shows that, for an increasing Reynolds number, using a body-fitted mesh becomes computationally far more interesting, from the standpoint of the number of unknowns.

Realizing that both approaches have complementary strengths in distinct parts of the flow domain naturally leads to the concept of a hybrid Eulerian-Lagrangian solver, where the near-wall flow is computed using an Eulerian method such as finite volumes, whereas the wake is simulated using a vortex method. A few successful achievements are reported in the vortex method literature about this type of coupling. The Eulerian solver must not necessarily be a finite volume solver: using any body-fitted approach is suitable, e.g. finite difference methods, finite elements, discontinuous Galerkin methods, ... Ould-Salihi et al. [98] used an overlapping domain technique based on finite differences and a Schwarz iteration in order to simulate 2-D flows. Daeninck [40] used a similar 2-D approach, yet one that does not require any Schwarz iteration. Based on the latter, Lonfils [83] made the first attempt in 3-D, using a compressible finite volume near-wall solver. Yet, the developed approach was not fully operational and validation was hence not carried out.

Following Lonfils [83], the present developments focus on coupling a 3-D VPM method to an incompressible finite volume solver from the OpenFoam [1, 128] software library. Sections 2.1 and 2.2 briefly present the two underlying

solvers. The coupling algorithm is presented in Section 2.3, and some results are shown thereafter in Section 2.4. As will be seen, these results suggest the need for further developments, as the global flow diagnostics (i.e. the forces) exhibit strong oscillations in time. Finally, some conclusions are presented in Section 2.5.

2.1 Vortex-particle mesh solver

One of the main motivations for vortex methods stems from the observation that the vorticity field $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ typically has a much smaller support compared to the velocity field $\mathbf{u}$, which makes the choice of using the vorticity as a primary variable particularly interesting, from a computational point of view. Indeed, the flow past solid bodies, as encountered in the framework of external aerodynamics, induces a non-zero vorticity that is confined inside the boundary layers and the wakes downstream of the bodies. On the contrary, the velocity significantly differs from the uniform upstream flow, even very far from the bodies, as can be observed for a simple potential flow.

Vortex methods are based on the vorticity-velocity formulation of the Navier-Stokes equations for incompressible flows ($\nabla \cdot \mathbf{u} = 0$)

$$\frac{D\boldsymbol{\omega}}{Dt} = (\nabla \mathbf{u}) \cdot \boldsymbol{\omega} + \nu \nabla^2 \boldsymbol{\omega} \,, \tag{2.1}$$

where $D/Dt \triangleq \partial/\partial t + \mathbf{u} \cdot \nabla$ is the material derivative and ν is the kinematic viscosity. The outer boundary condition in an unbounded domain is typically a uniform upstream velocity field, it is given by $\mathbf{u} = \mathbf{U}_\infty$ when $|\mathbf{x}| \to \infty$. Thanks to the incompressibility and to the resulting Helmholtz decomposition $\mathbf{u} = \nabla \times \boldsymbol{\Psi} + \mathbf{U}_\infty$, the velocity can be computed from the streamfunction $\boldsymbol{\Psi}$ by solving the following Poisson equation

$$\nabla^2 \boldsymbol{\Psi} = -\boldsymbol{\omega} \,, \tag{2.2}$$

as $\boldsymbol{\Psi}$ is chosen according to Lorenz gauge ($\nabla \cdot \boldsymbol{\Psi} = 0$).

The spatial discretization is performed by using N particles of position $\mathbf{x}_p(t)$ and carrying a vorticity intensity $\boldsymbol{\alpha}_p(t) \triangleq \int_{\Omega_p} \boldsymbol{\omega} \, d\mathbf{x} \simeq \boldsymbol{\omega}_p V_p$, where $\boldsymbol{\omega}_p(t)$ is the particle vorticity, $\Omega_p(t)$ its domain and V_p its volume. According to [132], the time evolution of these particles is prescribed by the following ordinary

differential equations resulting from the Navier-Stokes Eq. (2.1)

$$\begin{aligned} \frac{d\mathbf{x}_p}{dt} &= \mathbf{u}_p \\ \frac{d\boldsymbol{\omega}_p}{dt} &= (\nabla \mathbf{u})_p \cdot \boldsymbol{\omega}_p + \nu \ \left(\nabla^2 \boldsymbol{\omega}\right)_p \ , \end{aligned} \tag{2.3}$$

where $\mathbf{u}_p = \mathbf{u}(\mathbf{x}_p)$ is the particle velocity (more generally, $f_p(t) \triangleq f(\mathbf{x}_p(t), t)$ with $f(\mathbf{x}, t)$ the Eulerian representation of a field f).

Purely Lagrangian methods make use of fast summation techniques based on multipole expansions of the free space Green's function [55, 56, 6, 111] in order to solve Eq. (2.2) and hence compute the velocity at each particle (and also the associated stretching term $(\nabla \mathbf{u})_p \cdot \boldsymbol{\omega}_p$). As a consequence, this procedure implicitly considers an unbounded domain for the Poisson solution. In these methods, particle strength exchange schemes [41] or random-walk methods can be used for the computation of the diffusion term.

Despite the significant reduction of the computational cost achieved by parallel fast multipole (PFM) methods [111] ($\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$ compared to $\mathcal{O}(N^2)$ operations required by a direct evaluation of Biot-Savart's law, with N the number of particles), the evaluation of the velocity remains quite expensive when N grows very large.

In response to this, vortex particle-mesh (VPM) methods, also called Vortex-in-cell (VIC) in the literature, were introduced by Christiansen [23]. VPM methods essentially differ from classical Lagrangian vortex particle (VP) approaches by their numerical treatment of the spatial differential operators, i.e. the evaluation of the right hand side in Eq. (2.3). They are based on a combination of particles and an underlying grid, and therefore provide access to the numerously available grid-based Poisson solvers, which surpass purely PFM methods in terms of computational efficiency.

The present VPM solver has been initially developed by Cocle [28, 27] and was further improved by Lonfils [83] so as to account for solid bodies and to provide a multi resolution framework. The boundary condition for Eq. (2.2), on the outer boundary of the computational domain, as well as on the inner boundaries separating the subdomains of the parallel computation, are evaluated using a PFM method. The solution is then obtained in the entire computational domain using Fishpack [121, 122, 120], a fast finite difference Poisson solver based on cyclic reduction. The diffusion and the stretching terms are also both computed on the grid, using finite differences.

As in any Lagrangian method, the advected particles need to be frequently reinitialized in order to prevent clustering and depletion of the particles that is induced by local velocity gradients. It consists in replacing the distorted set of particles, after a few time steps, by new particles placed at the node positions of an underlying grid. The operation is called "redistribution" or "remeshing" and it helps maintaining an accurate representation of the vorticity field [71]. The high order kernel M_4' from [93] is used for both the redistribution and the particle-mesh interpolation operations. More details about the interpolation procedure (and the associated kernel) are provided in Chapter 6.

A second order time integration is performed, i.e. Adams-Bashforth for the diffusion and Leap-Frog for the displacement, or a Runge-Kutta (RK) scheme for both equations when the particles have been freshly redistributed.

The presence of the body is accounted for using an immersed boundary method. First, a vortex sheet is computed using a vortex panel method [58] (i.e. a boundary element method) so as to cancel the through flow velocity. Based on that, a vorticity flux is computed at the wall and the associated near-wall diffusion process is used to model the required no-slip condition for the velocity. The near-wall diffusion is performed by means of integral formulas developed in [103]. The origin of this model, and its application to 2-D flows, are further discussed in Chapter 3.

One of the issues encountered in 3-D vortex methods concerns the divergence of the vorticity field. Vorticity is solenoidal ($\nabla \cdot \boldsymbol{\omega} = 0$), as $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. Yet, the absolute value of the vorticity divergence may grow in time during the simulation and a regular reprojection of $\boldsymbol{\omega}$ is required.

Performing simulations on hierarchically refined grids is allowed using a multi resolution technique inspired from Bergdorf et al. [9, 10], which is based on average-interpolating wavelets [123].

2.2 Finite volume solver

An unstructured incompressible solver from the OpenFOAM open-source software library [1, 128] is used here. The Navier-Stokes equations are solved in the velocity-pressure formulation

$$\begin{aligned} \frac{D\mathbf{u}}{Dt} &= -\nabla P + \nu \nabla^2 \mathbf{u} \\ \nabla \cdot \mathbf{u} &= 0 \,, \end{aligned} \tag{2.4}$$

where $P \triangleq p/\rho$ is the reduced pressure, with p the pressure and ρ the density. The spatial discretization is performed using a cell-centered finite-volume approach and the PISO algorithm [61] is adopted in order to couple the velocity and the pressure at the time step level. The time integration is carried out using the implicit second order three-point backward differencing scheme

$$\left(\frac{\partial U}{\partial t}\right)^{n+1} = \frac{3U^{n+1} - 4U^n + U^{n-1}}{2\Delta t} + \mathcal{O}(\Delta t^2) \,,$$

where $(\cdot)^n$ indicates the evaluation of the function at the time $t^n \triangleq t^{n-1} + \Delta t$ (Δt is the time step). For the present calculations, a second order centered convection scheme is chosen. The velocity and the pressure equations are solved using respectively a preconditioned biconjugate gradient solver and a geometric-algebraic multigrid solver.

2.3 Hybrid FV-VPM solver

Vortex methods are particularly well-suited for the study of free vortical flows, thanks to the low numerical dissipation and dispersion errors, when the redistribution frequency is not too high (see Chapter 8). Thanks to the evaluation of the Poisson boundary condition using the PFM method, the VPM approach implicitly accounts for an unbounded domain and a compact computational domain tightly encompassing the vorticity support can be used.

Another feature of vortex methods is the absence of a CFL-like stability constraint (Courant-Friedrichs-Lewy condition) associated to the advection, as opposed to explicit Eulerian approaches. A Lagrangian condition must be satisfied by the time step Δt, though. The latter is based on the local strain of the flow and on the rotation of the particles with respect to each other

$$\|\mathbf{S}\|\Delta t < C_1 \quad \text{and} \quad \|\boldsymbol{\omega}\|\Delta t < C_2 \,,$$

where $\mathbf{S} \triangleq \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the strain rate tensor, $C_1 = 0.2, \ldots, 0.25$ and $C_2 = 0.2, \ldots, 0.25$, typically. Most of the time, this is less restrictive than a classical CFL condition and, strictly speaking, it does not represent a stability criterion but rather expresses a condition so as to prevent particle collision and hence ensure the accuracy of the particle-mesh interpolation.

Yet, accounting for bodies remains challenging, and properly capturing the boundary layers, especially at high Reynolds number, is computationally costly, as previously mentioned.

In a complementary fashion, body-fitted Eulerian methods provide a natural and efficient treatment for arbitrary solid walls, by allowing to work with anisotropic meshes in the vicinity of the walls. However, approaching the solution in an unbounded domain generally requires using a large computational domain.

Henceforth, the domain-based problem decomposition shown in Fig. 2.1 clearly benefits from the complementary strengths of both approaches: the near-wall flow is accurately captured by the finite volume solver, whereas the wake computation is performed by the vortex particle-mesh method.

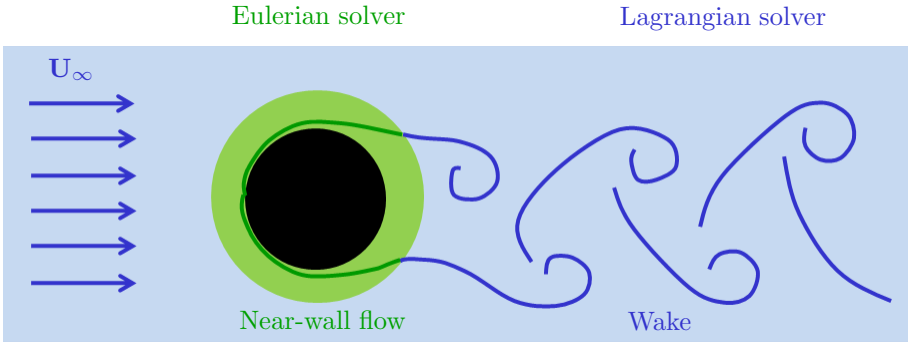


Figure 2.1: *Sketch of the different flow regions for a typical bluff-body flow.*

The present coupling approach is inspired from Daeninck [40] and, it is based on an overlapping domain technique. As shown in Fig. 2.2, this way of proceeding implies that the VPM solver actually computes the solution in the whole domain, i.e. up to the wall (the computational domain for the VPM solver is called Ω_{VPM} here, its outer boundary is $\partial\Omega_{\text{VPM}}$ and the body boundary is described by $\partial\Omega_b$). Yet, the solution is intentionally under resolved near the wall. Thanks to the panel solver, which has a global view of the flow, the correct amount of vorticity is still provided at the body boundary, despite the poorly captured near-wall physics. At a certain distance from the wall, and hence far from the attached boundary layers, the VPM solution is assumed to remain accurate, by means of the global character of the panel solver.

As a consequence, the VPM solution may serve as an outer boundary condition for the finite-volume solver, *without* requiring any Schwarz iteration in

order to match the VPM and FV solutions at the outer boundary $\partial\Omega_{\mathrm{FV}}$ of the FV computational domain Ω_{FV}. From the near-wall point of view, the VPM solver corresponds to a predictor and the FV solver plays the corrector role, in the sense that it subsequently recomputes the solution in Ω_{FV} so as to refine and improve the near-wall VPM vorticity in the correction domain $\Omega^c_{\mathrm{FV}} \subset \Omega_{\mathrm{FV}}$ (see Fig. 2.2).

By pushing this idea even further, one could equivalently consider the VPM solver as a means to provide an outer boundary condition for the Eulerian solver that is consistent with the far-field condition, even for a larger domain Ω_{FV}, i.e. a domain that is not limited to the near-wall region.

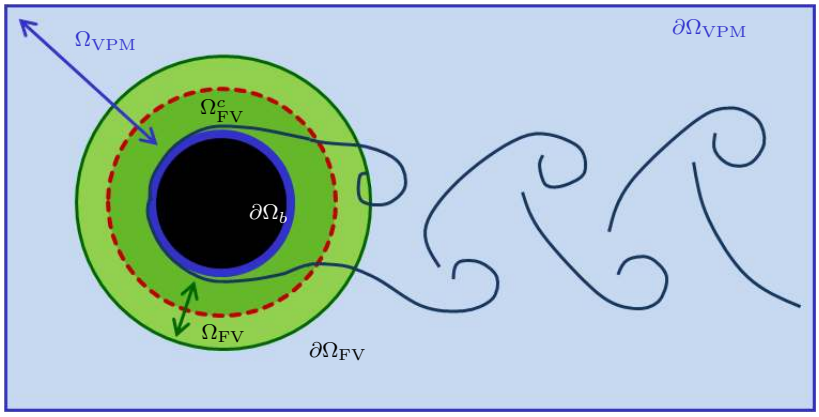


Figure 2.2: *Sketch of the decomposition using overlapping domains for the hybrid Eulerian-Lagrangian FV-VPM solver.*

Let us assume that, at the time t^n, the FV solution $\mathbf{u}^n$ and P^n is known in Ω_{FV} and the VPM particle vorticity $\boldsymbol{\omega}_p^n$ is available in Ω_{VPM}. The sequence of operations to compute the solution at the time t^{n+1} reads

1. Perform the explicit VPM computation and obtain $\boldsymbol{\omega}_p^{n+1}$ in $\Omega_{\mathrm{VPM}} \backslash \Omega^c_{\mathrm{FV}}$ and $\boldsymbol{\omega}_p^{n+1,*}$ in Ω^c_{FV}.

2. Compute the boundary condition $\mathbf{u}^{n+1}$ and $(\partial P/\partial n)^{n+1}$ on $\partial\Omega_{\mathrm{FV}}$ for the FV solver ($\partial/\partial n$ is the derivative in the direction normal to $\partial\Omega_{\mathrm{FV}}$).

3. Perform the implicit FV computation and obtain $\mathbf{u}^{n+1}$ and P^{n+1} in Ω_{FV}. The FV solution is retained as the proper solution in Ω^c_{FV}.

4. Correct the VPM near-wall vorticity by replacing $\boldsymbol{\omega}_p^{n+1,*}$ by $\boldsymbol{\omega}_p^{n+1} = (\nabla \times \mathbf{u})_p^{n+1}$ in Ω^c_{FV}.

As a consequence, both solvers communicate only twice per time step, since data is exchanged between the VPM subdomains and the FV partitions only at the coupling steps 2 (surface fields) and 4 (volume field).

Apart from the situation where the particles have just been redistributed, the coupling is second order accurate in time. The coupling is first order accurate in time for the time step following the redistribution, as no information is exchanged during the VPM RK2 predictor computation (remedying this shortcoming is not trivial, since the time integrators used for the VPM and FV solvers are different). For the other time steps, one VPM vorticity correction at the end of the time step is sufficient to ensure the accuracy, since only one right hand side evaluation is performed by the multi step time integrators.

One should notice that, contrary to appearances, the pressure must not be evaluated by the VPM solver in step 2. A projection of the Navier-Stokes Eq. (2.4) in velocity-pressure formulation onto the normal $\mathbf{n}$ yields the desired quantity on $\partial\Omega_{\mathrm{FV}}$, according to [40]

$$\frac{\partial P}{\partial n} = \mathbf{n} \cdot \left(\nu \nabla^2 \mathbf{u} - \frac{\partial \mathbf{u}}{\partial t} - \mathbf{u} \cdot \nabla \mathbf{u} \right) ,$$

where all terms are computed using second order finite differences. The results $\mathbf{u}^{n+1}$ and $(\partial P/\partial n)^{n+1}$ are then interpolated linearly onto the finite volume cell face centers. However, due the inconsistent discretizations between FV and VPM, and due to the interpolation, an error subsists on the incompressibility of $\mathbf{u}^{n+1}$ and for the Neumann boundary condition compatibility equation of the pressure Poisson equation resulting from the PISO algorithm (which is similar to the pressure equation from classical projection methods [69])

$$\nabla \cdot (\beta \nabla P) = \nabla \cdot \mathbf{u}^* ,$$

where $\mathbf{u}^*$ is an intermediate velocity field and β are numerical discretization coefficients related to the PISO algorithm (i.e. for the discretization of the convection term). A uniform correction is hence applied for the boundary conditions on $\partial\Omega_{\mathrm{FV}}$

$$\mathbf{u}^{n+1} \leftarrow \mathbf{u}^{n+1} - \frac{\mathbf{n}}{S_{\mathrm{FV}}} \int_{\partial\Omega_{\mathrm{FV}}} \mathbf{u}^{n+1} \cdot \mathbf{n} \, d\mathbf{x}$$

$$\left(\frac{\partial P}{\partial n} \right)^{n+1} \leftarrow \left(\frac{\partial P}{\partial n} \right)^{n+1} - \frac{\int_{\partial\Omega_{\mathrm{FV}}} \beta \left(\frac{\partial P}{\partial n} \right)^{n+1} \, d\mathbf{x}}{\int_{\partial\Omega_{\mathrm{FV}}} \beta \, d\mathbf{x}} ,$$

where $S_{\mathrm{FV}} = \int_{\partial\Omega_{\mathrm{FV}}} 1 \, d\mathbf{x}$ and since we impose that $\partial P/\partial n = 0$ on $\partial\Omega_b$.

The cell-averaged finite volume vorticity is computed as $\overline{\boldsymbol{\omega}}_e = (1/V_e) \sum_f \mathbf{n}_f \times \mathbf{u}_f \, S_f$, where V_e is the cell volume, the subscript e refers to the cell and f to one of its faces (S_f is the face area and $\mathbf{n}_f$ the related normal vector). A linear reconstruction of the vorticity is performed by computing the vorticity gradient based on the neighboring cell vorticity. The correction for the particles inside Ω^c_{FV} is computed as $\boldsymbol{\alpha}_p/h^3$ based on the reconstructed FV field and using a Gauss quadrature over the particles that are considered as cubes of side length h. The approach is thus not fully conservative, yet the interpolated field appears to be consistent and smooth across $\partial\Omega_{\mathrm{FV}}$.

As an alternative, one could also imagine considering the finite volume cells equivalently as particles and thus redistributing the associated vorticity intensity $\overline{\boldsymbol{\omega}}_e V_e$ to the nearest VPM particle. This would ensure the conservation but it would lead to a less smooth VPM vorticity field, which could affect the accuracy of the finite difference evaluation of the diffusion and stretching terms. Another conservative approach would require computing explicitly the intersections between the particles and the FV mesh [48, 47], but this would come at a very high cost, yet without significantly improving the methodology.

2.4 Results

The test case consists in simulating the flow past a sphere at $Re = U_\infty D/\nu = 300$, where D is the sphere diameter and U_∞ is the uniform upstream velocity field. First, a preliminary study compares the results obtained using the two solvers individually and secondly, some results are shown for the hybrid approach.

The simulation using the OpenFOAM finite volume solver was performed using the surface mesh shown in Fig. 2.3, extruded up to $r = 10D$ ($650K$ nodes and $1.3M$ cells). The first cells at the wall have a mesh size $\Delta y_{\mathrm{wall}}/D \simeq 0.004$ in the direction perpendicular to the wall and $\Delta x_{\mathrm{wall}} \simeq 5\Delta y_{\mathrm{wall}}$ in both directions parallel to the wall. The surface mesh is extruded using a geometric progression of common ratio 1.1. The boundary conditions are given by $\mathbf{u} = \mathbf{U}_\infty$ and $\partial P/\partial n = 0$ on the outer upstream half sphere, by $\partial \mathbf{u}/\partial n = 0$ and $P = 0$ on the outer downstream half sphere and by $\mathbf{u} = 0$ and $\partial P/\partial n = 0$ on the wall. The time step is $U_\infty \Delta t/D = 1 \cdot 10^{-2}$.

Figure 2.3: *Prismatic mesh used for the OpenFOAM finite volume computation (it has been obtained as an extruded surface mesh from a cube projected onto a sphere).*

The results for the vortex particle-mesh simulation are reported from Lonfils [83]. A multi resolution technique was used with a near-wall mesh size $h/D = 1/75 = 0.0133$, and the domain extended up to $38D$ downstream of the sphere, which amounts to a total of approximately $1.1 \cdot 10^7$ grid points. The time step was $U_\infty \Delta t/D = 8.5 \cdot 10^{-3}$ and a particle redistribution was carried out every 5 time steps. Fig. 2.4 is reproduced from [83] and shows the near-wall flow, along with the hierarchically refined grid.

Table 2.1 compares the FV solver and the VPM solver results, together with some reference results from the literature, in terms of the lift coefficient C_L, the drag coefficient C_D and the Strouhal number St, defined by

$$C_D \triangleq \frac{\mathbf{F} \cdot \hat{\mathbf{v}}}{\frac{1}{2}\rho U_\infty^2 \pi R^2} \qquad C_L \triangleq \frac{\|\mathbf{F} - \mathbf{F} \cdot \hat{\mathbf{v}}\|}{\frac{1}{2}\rho U_\infty^2 \pi R^2} \qquad St \triangleq \frac{fD}{U_\infty} \,,$$

where $\mathbf{F}$ is the force exerted by the flow on the sphere, $\hat{\mathbf{v}}$ is the upstream flow direction ($\mathbf{U}_\infty = U_\infty \hat{\mathbf{v}}$), f is the frequency associated to the mode with the highest amplitude of the lift fluctuation and $R = D/2$ is the sphere radius. Results show an excellent agreement between the FV solver and the VPM solver, which further supports the intention to couple both approaches.

Concerning now the hybrid Eulerian-Lagrangian approach, the domain Ω_{FV} extends up to $r = 2.44R$ and the associated mesh consists of $390K$ nodes and $810K$ cells. The surface mesh is the same as that shown in Fig. 2.3; the first cells at the wall have a mesh size $\Delta y_{\mathrm{wall}} \simeq 0.002D$ in the direction perpendicular

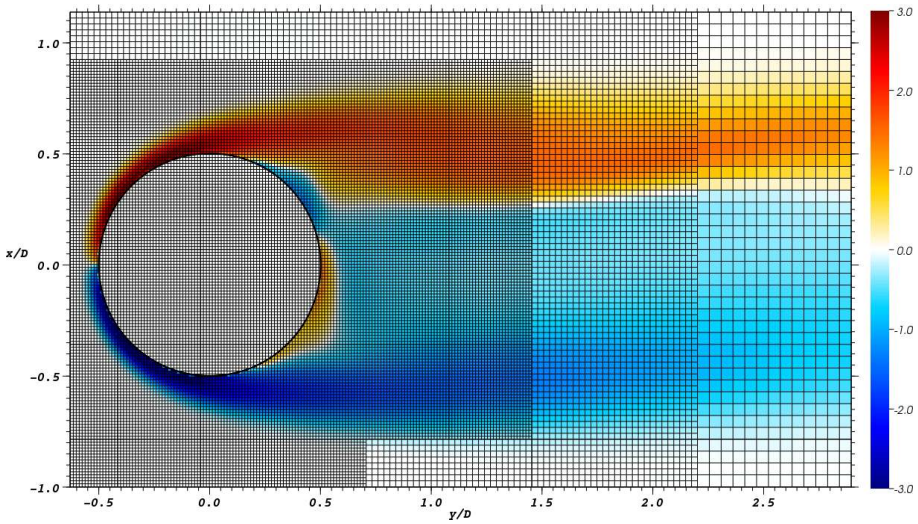


Figure 2.4: *Flow past a sphere at $Re = 300$ using the multi resolution VPM solver from Lonfils [83]: the vorticity measure $sign(\omega_z)\log\left(1+\frac{D}{U_\infty}|\omega_z|\right)$ is shown here, along with the underlying grid; the figure is reproduced from Lonfils [83].*

	$\overline{C}_D$	$\overline{C}_L$	St
Present FV	0.666	0.070	0.136
Lonfils VPM [83]	0.677	0.070	0.134
Ploumhans et al. [103]	0.683	0.061	0.135
Georges [52]	0.661	0.066	0.134
Johnson & Patel [64]	0.656	0.069	0.137
Constantinescu et al. [31]	0.655	0.065	0.136

Table 2.1: *Time averaged drag, time averaged lift and Strouhal number for the flow past a sphere at $Re = 300$; comparison between the FV solver, the VPM solver and other reference results from the literature.*

to the wall and $\Delta x_{\text{wall}} \simeq 10\Delta y_{\text{wall}}$ in both directions parallel to the wall. The correction zone Ω^c_{FV} is defined by $R \leq r \leq 1.6R$.

In the present case, a uniform grid is used for the VPM solver with again $h/D = 1/75 = 0.0133$ and the domain Ω_{VPM} extends up to $20D$ downstream of the sphere with an outflow condition at the outflow plane corresponding to a "through flow plane" or "dominating drag condition" (odd symmetry for the normal component of $\boldsymbol{\omega}$ across the outflow plane and even symmetry for the components tangent to the outflow plane, see [28]). A comparison between the

FV mesh and the VPM grid is shown in Fig. 2.5. It further shows that the Lagrangian grid is clearly less adapted to capture the near-wall flow, compared to the FV mesh.

The time step is given by $U_\infty \Delta t/D = 5 \cdot 10^{-3}$ and redistribution in the VPM solver is done every 4 time steps.

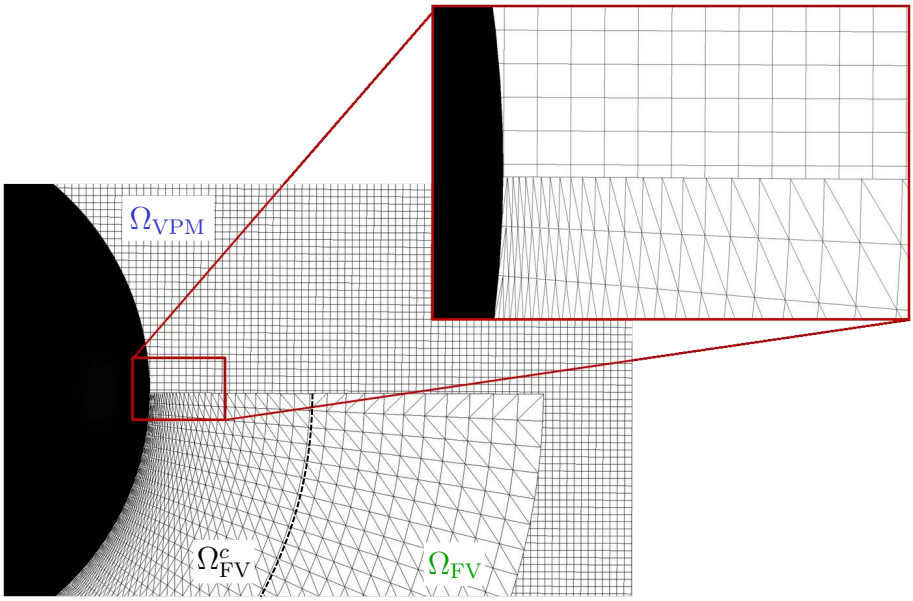


Figure 2.5: *Meshes used for the hybrid Eulerian-Lagrangian computation : near-wall comparison between the VPM grid and the FV prismatic mesh (only the bottom half of the FV mesh is displayed here; the diagonal lines of the quadrangles appearing in the FV mesh are spurious and correspond to a bug inside the visualization software).*

Fig. 2.6 shows the iso contours of the vorticity magnitude computed by the FV solver inside Ω^c_{FV} and the VPM solution inside $\Omega_{\mathrm{VPM}} \backslash \Omega^c_{\mathrm{FV}}$. Despite the different discretization approaches adopted in both subdomains, the contours appear to match across the correction domain boundary $\partial\Omega^c_{\mathrm{FV}}$, except for some very small discrepancies.

The near-wall vorticity from the FV and VPM solutions are also compared in Fig. 2.7. Note that, in the present case, the VPM solution is actually quite well resolved, as the grid size h is similar to the FV mesh size parallel to the wall (see Fig. 2.5). Therefore, the correction domain Ω^c_{FV} could be narrower, considering that inside Ω^c_{FV} the most distant FV cells from the wall are roughly twice as big as the VPM particles. Nevertheless, the correction provided by the FV solution helps maintaining a good representation of $\boldsymbol{\omega}$ in Ω_{VPM} all

the way up to the wall, as it accounts for the large gradient in the direction perpendicular to the wall, which are not well captured by the VPM approach (the noise observed in the VPM solution near the wall is partially due to the graphical interpolation of the solution into the RGB color space). Yet, one could argue that the particles in the direct vicinity of the wall are not corrected properly, as the quadrature points for the integration only "see" a limited fraction of the near-wall FV cells, due to the small size of the cells in the direction perpendicular to the wall.

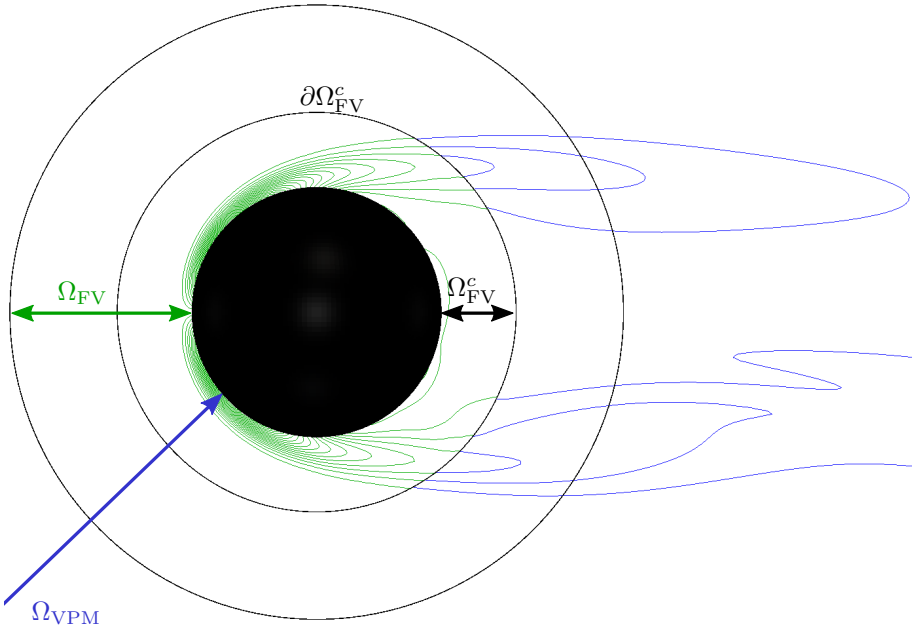


Figure 2.6: *Iso contours of $\|\boldsymbol{\omega}\|D/U_\infty$ (by steps of 1.5) for the flow past a sphere at $U_\infty t/D = 45$ and for $Re = 300$: Eulerian FV solution in Ω^c_{FV} (green contours), Lagrangian VPM solution in $\Omega_{VPM}\backslash\Omega^c_{FV}$ (blue contours) and representation of the domains Ω_{VPM}, Ω_{FV} and Ω^c_{FV} ($\partial\Omega^c_{FV}$ is the outer boundary of Ω^c_{FV}).*

Fig. 2.8 shows the drag coefficient C_D resulting from the application of the hybrid scheme. It is computed by using two different force evaluation techniques. The first technique is typically used for VPM methods and it consists in evaluating the force by computing surface integrals over a control volume containing the body, according to [96, 97] (the surfaces of the control volume are thus included in Ω_{VPM}). The second technique requires the FV

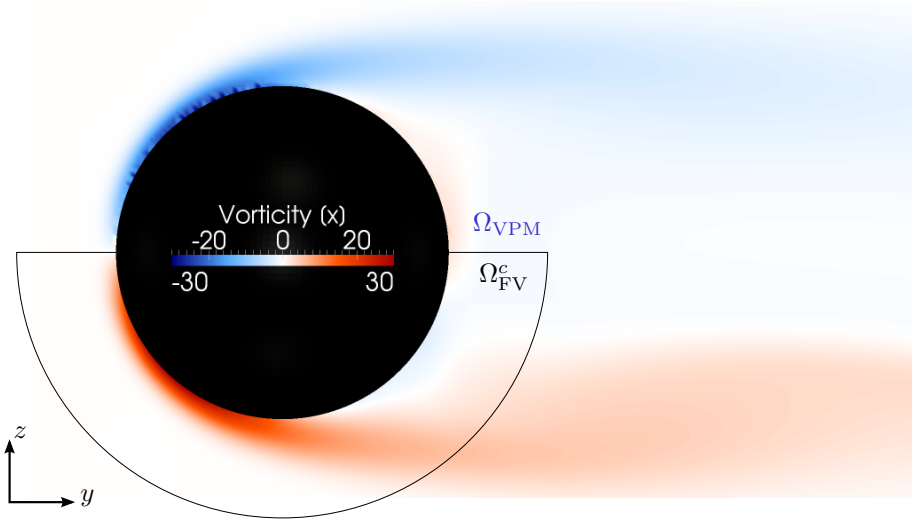


Figure 2.7: *Vorticity $\omega_x D/U_\infty$ for the flow past a sphere at $U_\infty t/D = 45$ and for $Re = 300$: comparison of the near-wall vorticity between the VPM solution in Ω_{VPM} and the FV solution in Ω^c_{FV} (only the bottom half of Ω^c_{FV} is displayed here).*

solution (pressure and friction), as it is based on a simple force integration over the surface of the body.

One may observe that strong oscillations are exhibited at half of the sampling frequency for the drag evaluated by the second technique, based on the FV solution. The amplitude is of the order of 5 to 10% of the total drag (2 to 3% for the drag based on the VPM solution). Yet, the average of the drag is quite in agreement with those of Table 2.1. Note that, a closer look at the VPM drag signal presented in Lonfils [83] (thus without hybrid scheme) also reveals some spurious oscillations. Using a near-wall FV solver in the framework of the hybrid scheme seems to worsen this phenomenon.

The repeated pattern matches the redistribution frequency. Considering the fact that the boundary layer is nearly steady at this Reynolds number, the flow does not significantly change between two redistributions. Hence, the advection of the particles and the subsequent redistribution should not overly alter the grid representation of the vorticity field near the wall, which however seems to be the case here, since the drag fluctuates. This observation may suggest that something goes wrong during the redistribution.

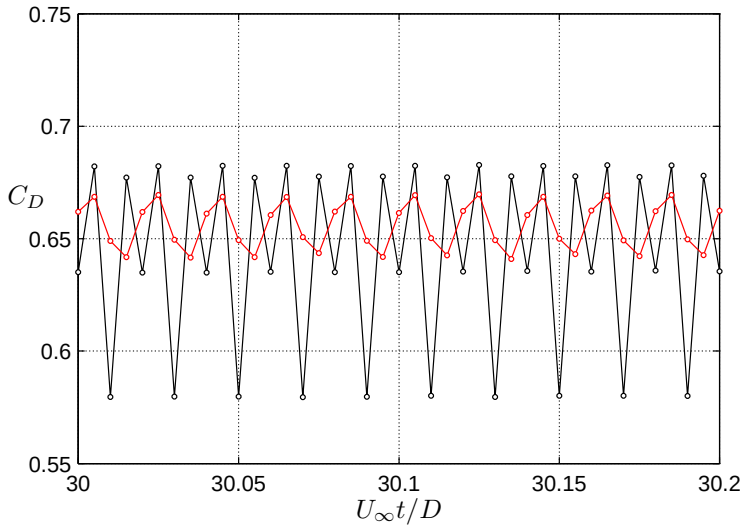


Figure 2.8: *Drag coefficient obtained for the flow past a sphere at $Re = 300$ using the hybrid scheme: drag computed from the FV domain (black curve) and from the VPM domain (red curve).*

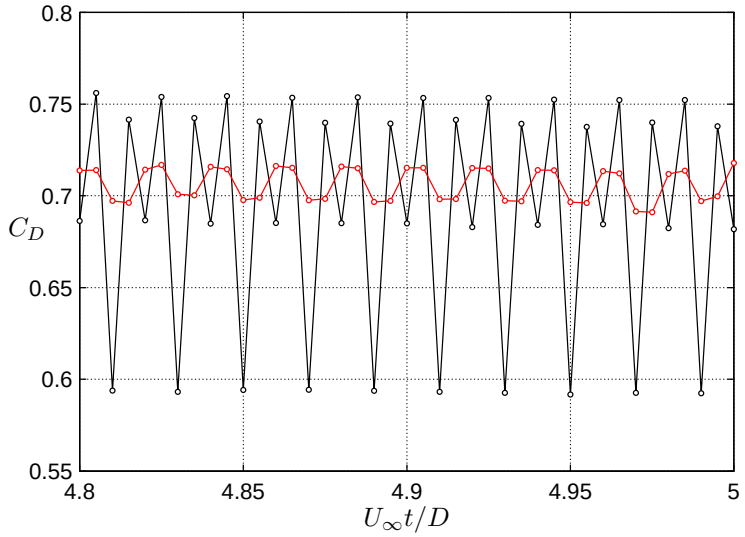
While not reported here, some tests were also performed at different redistribution frequencies. The pattern is obviously affected by the redistribution frequency, yet the amplitude of the oscillation remains similar. One may moreover argue that using a different time integration schemes just after the redistribution is inconsistent (Runge-Kutta 2 vs. Adams-Bashforth 2 and Leap-Frog). However, redistributing the particles every time step (and thus using the RK2 time integrator for every time step) does not reduce the amplitude of the drag signal much.

The oscillatory behavior of the drag may further be related to the linear impulse associated with the vortex panels used in the VPM solver. This is measured by computing a "panel-induced drag" defined by

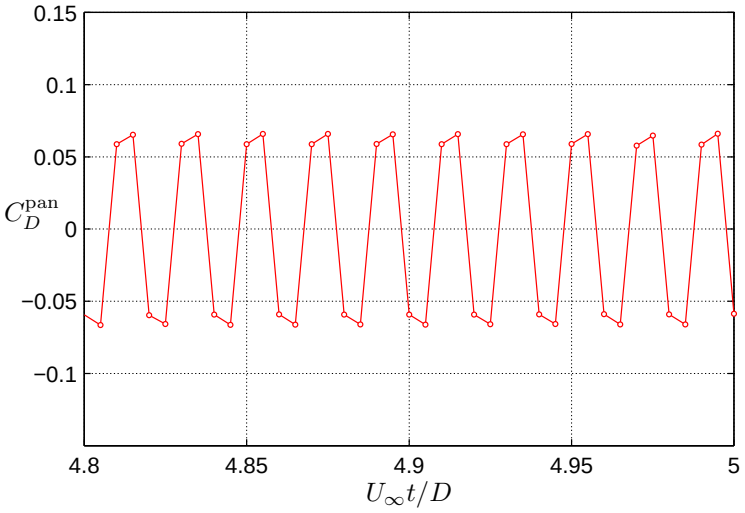
$$C_D^{\text{pan}} \triangleq -\left(\hat{\mathbf{v}} \cdot \frac{d\mathbf{I}^{\text{pan}}}{dt}\right) \frac{1}{\frac{1}{2}\rho U_\infty^2 \pi R^2} ,$$

where $\mathbf{I}^{\text{pan}} \triangleq \int_{\partial\Omega_b} \mathbf{x} \times \Delta\boldsymbol{\gamma}\, d\mathbf{x}$ is the linear impulse induced by the vortex panels of intensity $\Delta\boldsymbol{\gamma}$. Fig. 2.9 shows the drag for a slightly different computational setup compared to the previous setting (basically, there is no symmetric outflow condition and the vortex panels do not diffuse in this case). One may clearly observe that the drag solution is correlated to the evolution of the panel-induced

drag. This observation suggests a consistency problem in the no-slip enforcement, between the vortex panel method and the finite difference-based VPM approach (Poisson solver, diffusion term, etc.).



(a)



(b)

Figure 2.9: *Drag coefficient obtained for the flow past a sphere at $Re = 300$ using the hybrid scheme (numerical setup without symmetric outflow condition and with vortex panels that do not diffuse): (a) drag computed from the FV domain (black curve) and from the VPM domain (red curve); (b) panel induced drag that is related to the VPM approach.*

Similarly to Fig. 2.8, the oscillation is more important in the FV solution, when one considers the associated drag evaluation, as can be seen in Fig. 2.9(a). Indeed, significant pressure variations are generated in the FV solution inside Ω_{FV} and they represent the major contribution to the drag oscillation compared to the friction (it is not reported here). This further indicates that the incompressible finite volume solver is quite sensitive to the outer boundary condition provided by the VPM solver, which does not help matters.

2.5 Conclusion

The results from Lonfils [83] that are reported here show that the immersed boundary method is well-suited for the simulation of bluff body flows using a VPM solver. Yet, as was already mentioned in [83], some oscillations occur in the force diagnostics, especially for the drag coefficient. These oscillations are not problematic, unless the VPM approach is further coupled to a near-wall Eulerian solver. According to the present results, the oscillation amplitude then reaches 5 to 10% of the average drag when the hybrid scheme is used.

Based on the above observations, it is certainly not easy to distinguish the cause from the consequence for this spurious oscillatory behavior. However, they hint at a consistency problem existing between the different computational operations performed inside the VPM approach, i.e. the use of a vortex panel method in conjunction with a finite difference-based evaluation of the spatial differential operations.

A further point that may support this assertion consists in observing that hybrid Eulerian-Lagrangian approaches were successfully developed for purely Lagrangian vortex methods [98, 40] and for VPM methods using body-fitted grids in relatively simple computational domains [98]. Accounting for more complicated body geometries that lead to arbitrary intersections of the body boundary with the grid may be more problematic when combining a vortex panel method with classical grid techniques.

More precisely, one may distinguish the following operations contributing to the inconsistency of the VPM solver:

- Computing a velocity field satisfying a no-through flow condition at the wall (Poisson solver + vortex panel method).
- Near-wall diffusion of the generated vortex sheet so as to enforce the no-slip condition at the wall.

- Redistribution of the particles near the wall and particle-mesh interpolation.

Yet, it is a priori not clear if this inconsistency is the reason for the drag oscillation. Next to this track, some other investigations could maybe overcome this difficulty, by improving for example the coupling technique and making use of an extended buffer layer. Another possibility could consist in using a different near-wall solver, that is perhaps less sensitive to the outer boundary condition.

The present work however focuses on the first track and lies the basis for searching for a more consistent approach, relying on immersed interface techniques [77] in order to perform all the aforementioned operations on the grid. The finite difference schemes are modified so as to account for the presence of the body and the spatial accuracy is preserved up to the wall.

The previous combination of the vortex panel method with the standard finite difference Poisson solver is replaced by a novel approach based on corrected finite differences. The solution satisfying both the far-field condition and the no-through flow condition at the body boundary is computed all at once (see Chapter 4). Moreover, the other operations required by the VPM approach, such as the diffusion operator with a flux condition at the wall for the no-slip enforcement (Chapter 5) or the particle-mesh interpolation (Chapter 6), also account for the presence of the body in a consistent manner.

Chapter 3

Enforcing a no-slip condition in a VPM method and application to a vortex dipole in a box

The aim of this chapter is twofold. First, in order to complete our brief introduction of Chapter 2, the vortex particle-mesh approach is here described more thoroughly. The necessary background for the following chapters is provided here, by detailing the different computational steps that are required in a 2-D VPM method. Secondly, the specific aspect of enforcing a no-slip condition at a solid wall is further discussed and investigated.

As a matter of fact, the simulation of wall-bounded flows using a vortex method remains challenging and it requires several modifications in the solution algorithm, compared to the simulation of free vortical flows. The way to account for solid bodies inside the flow domain has been studied extensively in the past decades since the pioneering work of Lighthill [79] in 1963.

After the description of the general VPM methodology, this chapter reviews and quantifies the performance of one specific technique to enforce a no-slip condition, that was introduced by Koumoutsakos et al. [72] and Cottet [33], and which is based on a vorticity flux evaluation. The main drawback here consists in the fact that it does not provide a high order temporal accuracy for

the treatment of the associated no-slip condition, as it is intrinsically linked to a fractional step algorithm. While the present work does not overcome this issue, an analysis about the splitting of the equations is carried out and some suggestions are made in order to improve the flux computation. The intent is here to gather some new ideas so as to possibly inspire future developments.

Studying the performance of the no-slip enforcing procedure calls for a well-defined (and simple) numerical framework and the present test case (vortex dipole in a cavity) just fulfills these requirements. The geometrical treatment of the walls is indeed greatly simplified, as the cavity walls coincide with the grid boundaries. Hence, the numerical errors due to possibly arbitrary intersections of the body boundary with the grid (that are usually encountered in immersed interface methods, i.e. the core business of the following chapters) are avoided. This test case is therefore particularly well-suited for the study and the validation of different procedures aiming at enforcing a no-slip condition at solid walls in vortex methods. As a consequence, this setup can be considered as a "sand box" problem for the implementation and the testing of various numerical techniques related to VPM methods.

First, Section 3.1 describes the chosen test case and, next, Section 3.2 details the VPM methodology by depicting a solver designed for the computation of viscous flows with a no-through flow condition at the wall (i.e. an inviscid wall), which is naturally accounted for in this type of method. Section 3.3 then treats the less straightforward aspect of enforcing a no-slip condition at the wall and thus, accordingly, of canceling at every time step the tangential slip velocity resulting from the solver of Section 3.2. Finally, some results are provided in Section 3.4 for the vortex dipole flow for the case at $Re = 1000$.

3.1 Description of the test case

The present test case, consisting of a self-propelling 2-D vortex dipole colliding with the solid walls of a square cavity, has been initially studied in [26] and was further formalized and thoroughly documented in [25], for benchmarking purposes. The underlying rationale is the study of the resulting wall-vortex interactions and the associated generation of secondary vorticity coming from the wall. Despite the simple problem geometry, the complex flow physics occurring after the collision makes this test case challenging from a numerical point of view.

The initial condition is the combination of two opposite sign vortices, each of zero total circulation. It reads

$$\begin{aligned}\omega(\mathbf{x},0)=\omega_0(\mathbf{x})= & \quad \omega_e\left(1-\left(\frac{r_1}{r_0}\right)^2\right)\exp\left(-\left(\frac{r_1}{r_0}\right)^2\right)\\ - & \quad \omega_e\left(1-\left(\frac{r_2}{r_0}\right)^2\right)\exp\left(-\left(\frac{r_2}{r_0}\right)^2\right),\end{aligned} \tag{3.1}$$

with $r_1 \triangleq |\mathbf{x}-\mathbf{x}_1|$ and $r_2 \triangleq |\mathbf{x}-\mathbf{x}_2|$. Everything is here dimensionless. The domain is $\Omega=[-1,1]\times[-1,1]$ and the numerical parameters are

$$\begin{aligned}\omega_e &= 299.528385375226\\ r_0 &= 0.1\\ \mathbf{x}_1 &= (0,+0.1)\\ \mathbf{x}_2 &= (0,-0.1)\,.\end{aligned}$$

The value of ω_e been computed, so as to enforce that the initial kinetic energy $E(t=0)=\frac{1}{2}\int_\Omega |\mathbf{u}|^2\,d\mathbf{x}=2$. The initial condition is shown in Fig. 3.1. A no-slip condition is enforced on the domain boundary $\partial\Omega$, which is compatible with the velocity induced by the initial vorticity field, that is equal to zero, as each of the vortices has a zero total circulation[1]. The Reynolds number is defined as

$$Re \triangleq \frac{1}{\nu}\,.$$

3.2 VPM solver with a no-through flow condition at the wall

In the context of vortex methods, one of the main difficulties resides in enforcing a no-slip condition at the wall. As a first step towards this objective, the present section provides a numerical tool to solve the 2-D Navier-Stokes equations in vorticity-velocity formulation with a *no-through flow* condition at the wall $\partial\Omega$,

[1]Note that the velocity is not exactly equal to zero at the solid walls, as the vorticity support is actually infinite. Yet, the decay far from the dipole is exponential and the error is therefore negligible. If we had used instead a dipole made of realistic vortices, thus each with non-zero total circulation, the far-field velocity would only have decayed as $1/r^2$.

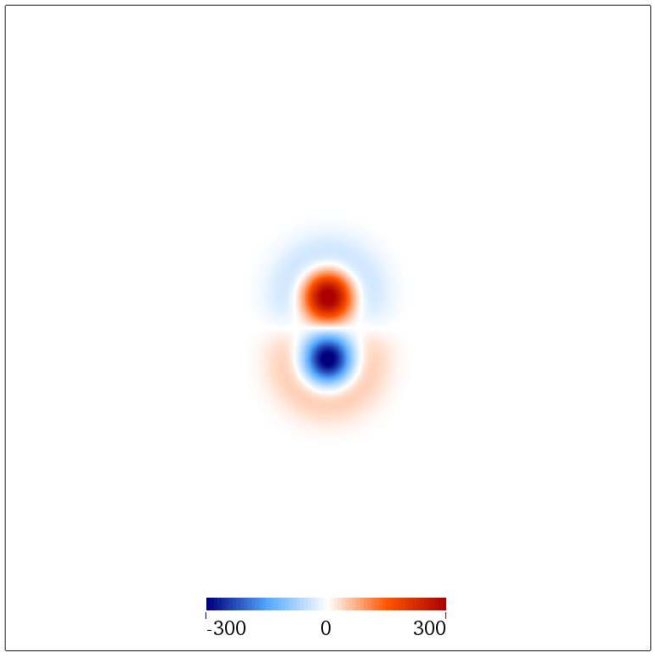


Figure 3.1: *Initial condition ω_0 for the "dipole in a box" test case.*

which is indeed more easily accounted for in vortex methods. We thus consider

$$\begin{aligned} \frac{D\omega}{Dt} &= \nu\nabla^2\omega \qquad \text{with} \quad q_\omega \triangleq -\nu\frac{\partial\omega}{\partial n} = 0 \quad \text{on } \partial\Omega \\ \mathbf{u} &= \nabla\times(\Psi\hat{\mathbf{e}}_z) \\ \nabla^2\Psi &= -\omega \qquad \text{with} \qquad \mathbf{u}\cdot\mathbf{n} = 0 \quad \text{on } \partial\Omega\,, \end{aligned} \tag{3.2}$$

with ω the vorticity, $\mathbf{u}$ the velocity, Ψ the streamfunction, ν the kinematic viscosity and $\mathbf{n}$ the inward pointing normal vector (see Fig. 3.2). As will appear in Section 3.3, the vorticity flux q_ω is associated to the no-slip enforcement; it aims at canceling the spurious tangential slip velocity remaining at the wall after the solution has been advanced using a no-through flow condition. Setting $q_\omega = 0$ is equivalent to considering an "inviscid wall", i.e. no vorticity enters the fluid at the wall and a tangential slip velocity is allowed.

As the name suggests, VPM methods make use of a set of particles and a grid in order to compute the flow field. On the one hand, the time evolution is computed in a Lagrangian fashion by following the particles and, on the other

hand, all spatial differential operations, such as for example the solution of the Poisson equation,

$$\nabla^2 \Psi = -\omega \ , \tag{3.3}$$

are performed on the underlying grid. The time evolution of a particle p at $\mathbf{x}_p(t)$, moving at a velocity $\mathbf{u}_p(t) \triangleq \mathbf{u}(\mathbf{x}_p(t), t)$ and carrying some vorticity $\omega_p(t) \triangleq \omega(\mathbf{x}_p(t), t)$ is given by

$$\begin{aligned} \frac{d\mathbf{x}_p}{dt} &= \mathbf{u}_p \\ \frac{d\omega_p}{dt} &= \nu \left(\nabla^2 \omega\right)_p \ , \end{aligned}$$

where $\left(\nabla^2 \omega\right)_p (t) \triangleq \nabla^2 \omega \left(\mathbf{x}_p(t), t\right)$.

The computational domain Ω is a square defined by $[-L/2, L/2] \times [-L/2, L/2]$ and the grid consists of $N \times N$ cell-centered nodes $\mathbf{x}_{ij} = (x_i, y_j)$, with a uniform mesh spacing $h \triangleq \Delta x = \Delta y = L/N$, as shown in Fig. 3.2. For the following, the notation for a grid field f at time t^n is $f_{ij}^n \triangleq f(\mathbf{x}_{ij}, t^n)$ and the field f carried by a particle at $\mathbf{x}_p^n \triangleq \mathbf{x}_p(t^n)$ is called $f_p^n \triangleq f(\mathbf{x}_p^n, t^n)$.

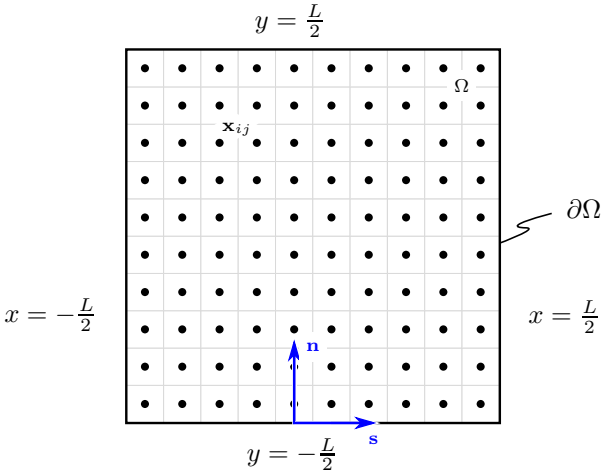


Figure 3.2: *Sketch of the computational domain Ω and of the grid (nodes are represented as bullets); definition of the normal vector $\mathbf{n}$ and of the tangential vector $\mathbf{s}$.*

The time integration is performed here using a mid-point second order Runge-Kutta scheme (RK2). A particle redistribution is operated every $(n^r)^{\text{th}}$ time step (the distorted set of particles at $\mathbf{x}_p$ is then replaced by a new set of particles whose positions coincide with the grid nodes $\mathbf{x}_{ij}$; see Chapter 6 for more details about this operation). The computation at t^n starts with

the particle field ω_p^n and the associated grid field ω_{ij}^n. Considering, for the moment, the more general case, where the flux $q_\omega^n \triangleq q_\omega(\mathbf{x}, t^n)$ for $\mathbf{x} \in \partial\Omega$ ($q_\omega^{n+\frac{1}{2}} \triangleq q_\omega(\mathbf{x}, t^{n+\frac{1}{2}})$) is prescribed, the time integration using a time step Δt consists of the two following sub steps:

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,\mathbf{u}_p^n$$

- *Diffusion :*
$$\omega_{ij}^n \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^n \quad \text{with} \quad -\nu \left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^n = q_\omega^n$$
$$\left(\nabla^2\omega\right)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2}} = \omega_p^n + \frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \mathbf{u}_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\,\mathbf{u}_p^{n+\frac{1}{2}}$$

- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu \left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n+\frac{1}{2}} = q_\omega^{n+\frac{1}{2}}$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1} = \omega_p^n + \Delta t\,\nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles.

The inviscid wall corresponds to the case $q_\omega^n = q_\omega^{n+\frac{1}{2}} = 0$. Appendix E gives the computational algorithm for the low storage third order Runge-Kutta scheme from [130] (RK3). The different operations that are performed during the time integration are detailed hereafter. Section 3.2.1 explains the solution process of Eq. (3.3) and Section 3.2.2 describes the diffusion and the particle-

mesh interpolation (P2M stands for particles-to-mesh interpolation, whereas M2P stands for mesh-to-particles interpolation; see Chapter 6). Some time convergence study results are then presented for the RK2 and RK3 schemes in Section 3.2.3, with the purpose of subsequently analyzing, in Section 3.3, the impact on the convergence of adding a no-slip enforcing procedure to the present solver.

By anticipating the content of the following chapters aiming at developing an immersed interface VPM solver, one should observe that the required computational steps are basically the same as those presented here.

3.2.1 Computation of the velocity

The Poisson Eq. (3.3) is solved using a no-through flow condition at the cavity walls for the velocity field (or equivalently, a no-slip condition is enforced below an infinitely thin vortex sheet with a jump of tangential velocity across the sheet). The corresponding boundary condition for the streamfunction is $\Psi = \overline{\Psi} = cst$, since $\mathbf{u} \cdot \mathbf{n} = -\partial\Psi/\partial s = 0$ along the body boundary (with $\mathbf{s}$ the tangential vector to the wall, as shown in Fig. 3.2). The constant $\overline{\Psi}$ is arbitrary as the flow domain is simply-connected, and we may thus set $\overline{\Psi} = 0$ without loss of generality (any other value of $\overline{\Psi}$ would lead to the same velocity field). For an unbounded multiply-connected flow domain (i.e. with multiple bodies), as will be considered in Chapter 4 in the framework of immersed interface methods, this constant is implicitly fixed at infinity, hence $\overline{\Psi}$ is not arbitrary on the body boundary and it is linked to the circulation of the body.

Eq. (3.3) is solved on the grid using the FFT algorithm [49], along the lines of the Fourier-based VPM solvers from [16, 17, 50]. Due to the homogeneous Dirichlet condition, the streamfunction Ψ is odd across the domain boundaries in the x and y directions (see Fig. 3.3). As the Laplacian preserves the symmetry properties, so should be the vorticity ω. A discrete sine transform (DST) taking into account the $h/2$ offset between the domain boundary $\partial\Omega$ and the grid nodes is performed accordingly on ω_{ij} (forward) and Ψ_{ij} (backward), so as to enforce the homogeneous Dirichlet condition on $\partial\Omega$.

The velocity field $\mathbf{u} = (u, v)$ is also computed spectrally from Ψ. Its symmetry properties are different, by construction : u is odd along x and even along y, whereas v is even along x and odd along y. Discrete sine (DST) and cosine (DCT) transforms are used accordingly, and the velocity component normal to each wall is thus explicitly canceled.

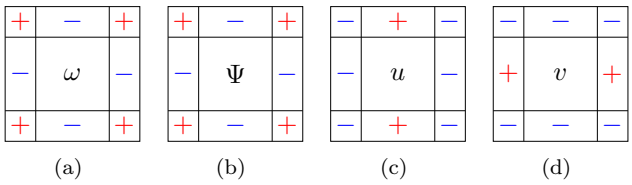


Figure 3.3: *Symmetries of the different fields taking part in the computation of the velocity field by solving Eq. (3.3) using an FFT solver: even symmetry (plus sign) and odd symmetry (minus sign).*

3.2.2 Diffusion and particle-mesh interpolation

The term $\nabla^2\omega$ is computed on the grid using second order finite differences for the RK2 integration scheme and fourth order finite differences for the RK3 integration scheme from Appendix E (the way to perform the evaluation of $\nabla^2\omega$ at grid nodes residing in the vicinity of immersed interfaces is explained in Chapter 5). According to the previous observation, and using ghost grid values so as to enforce $q_\omega = 0$ on $\partial\Omega$, an even extension of ω is computed across the wall, as shown in Fig. 3.4. It should be observed that this extension departs from the oddness required for ω in Eq. (3.3). There, the odd extension is just a numerical maneuver in order to provide a no-through flow condition at the wall, as the vorticity may be different from zero on $\partial\Omega$, in general.

The M2P interpolations (mesh-to-particles), taking place in the corrector sub step, also require extensions when applying the 2-D M_4' interpolation scheme [93] on $\nabla^2\omega$ and on the velocity $\mathbf{u}$ (see Chapter 6 for more details about the interpolation). The term $\nabla^2\omega$ has the same symmetry properties as ω for the diffusion, i.e., it is also even. The velocity extension has the same symmetry as previously in Section 3.2.1, namely odd in the velocity component normal to the wall and even in the tangential component.

The even symmetry for ω is also imposed for the P2M interpolation. Any particle in the vicinity of the wall with an M_4' interpolation range affecting grid nodes outside of the flow domain is assigned an image particle of identical vorticity and placed symmetrically on the other side of the wall. This approach conserves the circulation, as the wall lies exactly in between two adjacent nodes. From an algorithmic point of view, this is equivalent to first redistributing the particles onto the grid without considering the wall and, secondly, to "folding back" the vorticity outside of the domain into the flow (first in one grid direction and then in the other one, see Fig. 3.5). The M2P and P2M ghost computation

for immersed interfaces is detailed in Chapter 6.

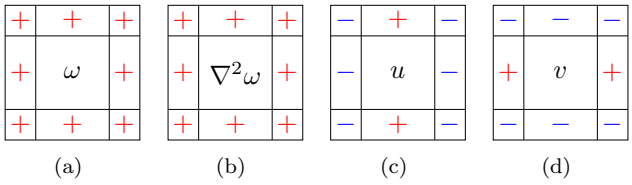


Figure 3.4: *Symmetries of the different fields required for the diffusion term $\nabla^2\omega$ and the particle-mesh interpolation (M2P and P2M), according to the above time integration.*

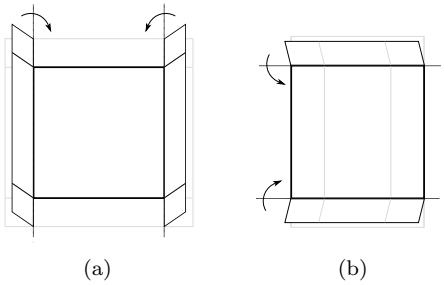


Figure 3.5: *P2M interpolation with a no-flux condition at the wall: "folding back" the vorticity inside the flow.*

3.2.3 Time convergence study

The time convergence of the method is studied by computing the solution of Eq. (3.2) with the initial condition Eq. (3.1) at $Re = 1000$, using both the RK2 and RK3 no-through flow solvers, on a 512×512 grid. The following error norms are computed at $t = 0.2$ (before the collision of the vortex dipole with the wall, see Fig. 3.6(a)) and at $t = 0.4$ (after the collision, see Fig. 3.6(b))

$$\epsilon_2 \triangleq \|\epsilon\|_2 \triangleq \frac{2h}{\omega_e L}\left(\sum_{i,j}\left(\omega_{ij} - (\omega_{\text{ref}})_{ij}\right)^2\right)^{\frac{1}{2}}$$

$$\epsilon_\infty \triangleq \|\epsilon\|_\infty \triangleq \frac{1}{\omega_e}\max_{i,j}\left|\omega_{ij} - (\omega_{\text{ref}})_{ij}\right| \ ,$$

with ω_{ij} the solution computed using a time step Δt and $(\omega_{\text{ref}})_{ij}$ the reference solution computed using a time step Δt_{ref}.

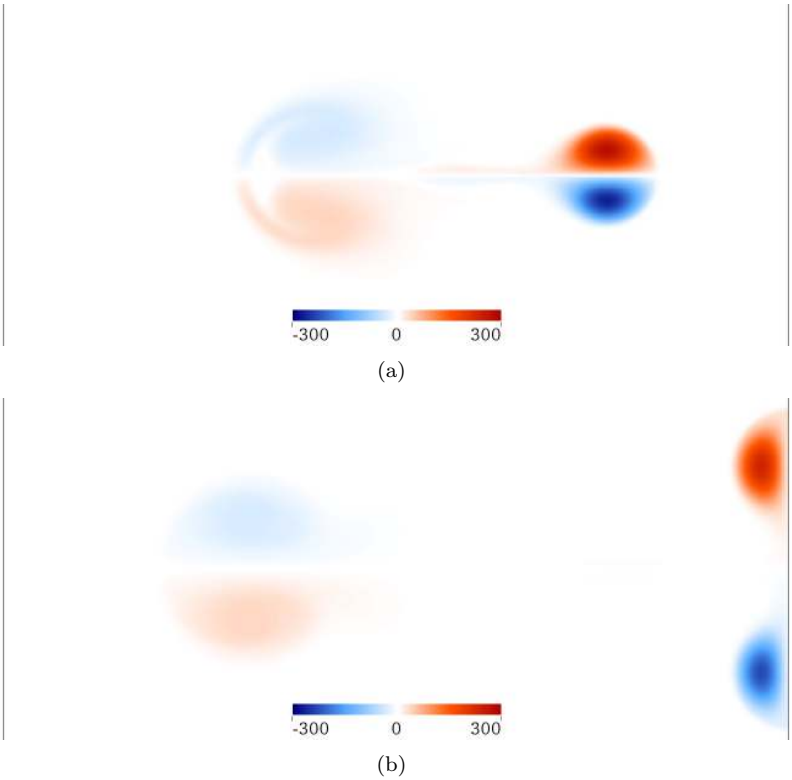


Figure 3.6: *Vorticity snapshots for the case at $Re = 1000$ and with a no-through flow condition at the cavity walls. The fields correspond to the times (a) $t = 0.2$ before the collision with the wall and (b) $t = 0.4$ after the collision, i.e. when the errors are measured for the time convergence study.*

According to the observations that will be made in Chapter 8, the periodic redistribution of the particles modifies the underlying equation by adding some spurious advection and diffusion terms. As a consequence, all the simulations performed for the time convergence study must have the same absolute redistribution period $n^r \Delta t$, in order to ensure the convergence to the solution of the same underlying discretized equation.

The reference solution is computed using $\Delta t_{\text{ref}} = 1 \cdot 10^{-5}$ and $n^r = 40$ for RK2 ($\Delta t_{\text{ref}} = 1 \cdot 10^{-5}$ and $n^r = 20$ for RK3). The results are shown in Fig. 3.7 and Table 3.1. They confirm the second order accuracy in time for the RK2 algorithm and the third order accuracy for the RK3 algorithm. Moreover, the error is seen to be sensibly higher at $t = 0.4$ than at $t = 0.2$, as expected.

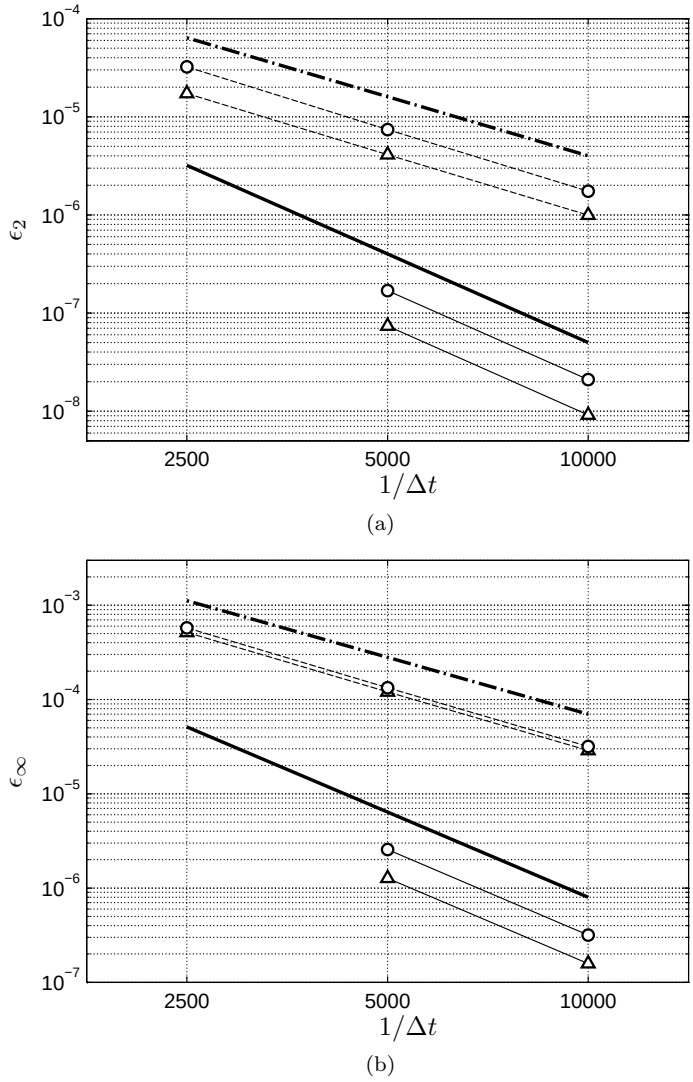


Figure 3.7: *Time convergence study on a* 512×512 *grid for the case at* $Re = 1000$ *with a no-through flow condition at the wall: (a)* L_2*-error norm and (b)* L_∞*-error norm. The error norms are represented by dashed lines for the RK2 computations and thin solid lines for the RK3 computations (with "*Δ*"-signs for the error measured at* $t = 0.2$ *and "*$\circ$*"-signs for that measured at* $t = 0.4$*); the thick solid line shows a third order slope and the thick dash-dotted line shows a second order slope.*

	RK2		RK3	
	$t = 0.2$	$t = 0.4$	$t = 0.2$	$t = 0.4$
ϵ_2	2.05	2.09	3.01	3.01
ϵ_∞	2.06	2.07	3.01	3.01

Table 3.1: *Observed order of convergence for the time convergence study on a* 512×512 *grid for the case at* $Re = 1000$ *and with a no-through flow condition at the wall.*

3.3 Enforcing a no-slip condition at the wall

Let us now consider the case where the boundary condition on $\partial\Omega$ is $\mathbf{u} = 0$:

$$\begin{aligned} \frac{D\omega}{Dt} &= \nu\nabla^2\omega \qquad \text{with} \quad q_\omega \triangleq -\nu\frac{\partial\omega}{\partial n} \neq 0 \quad \text{on } \partial\Omega \\ \mathbf{u} &= \nabla \times (\Psi\hat{\mathbf{e}}_z) \\ \nabla^2\Psi &= -\omega \qquad \text{with} \quad \mathbf{u} = 0 \quad \text{on } \partial\Omega\,, \end{aligned} \tag{3.4}$$

i.e., next to the no-through flow condition $\mathbf{u} \cdot \mathbf{n} = 0$, the no-slip condition $\mathbf{u} \cdot \mathbf{s} = 0$ is also prescribed, compared to Eq. (3.2). First, notice that solving Eq. (3.3) with $\mathbf{u} \cdot \mathbf{n} = 0$ and $\mathbf{u} \cdot \mathbf{s} = 0$ is over-determined according to [76], as it would amount to solve a Poisson equation with both a Dirichlet condition ($\Psi = 0$ resulting from $\mathbf{u} \cdot \mathbf{n} = 0$) and a Neumann condition ($\partial\Psi/\partial n = 0$ resulting from $\mathbf{u} \cdot \mathbf{s} = 0$). Moreover, the translation of the no-slip condition to the vorticity field is not trivial when considering the time evolution equation for the vorticity.

As a consequence (and conforming to what was originally done in [33], [72] and [102]), we merely build the approach upon the solver from the previous Section 3.2 - we refer to the latter as to the *no-through flow solver* - and, in the present section, we describe the required corrections in order to provide the correct boundary condition at the wall, i.e the no-slip condition.

In the light of the previous discussion, the tangential slip velocity $\mathbf{u} \cdot \mathbf{s}$ allowed by the no-through flow solver is considered spurious here, and it thus has to be canceled. As already mentioned in Section 3.2, the vorticity flux emanating from the wall provides a mechanism to do so. The vorticity flux and the no-slip condition are indeed intrinsically linked, as for an incompressible viscous flow, the only source of vorticity is the flux resulting from the presence of solid walls.

Computing the cross product between $\mathbf{n}$ and the Navier-Stokes equations in the velocity-pressure formulation leads to an exact expression of the vorticity flux at the wall, in 3-D,

$$\nu \, \frac{\partial \boldsymbol{\omega}}{\partial n} = \mathbf{n} \times \frac{D\mathbf{u}}{Dt} + \frac{1}{\rho} \, (\mathbf{n} \times \nabla p) + \nu \, (\mathbf{n} \times \nabla) \times \boldsymbol{\omega} \, , \tag{3.5}$$

according to Wu et al. [133] and to the pioneering work of Lighthill [79]. Note that the following vector identity (valid on the wall surface) has been used

$$\frac{\partial \boldsymbol{\omega}}{\partial n} = (\mathbf{n} \times \nabla) \times \boldsymbol{\omega} - \mathbf{n} \times (\nabla \times \boldsymbol{\omega}) \, .$$

The last term from Eq. (3.5) is non-zero only for 3-D flows and can be rewritten as [133]

$$\begin{aligned} (\mathbf{n} \times \nabla) \times \boldsymbol{\omega} &= \mathbf{n} \cdot \left[(\nabla \boldsymbol{\omega})^{\mathrm{T}} - (\nabla \cdot \boldsymbol{\omega}) \mathbf{I} \, \right] \\ &= -\mathbf{n} (\nabla_\pi \cdot \boldsymbol{\omega}) + \nabla_\pi \, \omega_n + \boldsymbol{\omega}_\pi \cdot \mathbf{K} \\ &= \mathbf{n} \frac{\partial \omega_n}{\partial n} + \nabla_\pi \, \omega_n + \boldsymbol{\omega}_\pi \cdot \mathbf{K} \, , \end{aligned}$$

where $\boldsymbol{\omega}_\pi$ represents the projection of $\boldsymbol{\omega}$ into the plane π that is tangent to the wall, ω_n is the normal component of $\boldsymbol{\omega}$ (hence $\boldsymbol{\omega} = \omega_n \mathbf{n} + \boldsymbol{\omega}_\pi$), $\nabla_\pi \triangleq \nabla - \mathbf{n}(\mathbf{n} \cdot \nabla)$ is the gradient operator in the plane π, $\mathbf{I}$ is the unit tensor and $\mathbf{K} \triangleq -\nabla_\pi \mathbf{n}$ is the surface curvature tensor. As a consequence, we have

$$\nu \, \frac{\partial \boldsymbol{\omega}}{\partial n} = \mathbf{n} \times \frac{D\mathbf{u}}{Dt} + \frac{1}{\rho} \, (\mathbf{n} \times \nabla p) + \nu \, \left(\mathbf{n} \frac{\partial \omega_n}{\partial n} + \nabla_\pi \, \omega_n + \boldsymbol{\omega}_\pi \cdot \mathbf{K} \right) \, , \tag{3.6}$$

which shows that the vorticity flux is a complex physical phenomenon induced by a tangential pressure gradient, a tangential acceleration and also by 3-D interactions between the vortex lines and the surface curvature.

Even for the present simplified case (2-D flow in the presence of non-moving walls), the evaluation of the vorticity flux still requires the knowledge of the pressure gradient, which is not a priori available in a vortex method.

The following discussion about the way to circumvent this apparent issue is inspired from [34] and starts with a brief summary about the work that has been done in past decades in order to compute this flux with the aim of accounting for solid walls in vortex methods. Based on that, the nature of the approximated wall vorticity flux is studied, as well as the error introduced, and various ways to integrate the underlying no-slip enforcement procedure inside a given time integration scheme are presented.

Let us start with an *admissible* vorticity field ω^n (i.e., and according to [34], a vorticity field that naturally provides a velocity field satisfying a no-slip condition at the wall when solving Eq. (3.3) using the no-through flow condition $\Psi = \overline{\Psi} = 0$ on $\partial\Omega$). If we advance the solution from t^n to $t^n + \Delta t$ using a no-through flow condition at the wall (like in Section 3.2), the resulting vorticity field ω^* is no longer admissible. A singular vortex sheet $\Delta\gamma^*$ of infinitesimal thickness can be associated to the resulting slip velocity on the wall $\partial\Omega$ at t^{n+1}. Its value is given by the jump of tangential velocity $\mathbf{u}^* \cdot \mathbf{s}$ (see Fig. 3.8) and can be deduced from a further solution of Eq. (3.3) with ω^* as right-hand side. It will be shown in Chapter 4 that

$$\Delta\gamma^* = -\frac{\partial \Psi^*}{\partial n} = -\mathbf{u}^* \cdot \mathbf{s} \ ,$$

where Ψ^* and $\mathbf{u}^*$ are taken on the flow side of the sheet (thus "above the sheet", i.e. the "+" side in Fig. 3.8) and where we also assume that $\mathbf{u}^* \cdot \mathbf{s} = 0$ on the body side (thus "below" the sheet, i.e. the "−" side in Fig. 3.8).

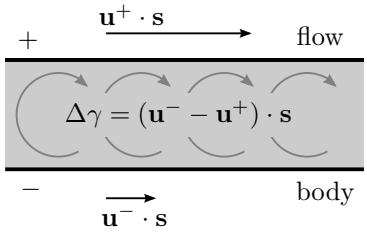


Figure 3.8: *Sketch of a vortex sheet $\Delta\gamma$ corresponding to the jump of tangential velocity $\mathbf{u} \cdot \mathbf{s}$.*

Using Eq. (3.3) and the divergence theorem, the total circulation of this vortex sheet is then implicitly

$$\oint_{\partial\Omega} \Delta\gamma^* \, d\mathbf{x} = \int_{\Omega} \omega^* \, d\mathbf{x} \ . \tag{3.7}$$

Eq. (3.7) shows that enforcing a no-slip condition in a cavity with non-moving walls is well-posed only if $\int_\Omega \omega \ d\mathbf{x} = 0$, as the slip velocity should tend to 0 when $\Delta t \to 0$ (when the flow is not subjected to an impulsive acceleration). As a side note, the vortex sheet $\Delta\gamma^*$ may be computed numerically based on the grid field $\mathbf{u}^*_{ij}$. Thanks to the symmetry property of the tangential velocity, we have, for example on the bottom wall $y = y_b$,

$$\Delta\gamma^*_i = -\frac{9}{8} \, u^*_{i,0} + \frac{1}{8} \, u^*_{i,1} + \mathcal{O}(h^4) \ ,$$

with $u^*_{i,j} \triangleq u^*(x_i, y_b + (0.5 + j)h)$.

According to Lighthill's model [79], the vortex sheet is actually part of the flow, and the following vorticity field is now "admissible"

$$\omega^*(\mathbf{x}) + \oint_{\partial\Omega} \Delta\gamma^*(\mathbf{x}')\delta(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}\,,$$

as the no-slip condition is satisfied below the sheet. The velocity field induced by the latter extended vorticity field still presents a non-physical jump in the tangential velocity component, by construction, and some regularization is therefore required (the velocity must be smooth, as it must converge to the solution of the Navier-Stokes equations for incompressible flows).

Despite being "non-physical", the vortex sheet gives a measure of the vorticity that must enter the flow. The bigger the slip velocity, the more vorticity must enter the flow and hence the higher the associated vorticity flux has to be in order to compensate the slip velocity.

Following Kinney et al. [70], Koumoutsakos et al. [72] further illustrates the link between the vorticity flux and the vortex sheet by considering the 2-D flow past a solid body described by its boundary $\partial\Omega_b$ and subjected to a solid rotation rate $W_b(t)$. According to [72] and [102], the vortex sheet that appears at the body boundary when using the no-through flow solver to advance the solution from t^n to $t^n+\Delta t$, may be used to provide the flow with the increment of circulation required by Kelvin's theorem when $W_b(t)$ is not constant. More details about this derivation may be found in Appendix D. The following relation between the vorticity flux and the vortex sheet is then obtained

$$-\oint_{\partial\Omega_b}\left(\nu\int_{t^n}^{t^n+\Delta t}\frac{\partial\omega}{\partial n}\,dt'\right)\,d\mathbf{x} = \oint_{\partial\Omega_b}\Delta\gamma^*\,d\mathbf{x}\,.$$

However, this link is global and it prescribes the total circulation the vortex sheet must have in order to be compatible with Kelvin's theorem. A further assumption, in agreement with the above observations, consists in stating that this link remains valid locally

$$\Delta\gamma^* \simeq -\nu\int_{t^n}^{t^n+\Delta t}\frac{\partial\omega}{\partial n}\,dt \quad \text{or} \quad \frac{\partial(\Delta\gamma^*)}{\partial t} \simeq -\nu\frac{\partial\omega}{\partial n}\,, \tag{3.8}$$

if the vortex sheet is furthermore given a time continuous meaning, as shown in [33] and [34].

For the 3-D case with surface curvature, a Robin type condition on the vorticity has to be used instead of a Neumann condition, as was put forward in [35] and studied in [106]. The fact that one must account for the curvature is noticeable in Eq. (3.6) and some applications using this Robin boundary condition are provided in [105].

The validity of Eq. (3.8) has been studied in [15] for the 3-D Stokes equations in a semi-infinite domain bounded by a planar surface (applying a Neumann condition is here valid since there is no curvature). The authors show that the vorticity flux and the vortex sheet are actually linked by an integral equation when comparing the solutions obtained from the diffusion of a vorticity field, on the one hand with a prescribed wall vorticity flux and, on the other hand, with a zero vorticity flux (as does the no-through flow solver). Some error estimates are then obtained when comparing the actual vorticity flux with its approximation based on the resulting vortex sheet Eq. (3.8)

$$\frac{1}{U}\left\|\Delta\gamma^* + \nu\int_{t^n}^{t^n+\Delta t}\frac{\partial\omega}{\partial n}\,dt\right\|_\infty = \mathcal{O}\left(\left(\frac{\nu\Delta t}{L^2}\right)^p\right) = \mathcal{O}\left(\left(\frac{\Delta\tilde{t}}{Re}\right)^p\right)\ ,$$

with $Re \triangleq UL/\nu$, $\Delta\tilde{t} \triangleq U\Delta t/L$, L a reference length and U a reference velocity. The order of the error is $p = \frac{3}{2}$ when the generated vortex sheet scales with Δt and $p = \frac{1}{2}$ when $\Delta\gamma^*/U = \mathcal{O}(1)$ (typically for impulsively accelerated flows).

Nevertheless, the approximated flux is in agreement with the underlying physical mechanism of vorticity production at the wall, as the model aims at enforcing a no-slip condition. Yet, the exact order of convergence of the solution, computed using this flux, to the solution of the Navier-Stokes equations remains an open question, as the contribution of the advection term to the error on the vorticity flux has not been quantified yet, to the author's knowledge.

The value of the vorticity flux may only be evaluated at the end of the time step computed using the no-through flow solver, i.e. when a measure of the committed slip error is available. The classical way to account for it consists in using a fractional step algorithm, as was originally proposed by [33], [72] and [73]. The computation is split into a no-through flow step (with zero vorticity flux) followed by a no-slip enforcement step imposing the value of the flux. In its basic form, the splitting is performed as follows in 2-D:

Step 1: Based on an admissible vorticity field ω^n at time t^n, solve Eq. (3.4) in order to obtain the vorticity field ω^* at time $t^n + \Delta t$ satisfying a no-through

flow condition at the wall (see Fig. 3.9(a)), using for example the solver from Section 3.2.

Step 2: Evaluate the generated spurious vortex sheet by solving

$$\begin{aligned} \nabla^2 \Psi^* &= -\omega^* \quad \text{with} \quad \Psi^* = 0 \quad \text{on } \partial\Omega \\ \Delta\gamma^* &= -\frac{\partial \Psi^*}{\partial n} \qquad\qquad\quad \text{on } \partial\Omega \ , \end{aligned}$$

and then solve the following diffusion equation, in the same time interval $[t^n, t^n + \Delta t]$

$$\begin{aligned} \frac{\partial \omega_w}{\partial t} &= \nu \nabla^2 \omega_w \quad \text{with} \quad -\nu \frac{\partial \omega_w}{\partial n} = \frac{\Delta\gamma^*}{\Delta t} \quad \text{on } \partial\Omega \\ \omega_w(\mathbf{x}, t^n) &= 0 \ , \end{aligned} \tag{3.9}$$

where the flux from Eq. (3.8) is assumed constant over the time step with no initial slip at t^n, by definition. This actually corresponds to using the time average of the flux over the time step. Simple explicit formulas have been developed [72] and then further improved in [102] (see Appendix E.8 for more details), in order to solve the diffusion Eq. (3.9) in a semi-infinite domain for a panel with uniform flux distribution (and furthermore constant in time).

The solution at $t^n + \Delta t$ is then $\omega^{n+1} = \omega^* + \omega_w$ and should satisfy the no-slip condition at the wall. Yet, in practice, the field ω^{n+1} is not exactly admissible (see Fig. 3.9(b)). Note that the combined field consisting of ω^* and the vortex sheet $\Delta\gamma^*$ is admissible, by construction. However, the diffusion of the vortex sheet $\Delta\gamma^*$, despite being conservative, slightly rearranges the vorticity in the flow and this reintroduces a small erroneous slip velocity at the wall. Section 3.3.2 deals with this issue by showing that an iteration may be required for the no-slip enforcement. The same observation holds for the initial vorticity field ω^n, which is never strictly admissible at the beginning of the time step computation, due to the previous observation. The correction procedure therefore also attempts at correcting the slip remaining from the previous time step.

We refer to the above splitting as to the "bulk-wall decomposition" (BWD) approach, since the first step handles the evolution of the bulk field using zero flux at the wall and the second step accounts for the contribution of the wall flux. The time integration algorithm using an RK2 scheme for the "no-through flow step" of the above BWD (i.e. Step 1), is called DRK2-END-PW-NS and is

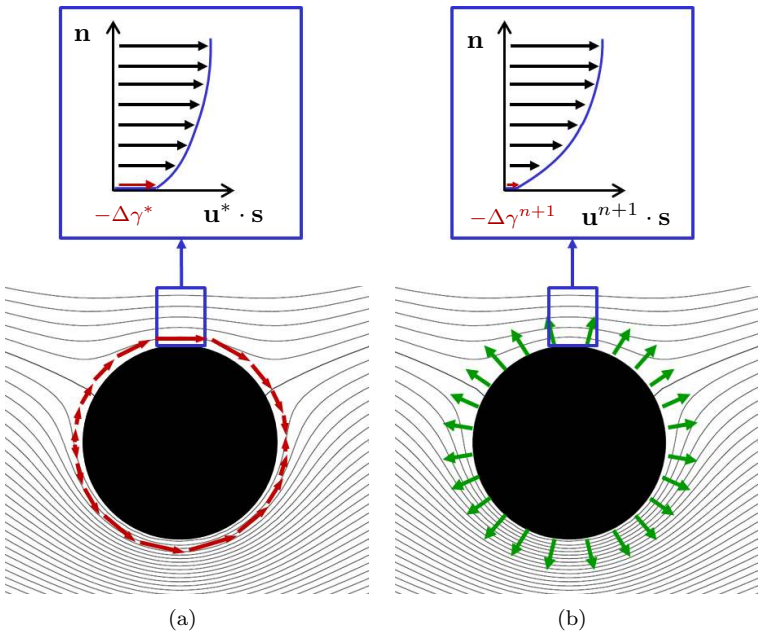


Figure 3.9: *Enforcement of the no-slip condition at the wall: (a) computation of a field* $\mathbf{u}^*$ *satisfying a no-through flow condition, (b) diffusion of the resulting vortex sheet* $\Delta\gamma^*$ *into the flow, which results in a field* $\mathbf{u}^{n+1}$ *with a smaller slip error* $\Delta\gamma^{n+1}$.

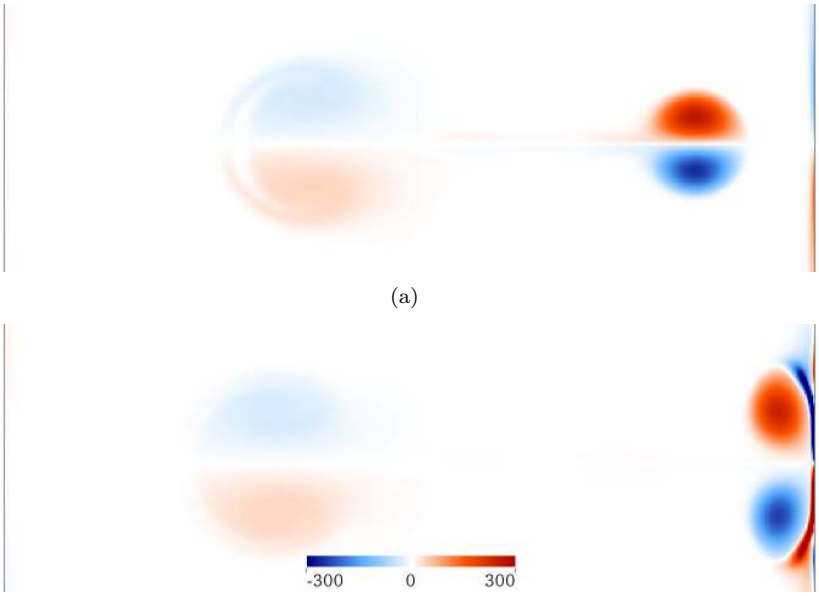
detailed in Appendix E (the notation DRK2-END-PW-NS means: Decomposed RK2 scheme computing the wall contribution at the END of the time step using the PW [102] formulas for the No-Slip enforcement; this will become clear in the following sections).

The present BWD approach is first order in time, according to [34], and the following time convergence study of the DRK2-END-PW-NS algorithm confirms this predicted accuracy. This is expected, since the wall contribution is only computed at the end of the RK2 time step, and not at every sub step. Results are shown in Table 3.2. The same numerical setup as in Section 3.2.3 is used here (the reference solution is also computed using DRK2-END-PW-NS with $\Delta t_{\text{ref}} = 10^{-5}$). The errors are measured at $t = 0.25$ (before the collision) and $t = 0.348$ (at the enstrophy maximum), see Fig. 3.10.

The main challenge therefore consists in adapting and incorporating the BWD approach at the sub step level of a given time integration scheme (e.g. inside the RK2 solver from Section 3.2 or inside the RK3 solver from Appendix E), so as to increase the order of convergence.

t	Δt	ϵ_2	order	ϵ_∞	order
$t = 0.25$	2×10^{-4}	2.6595×10^{-5}		1.1701×10^{-3}	
	1×10^{-4}	1.2407×10^{-5}	1.10	5.5341×10^{-4}	1.08
$t = 0.348$	2×10^{-4}	3.0277×10^{-4}		2.7836×10^{-2}	
	1×10^{-4}	1.4365×10^{-4}	1.08	1.3711×10^{-2}	1.02

Table 3.2: *Time convergence study on a* 512×512 *grid for the algorithm DRK2-END-PW-NS and for the case at* $Re = 1000$.



(a)

(b)

Figure 3.10: *Vorticity snapshots for the case at* $Re = 1000$ *with a no-slip condition at the cavity walls (obtained using DRK2-END-PW-NS). The fields correspond to the times (a)* $t = 0.25$ *before the collision with the wall and (b)* $t = 0.348$ *at the enstrophy maximum, i.e. when the errors are measured for the time convergence study.*

At this point, two different aspects of the problem should be distinguished:

- the splitting of the equations,
- the approximation of the vorticity flux.

These aspects can be studied separately, and increasing the order of convergence of the whole no-slip enforcing approach implies improving both aspects. The splitting is studied in Section 3.3.1 and the computation of the flux in Section 3.3.2.

3.3.1 Study of the splitting effect for a prescribed flux

The effect of the splitting can be examined by considering a (non-physical) problem, where the wall flux q_ω is prescribed. The resulting flow still satisfies a no-through flow condition at the wall, but the vorticity production is artificial, as it is not related to the no-slip condition here, i.e. the wall is no longer "inviscid".

First, the solution to this problem is computed without splitting of the equations, as the flux is known at any time, and the convergence is assessed both for the RK2 and the RK3 time integration schemes. Secondly, the effect of the splitting, that is used for the aforementioned no-slip enforcement procedure, is studied.

Time convergence study for the approach without splitting

The only computational step that is modified in the present case compared to the RK2 no-through flow solver (or equivalently to the RK3 solver from Appendix E) is the computation of $\nabla^2\omega$ that must be performed with a non-zero prescribed flux. For a second order discretization of $\nabla^2\omega$, the approach from Section 3.2.2, using an even symmetry of the vorticity, is no longer applicable and the computation of fourth order ghosts is required so as to satisfy the Neumann boundary condition on $\partial\Omega$ (sixth order ghosts are required for a fourth order discretization of $\nabla^2\omega$). The ghosts are obtained by extrapolation using the prescribed derivative at the boundary and a set of nodes adjacent to the wall.

The flux is given by

$$\begin{aligned} q_\omega(x,-\tfrac{L}{2},t) &= -Q_\omega(t)\cos\left(\frac{\pi x}{L}\right) \\ q_\omega(x,+\tfrac{L}{2},t) &= +Q_\omega(t)\cos\left(\frac{\pi x}{L}\right) \\ q_\omega(-\tfrac{L}{2},y,t) &= -Q_\omega(t)\sin\left(\frac{2\pi y}{L}\right) \\ q_\omega(+\tfrac{L}{2},y,t) &= -Q_\omega(t)\sin\left(\frac{2\pi y}{L}\right) \\ Q_\omega(t) &= 100\sin\left(\frac{2\pi t}{T_q}\right) \\ T_q &= 0.1\,, \end{aligned} \tag{3.10}$$

where $L = 2$ is the cavity side length. The order of magnitude of the prescribed flux is comparable to the observed flux associated with the no-slip condition

at $Re = 1000$. The numerical setup used for the time convergence study is identical to that performed in Section 3.2.3 (the vorticity field for this test case is shown in Fig. 3.11 at $t = 0.2$ and $t = 0.4$).

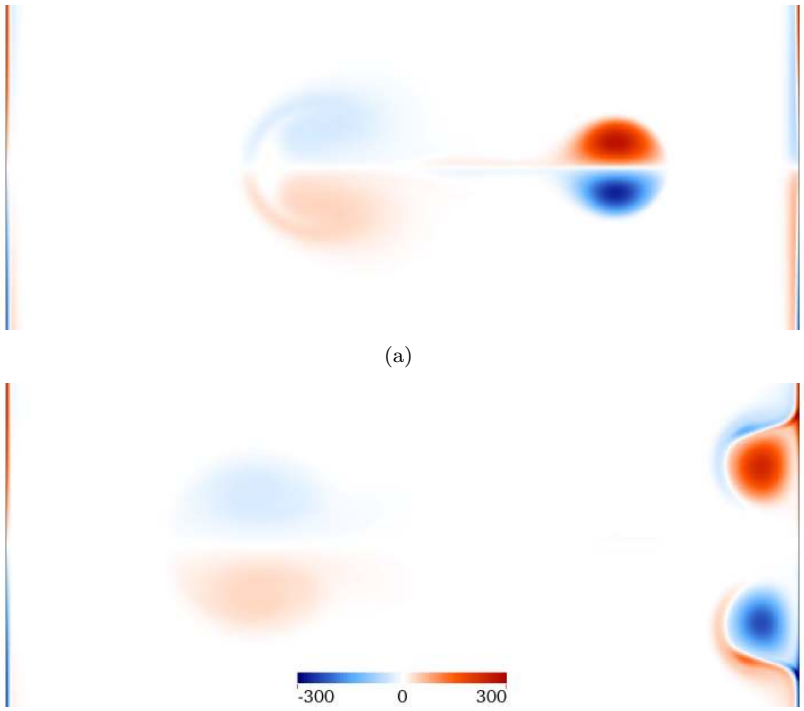


Figure 3.11: *Vorticity snapshots for the case at* $Re = 1000$ *with a no-through flow condition and a prescribed non-zero flux at the cavity walls. The fields correspond to the times (a)* $t = 0.2$ *before the collision with the wall and (b)* $t = 0.4$ *after the collision (the errors are measured at those times for the convergence study).*

The solution computed using the RK2 scheme without splitting serves as a reference for the remainder of this chapter, as any alternative time integration scheme (with splitting) should converge to the same solution. Results are given in Fig. 3.12 and Table 3.3. They show that the rate of convergence observed in Section 3.2.3 is not affected by the addition of a non-zero vorticity flux. As a side note, the particle-mesh interpolation procedure still assumes an even symmetry for the vorticity field, which is no longer true here. Interestingly, this simplification does not affect the convergence.

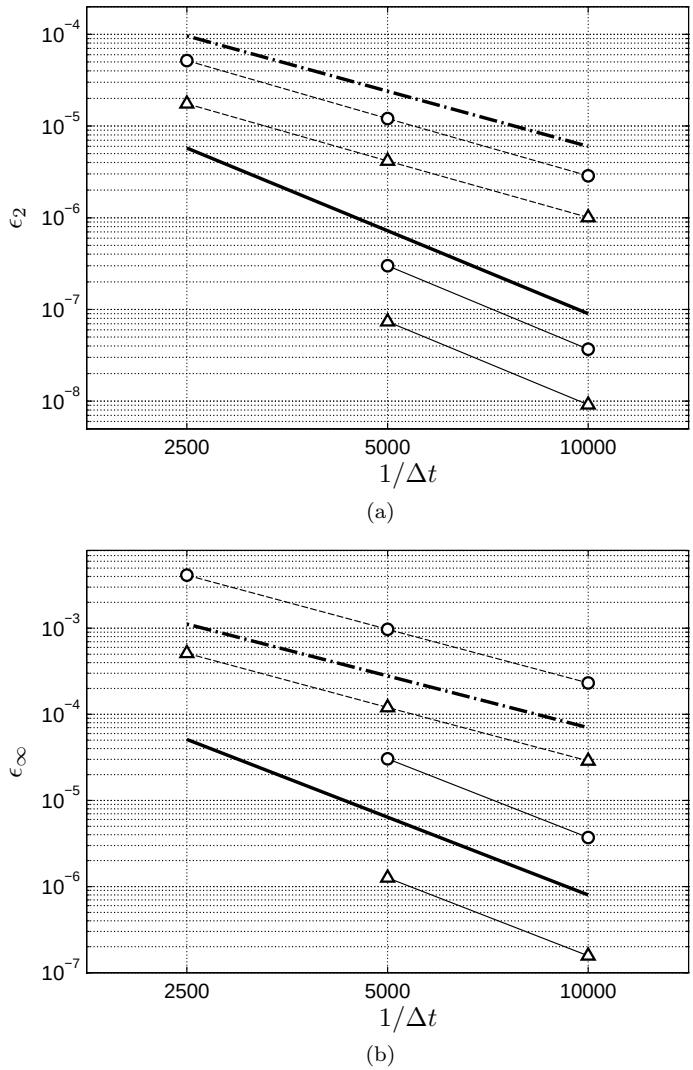


Figure 3.12: *Time convergence study on a* 512×512 *grid for the case at* $Re = 1000$ *with a no-through flow condition and a prescribed vorticity flux at the wall: (a)* L_2*-error norm and (b)* L_∞*-error norm. The error norms are represented by dashed lines for the RK2 computations and thin solid lines for the RK3 computations (with "*Δ*"-signs for the error measured at* $t = 0.2$ *and "*$\circ$*"-signs at* $t = 0.4$*); the thick solid line shows a third order slope and the thick dash-dotted line shows a second order slope.*

	RK2		RK3	
	$t = 0.2$	$t = 0.4$	$t = 0.2$	$t = 0.4$
ϵ_2	2.05	2.07	3.01	3.02
ϵ_∞	2.06	2.07	3.01	3.04

Table 3.3: *Observed order of convergence for the time convergence study on a* 512×512 *grid for the case at* $Re = 1000$ *with a no-through flow condition and a prescribed vorticity flux at the wall.*

Accuracy of the no-slip enforcing procedure

First, one should observe that a noticeable difference exists between the above no-slip enforcing procedure based on the bulk-wall decomposition (BWD) and the classical splitting algorithm. The latter was introduced in the context of vortex methods by Chorin [22] in order to account for the diffusion by using a random-walk formulation (the accuracy of the splitting was later studied in [7]). It consists in splitting the time step computation of the Navier-Stokes equations into a first step solving the Euler equations and a second step consequently solving the Stokes equations based on the solution obtained in the first step.

The BWD does not strictly follow this splitting scheme. The difference becomes clear when analyzing the solution of the following linear convection-diffusion equation (the velocity $\mathbf{c}(\mathbf{x}, t)$ is assumed independent of ω here)

$$\frac{\partial \omega}{\partial t} + \mathbf{c} \cdot \nabla \omega = \nu \nabla^2 \omega \; . \tag{3.11}$$

Similarly to [34], the convection operator is defined as $H \triangleq -\mathbf{c} \cdot \nabla$ and the diffusion operator as $D \triangleq \nu \nabla^2$; the matrices $\mathbf{H}$ and $\mathbf{D}$ correspond to the numerical discretization of these operators. For the sake of clarity, we consider here a Eulerian approach that computes the evolution of $\mathbf{W}$, the vector of unknowns ω, and we may rewrite the linear PDE as an ODE

$$\frac{d\mathbf{W}}{dt} = \mathbf{H}\,\mathbf{W} + \mathbf{D}\,\mathbf{W} \; .$$

The exact solution $\mathbf{W}(t) = e^{(\mathbf{H}+\mathbf{D})t}\,\mathbf{W}_0$ (with $\mathbf{W}_0$ the initial condition) can also be rewritten as

$$\mathbf{W}^{n+1} = e^{(\mathbf{H}+\mathbf{D})\Delta t}\,\mathbf{W}^n \; ,$$

when solving the problem over one time step Δt between t^n and t^{n+1}.

Splitting the computational step into an inviscid step (Euler equations) and a viscous step (Stokes equations) in the sense of Chorin [22] amounts to first solve for $\tilde{\mathbf{W}} \triangleq e^{\mathbf{H}\Delta t}\ \mathbf{W}^n$ and, based on that, subsequently for $\mathbf{W}^{n+1} \triangleq e^{\mathbf{D}\Delta t}\ \tilde{\mathbf{W}} = e^{\mathbf{D}\Delta t}e^{\mathbf{H}\Delta t}\ \mathbf{W}^n$, which leads to an $\mathcal{O}(\Delta t^2)$ error at every time step, as $e^{\mathbf{D}\Delta t}e^{\mathbf{H}\Delta t} = e^{(\mathbf{H}+\mathbf{D})\Delta t} + \frac{\Delta t^2}{2}(\mathbf{DH}-\mathbf{HD}) + \mathcal{O}(\Delta t^3)$ ($\mathbf{H}$ and $\mathbf{D}$ do not commute, unless $\mathbf{c}$ is uniform), and hence the approach is first order overall.

Now, in the case of the BWD approach, consider again the simplified case where the flux q_ω at the wall does not depend on the solution but is prescribed. Thanks to the linearity, the solution of Eq. (3.11) may be decomposed as $\omega \triangleq \omega^* + \omega_w$, where ω^* and ω_w are respectively solutions of

$$\begin{array}{rcl} \dfrac{\partial \omega^*}{\partial t} + \mathbf{c}\cdot\nabla\omega^* &=& \nu\nabla^2\omega^* \\ \omega^*(t=t^n) &=& \omega^n \\ -\nu\dfrac{\partial \omega^*}{\partial n} &=& 0 \end{array} \quad \text{and} \quad \begin{array}{rcl} \dfrac{\partial \omega_w}{\partial t} + \mathbf{c}\cdot\nabla\omega_w &=& \nu\nabla^2\omega_w \\ \omega_w(t=t^n) &=& 0 \\ -\nu\dfrac{\partial \omega_w}{\partial n} &=& q_\omega(t) \end{array} \quad . \tag{3.12}$$

This decomposition requires to add the flux vector $\mathbf{Q}_\omega$ to the vector of unknowns (note that it is still prescribed, though, as explained hereafter) and to extend the definition of $\mathbf{H}$ and $\mathbf{D}$ so as to also account for a wall flux

$$\mathbf{V} = \begin{bmatrix} \mathbf{W} \\ \mathbf{Q}_\omega \end{bmatrix} \triangleq \mathbf{V}^* + \mathbf{V}_w \quad \text{and} \quad \mathbf{H}+\mathbf{D} \triangleq \begin{bmatrix} \mathbf{A}_{WW} & \mathbf{A}_{WQ} \\ \mathbf{A}_{QW} & \mathbf{A}_{QQ} \end{bmatrix} \quad ,$$

where $\mathbf{V}^*$ corresponds to ω^*, $\mathbf{V}_w$ to ω_w, $\mathbf{A}_{QQ}(t)$ is such that $e^{\mathbf{A}_{QQ}t}\mathbf{Q}_\omega(t=0)$ yields the prescribed flux $\mathbf{Q}_\omega(t)$ and $\mathbf{A}_{QW} = 0$, since the flux does not depend on the vorticity in the present analysis (for the no-slip enforcement we would have $\mathbf{A}_{QW} \neq 0$). The associated ODE's read

$$\begin{array}{rcl} \dfrac{d\mathbf{V}^*}{dt} &=& \mathbf{H}\ \mathbf{V}^* + \mathbf{D}\ \mathbf{V}^* \\ \mathbf{V}^*(t^n) &=& \widehat{\mathbf{V}}^n = \begin{bmatrix} \mathbf{W}^n \\ 0 \end{bmatrix} \end{array} \quad \text{and} \quad \begin{array}{rcl} \dfrac{d\mathbf{V}_w}{dt} &=& \mathbf{H}\ \mathbf{V}_w + \mathbf{D}\ \mathbf{V}_w \\ \mathbf{V}_w(t^n) &=& \breve{\mathbf{V}}^n = \begin{bmatrix} 0 \\ \mathbf{Q}_\omega^n \end{bmatrix} \end{array} \quad ,$$

where the operators $\widehat{\ }$ and $\breve{\ }$ are defined as follows

$$\widehat{\mathbf{V}} \triangleq \begin{bmatrix} \mathbf{W} \\ 0 \end{bmatrix} \quad \text{and} \quad \breve{\mathbf{V}} \triangleq \begin{bmatrix} 0 \\ \mathbf{Q}_\omega \end{bmatrix} \quad , \quad \text{hence } \mathbf{V} = \widehat{\mathbf{V}} + \breve{\mathbf{V}} \ .$$

The solution $\mathbf{V}^{n+1}$ is then obtained as the sum of $\mathbf{V}^{n+1,*}$ and $\mathbf{V}_w^{n+1}$

$$\mathbf{V}^{n+1} = \mathbf{V}^{n+1,*} + \mathbf{V}_w^{n+1} = e^{(\mathbf{H}+\mathbf{D})\Delta t}\,\widehat{\mathbf{V}}^n + e^{(\mathbf{H}+\mathbf{D})\Delta t}\,\breve{\mathbf{V}}^n = e^{(\mathbf{H}+\mathbf{D})\Delta t}\,\mathbf{V}^n\ .$$

Note that the sequence of operations cannot be written as the composition $e^{\mathbf{D}_w\Delta t}e^{(\mathbf{H}+\mathbf{D}^*)\Delta t}$ of a convection-diffusion operator $\mathbf{H}+\mathbf{D}^*$ (with zero flux) and a near-wall diffusion operator $\mathbf{D}_w$. As a consequence, contrary to the classical splitting approach in the sense of Chorin [22], which relies on successively applied operators, the BWD is based on a field decomposition, by construction. Instead of being performed sequentially, the operations of this decomposition are "parallel in time" for a prescribed flux. However, this breaks down for a flux that depends on $\mathbf{V}^*$. We may now properly define the DRK2 algorithm as the "*Decomposed* RK2 scheme" corresponding to the time integration algorithm based on the bulk-wall decomposition and using the mid-point rule RK2 scheme.

A comparison between Eq. (3.9) and Eq. (3.12) reveals that the convection operator is missing in Eq. (3.9) for the original BWD approach. Similarly to Chorin's splitting, an error is therefore introduced at every time step. Indeed, the exact integration of these equations yields

$$\begin{aligned} e^{(\mathbf{H}+\mathbf{D})\Delta t}\,\widehat{\mathbf{V}}^n + e^{\mathbf{D}\Delta t}\,\breve{\mathbf{V}}^n &= e^{(\mathbf{H}+\mathbf{D})\Delta t}\,(\widehat{\mathbf{V}}^n + \breve{\mathbf{V}}^n) - \Delta t\mathbf{H}\,\breve{\mathbf{V}}^n \\ &\quad - \frac{\Delta t^2}{2}(\mathbf{H}^2 + \mathbf{HD} + \mathbf{DH})\,\breve{\mathbf{V}}^n + \mathcal{O}(\Delta t^3) \\ &= \mathbf{V}^{n+1} - \frac{\Delta t^2}{2}\mathbf{HD}\,\breve{\mathbf{V}}^n + \mathcal{O}(\Delta t^3) \\ &= \mathbf{V}^{n+1} + \mathcal{O}(\Delta t^2)\ , \end{aligned}$$

as $\mathbf{H}\breve{\mathbf{V}}^n = 0$, in practice. This is due to the fact that $\mathbf{H}$ is a hyperbolic operator that does not require any boundary condition along the wall, considering that the latter is a streamline and hence also a characteristic. From a different point of view, this can also be explained by the fact that, since the wall velocity is parallel to the boundary (at least in the "no-through flow" case), the derivative $\mathbf{c}\cdot\nabla$ is perpendicular to the normal derivative of the flux and must therefore be independent on it. As a consequence, we are left with an overall $\mathcal{O}(\Delta t)$ error. The time integration of this problem, in the context of VPM methods and using an RK2 scheme for the zero-flux contribution ω^*, is called the DRK2-END algorithm, i.e. Decomposed RK2 scheme computing the wall contribution

at the END of the time step, which means that it is evaluated only once (see Appendix E for a detailed description of this scheme).

Performing the decomposition at every sub step of the time integrator improves the formal order of convergence, according to the following analysis for the RK2 scheme:

Predictor :

$$\begin{aligned}
\mathbf{V}^{n+\frac{1}{2},*} &= \widehat{\mathbf{V}}^n + \frac{\Delta t}{2}(\mathbf{H}+\mathbf{D})\,\widehat{\mathbf{V}}^n \\
\mathbf{V}_w^{n+\frac{1}{2}} &= \breve{\mathbf{V}}^n + \frac{\Delta t}{2}\,\mathbf{D}\,\breve{\mathbf{V}}^n \\
\Rightarrow\quad \mathbf{V}^{n+\frac{1}{2}} &= \mathbf{V}^n + \frac{\Delta t}{2}(\mathbf{H}+\mathbf{D})\,\mathbf{V}^n - \frac{\Delta t}{2}\,\mathbf{H}\,\breve{\mathbf{V}}^n\,.
\end{aligned}$$

The computation of the next sub step requires to re-decompose the vector $\mathbf{V}^{n+\frac{1}{2}}$ into a "zero-flux" component $\widehat{\mathbf{V}}^{n+\frac{1}{2}}$ and a "zero-initial condition" component $\breve{\mathbf{V}}^{n+\frac{1}{2}}$.

Corrector :

$$\begin{aligned}
\mathbf{V}^{n+1,*} &= \widehat{\mathbf{V}}^n + \Delta t\,(\mathbf{H}+\mathbf{D})\,\widehat{\mathbf{V}}^{n+\frac{1}{2}} \\
\mathbf{V}_w^{n+1} &= \breve{\mathbf{V}}^n + \Delta t\,\mathbf{D}\,\breve{\mathbf{V}}^{n+\frac{1}{2}} \\
\Rightarrow\quad \mathbf{V}^{n+1} &= \mathbf{V}^n + \Delta t\,(\mathbf{H}+\mathbf{D})\,\mathbf{V}^{n+\frac{1}{2}} - \Delta t\,\mathbf{H}\,\breve{\mathbf{V}}^{n+\frac{1}{2}} \\
&= \Big[\,\mathbf{I} + \Delta t\,(\mathbf{H}+\mathbf{D}) + \frac{\Delta t^2}{2}(\mathbf{H}+\mathbf{D})^2\Big]\,\mathbf{V}^n \\
&\quad - \frac{\Delta t^2}{2}(\mathbf{H}+\mathbf{D})\mathbf{H}\,\breve{\mathbf{V}}^n - \Delta t\,\mathbf{H}\,\breve{\mathbf{V}}^{n+\frac{1}{2}} \\
&= e^{(\mathbf{H}+\mathbf{D})\Delta t}\,\mathbf{V}^n + \mathcal{O}(\Delta t^3)\,,
\end{aligned}$$

which shows that the overall error is now $\mathcal{O}(\Delta t^2)$, conforming to the accuracy of the RK2 scheme ($\mathbf{I}$ is the unit matrix). Yet, in the framework of VPM methods, there is still a lack of consistency, due to the missing convection term, and it does not appear in the above analysis valid for a linear problem (with an advection velocity thus independent of the vorticity) and using a Eulerian approach.

Consider to this end the DRK2-SUB scheme implementing this approach for a VPM method (it corresponds to a Decomposed RK2 scheme computing the wall contributions at the end of every SUB step of the time integrator, see Appendix E).

It performs the following operations for the predictor diffusion step:

$$
\begin{aligned}
\omega_{ij}^{n} &\xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n} \quad \text{with } -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n} = 0 \\
\left(\nabla^2\omega\right)_{ij}^{n} &\xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_{p}^{n} \\
\omega_p^{n+\frac{1}{2},*} &= \omega_p^n + \frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n \\
\omega_p^{n+\frac{1}{2},*} &\xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2},*} \\
\omega_{ij}^{n+\frac{1}{2}} &= \omega_{ij}^{n+\frac{1}{2},*} + \left(\omega_w\right)_{ij}^{n+\frac{1}{2}} ,
\end{aligned}
$$

where $\left(\omega_w\right)_{ij}^{n+\frac{1}{2}}$ is the contribution coming from the wall and computed on the grid using the flux at t^n.

One may observe that the bulk diffusion increment $\frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$ (evaluated with zero wall flux) is advected, since it is computed at $\mathbf{x}_p^n$ and then transported to $\mathbf{x}_p^{n+\frac{1}{2}}$, along with ω_p^n. This is not the case for the Eulerian approach described by the above error analysis, where only ω^n is advected, and not the bulk diffusion increment. However, a closer look to the DRK2-SUB scheme shows that the wall contribution is here *not* advected, as it is computed at $\mathbf{x}_p^{n+\frac{1}{2}}$ and then simply added to $\omega_{ij}^{n+\frac{1}{2},*}$.

Yet, the wall contribution is the complement of the bulk diffusion increment and it has the same status. The inconsistency arises from the different treatment that $\left(\omega_w\right)_{ij}^{n+\frac{1}{2}}$ is subject to, compared to $\frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$. Nevertheless, the missing advection term can be accounted for by mapping $\left(\omega_w\right)_{ij}^{n+\frac{1}{2}}$ onto the particle position $\mathbf{x}^n$ and by adding the result to the particle that already resides in $\mathbf{x}^{n+\frac{1}{2}}$, just the same way as for $\left(\nabla^2\omega\right)_{ij}^n$. The last step accounting for the wall flux contribution can thus be replaced by

$$
\begin{aligned}
\left(\omega_w\right)_{ij}^{n+\frac{1}{2}} &\xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\omega_w\right)_{p}^{n+\frac{1}{2}} \\
\omega_p^{n+\frac{1}{2}} &= \omega_p^{n+\frac{1}{2},*} + \left(\omega_w\right)_p^{n+\frac{1}{2}} \\
\omega_p^{n+\frac{1}{2}} &\xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2}} .
\end{aligned}
$$

The complete VPM algorithm, including the required modification in the corrector, is referred to as DRK2-CSUB (Decomposed RK2 scheme computing *Convected* wall contributions at the end of every SUB step) and is detailed in Appendix E.

The time accuracy of the algorithms DRK2-END-PW, DRK2-SUB-PW and DRK2-CSUB-PW is now assessed numerically for the flux prescribed by Eq. (3.10) and using the numerical setup from Section 3.3.1. All approaches use the explicit formulas from [102] (as indicated by the notation "PW"; see Appendix E.8), with a time-averaged flux computed as

$$\bar{q}_\omega \triangleq \frac{1}{\Delta t}\int_{t^n}^{t^n+\Delta t} q_\omega \, dt \; .$$

This is equivalent to the flux evaluation Eq. (3.9) from the original BWD-based no-slip enforcing procedure, as $\Delta\gamma^*/\Delta t$ measures the average of the flux over the time step, according to Eq. (3.8). Note that for the RK2 predictor used in DRK2-SUB-PW and DRK2-CSUB-PW, the flux average is performed over $\Delta t/2$, as the computation goes from t^n to $t^{n+\frac{1}{2}}$.

The results of the time convergence study are summarized in Table 3.4. The DRK2-CSUB-PW algorithm converges one order faster than DRK2-END-PW and DRK2-SUB-PW, when the error is measured with respect to a reference at $\Delta t_{\text{ref}} = 10^{-5}$ that is computed using the same algorithm. This shows that accounting for the missing advection term in DRK2-CSUB-PW clearly improves the consistency of the methodology. This stands in contrast with the "Eulerian" error analysis which predicts a second order accuracy also for DRK2-SUB-PW. As explained above, this analysis indeed ignores the advection of the diffusion increment. Nevertheless, the first order convergence predicted by the error analysis using the exact integration of the BWD approach (with one wall contribution computed at the end of the time step) is in agreement with the convergence rate observed for DRK2-END-PW and for DRK2-END-PW-NS, where the flux is linked to the no-slip condition (see Table 3.2).

On the contrary, all three approaches do not converge when the error is measured with respect to the RK2 reference solution, where the flux is imposed by using ghosts in the finite difference evaluation of $\nabla^2\omega$. This could be explained by the leading spatial error term that may differ from one approach to another.

Another possibility consists in using the classical splitting approach, as was done in Poncet [104]. In the first order variant, the particles are first advected and then interpolated onto the grid, where the diffusion is subsequently computed with zero flux. Finally, the wall contribution can be added to the vorticity field. The leading error term is $\frac{\Delta t^2}{2}(\mathbf{DH} - \mathbf{HD})\mathbf{V}^n$, as previously derived. Note that this error is introduced in the whole domain, since $\mathbf{V}^n$ in-

			RK2	
DRK2	ϵ_2	ϵ_∞	ϵ_2	ϵ_∞
END-PW	1.188×10^{-4}	6.931×10^{-3}	3.263×10^{-3}	1.543×10^{-1}
	1.08	1.02	-0.05	-0.02
SUB-PW	8.480×10^{-5}	1.280×10^{-2}	3.354×10^{-3}	1.590×10^{-1}
	1.05	1.07	-0.01	0.02
CSUB-PW	3.035×10^{-6}	2.246×10^{-4}	3.379×10^{-3}	1.571×10^{-1}
	2.18	2.14	0.00	0.00

Table 3.4: *Results for the time convergence study on a* 512×512 *grid for the case at* $Re = 1000$ *(*$t = 0.4$*) and with a no-through flow condition and a prescribed vorticity flux at the wall, using different time integration schemes (the indicated errors correspond to* $\Delta t = 10^{-4}$*); in the right part of the table, the error is measured with respect to the RK2 reference solution obtained with* $\Delta t_{ref} = 10^{-5}$*, whereas in the left part, the error is computed with respect to the own reference solution.*

cludes both the bulk and the wall contributions. This can be compared to the leading error term $\frac{\Delta t^2}{2}\mathbf{HD}\,\breve{\mathbf{V}}^n$ induced by the BWD approach. Interestingly, the latter introduces the first order error only in the vicinity of the wall, as $\breve{\mathbf{V}}^n$ solely contains the wall-flux.

The Strang formula [118] may be used in order to obtain a second order splitting

$$e^{\mathbf{D}\Delta t/2} e^{\mathbf{H}\Delta t} e^{\mathbf{D}\Delta t/2} = e^{(\mathbf{H}+\mathbf{D})\Delta t} + \mathcal{O}(\Delta t^3) \,,$$

while every sub step ($e^{\mathbf{D}\Delta t/2}$, $e^{\mathbf{H}\Delta t}$ and again $e^{\mathbf{D}\Delta t/2}$) must be integrated with a second order scheme (e.g. RK2). The latter methodology could become very costly when used for the no-slip enforcement, as a Poisson solution is required at each convection sub step, and also at each diffusion sub step in order to compute the resulting vortex sheet to be diffused.

More recently, another second order splitting algorithm, based on the present RK2 scheme (mid-point rule), was developed in Chatelin [19]. In the framework of Lagrangian vortex methods, studying the time accuracy of this algorithm requires to define the linearized convection operator $\boldsymbol{\mathcal{H}}$ and diffusion operator $\boldsymbol{\mathcal{D}}$ as follows

$$\boldsymbol{\mathcal{H}}\,\boldsymbol{\alpha} = \begin{bmatrix} 0 \\ \mathbf{u}_p \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mathcal{D}}\,\boldsymbol{\alpha} = \begin{bmatrix} \nu(\nabla^2\omega)_p \\ 0 \end{bmatrix} \,, \quad \text{with} \quad \boldsymbol{\alpha} \triangleq \begin{bmatrix} \omega_p \\ \mathbf{x}_p \end{bmatrix}$$

the vector of particle unknowns.

The sequence of operations is then

$$\begin{aligned}
\textbf{Predictor :} \qquad \widetilde{\boldsymbol{\alpha}}^{n+\frac{1}{2}} &= \boldsymbol{\alpha}^n + \frac{\Delta t}{2}\,\boldsymbol{\mathcal{H}}\,\boldsymbol{\alpha}^n \\
\boldsymbol{\alpha}^{n+\frac{1}{2}} &= \widetilde{\boldsymbol{\alpha}}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\,\boldsymbol{\mathcal{D}}\,\widetilde{\boldsymbol{\alpha}}^{n+\frac{1}{2}} \\
\textbf{Corrector :} \qquad \widetilde{\boldsymbol{\alpha}}^{n+1} &= \boldsymbol{\alpha}^n + \Delta t\,\boldsymbol{\mathcal{H}}\,\boldsymbol{\alpha}^{n+\frac{1}{2}} \\
\boldsymbol{\alpha}^{n+1} &= \widetilde{\boldsymbol{\alpha}}^{n+1} + \Delta t\,\boldsymbol{\mathcal{D}}\,\boldsymbol{\alpha}^{n+\frac{1}{2}}\ ,
\end{aligned}$$

which leads to an $\mathcal{O}(\Delta t^3)$ error at every time step, as shown in [19]. The associated VPM time integration scheme is called SRK2 (*Split* RK2) and is given in Appendix E, for reference.

Strictly speaking, the corrector sub step does not follow the rules of "classical splitting". For "classical splitting" approaches, the advection-diffusion is treated as a composed operation, i.e. the output of the advection operation is handled as an input for the diffusion. For the SRK2 scheme, the diffusion step inside the corrector computes the right hand side $\nu(\nabla^2\omega)_p$ at $t^{n+\frac{1}{2}}$ (i.e. when the particles are at a position $\mathbf{x}^{n+\frac{1}{2}}$), instead of computing it at t^{n+1}, at the particle position $\mathbf{x}^{n+1}$ that has just been computed during the advection step. Yet, as explained in [19], this small change is required to yield the second order accuracy.

Actually, a closer comparison between SRK2 and RK2 in Appendix E shows that the corrector step is identical in both cases, whereas the predictor performs the actual splitting of the operations. This integration scheme therefore still requires using "convected" sub step contributions in the corrector when the near-wall diffusion is accounted for in the decomposed scheme (DSRK2-CSUB in Appendix E).

The time convergence is now examined for the DRK2-CSUB-PW-NS algorithm (thus in the framework of the no-slip enforcing procedure), as it is similar to the DSRK2-CSUB scheme. The numerical setting for this study is the same as the one used for Table 3.2. Note that for the wall contribution inside the RK2 predictor, the flux is computed as $\Delta\gamma^{n+\frac{1}{2},*}/(\frac{1}{2}\Delta t)$, as the computation goes from t^n to $t^{n+\frac{1}{2}}$. Indeed, in the next Section 3.3.2, it will be shown that the evolution of the vortex sheet is linear with respect to the time period over which it was generated, i.e. $\frac{1}{2}\Delta t$ for the predictor. The corrector flux is classically computed as $\Delta\gamma^{n+1,*}/\Delta t$. Results are provided in Table 3.5 (the reference solution is computed using DRK2-CSUB-PW-NS with $\Delta t_{\text{ref}} = 10^{-5}$).

t	Δt	ϵ_2	order	ϵ_∞	order
$t = 0.25$	2×10^{-4}	2.7934×10^{-5}		1.6826×10^{-3}	
	1×10^{-4}	1.3190×10^{-5}	1.08	8.0097×10^{-4}	1.07
$t = 0.348$	2×10^{-4}	2.7644×10^{-4}		3.1857×10^{-2}	
	1×10^{-4}	1.3040×10^{-4}	1.08	1.5352×10^{-2}	1.05

Table 3.5: *Time convergence study on a* 512×512 *grid for the algorithm DRK2-CSUB-PW-NS and for the case at* $Re = 1000$.

The rate of convergence is still first order for both error norms L_2 and L_∞, with a similar error level as those of Table 3.2. This shows that, next to the improvement of the time algorithm by consistently accounting for the wall flux contribution inside the sub steps, a more accurate evaluation of the wall flux itself is needed. The next Section 3.3.2 deals with this subject.

3.3.2 Computation of the vorticity flux

As the vorticity flux required for the no-slip enforcement is based on the vortex sheet, the present section first examines the behavior of this sheet during the time step computation. Secondly, some suggestions are made in order to improve the flux evaluation.

For all the following computations, the case at $Re = 1000$ is again studied on a 512×512 grid, using the numerical setup from Table 3.2.

Convergence of the slip error

The DRK2-END-PW-NS algorithm is first order in time and we are now interested in the accuracy of the associated no-slip enforcement. A direct measure of the slip error is given by the vortex sheet $\Delta\gamma^n$ computed at the beginning of the time step and based on the vorticity field ω^n. Fig. 3.13 shows the vortex sheet on the right wall of the cavity at $t = 0.348$ (at the enstrophy maximum, during the collision of the dipole with the wall, see Fig. 3.10). It can be observed that the slip error scales with the time step Δt and the computed convergence rate for the L_∞-error norm $\epsilon_\infty \triangleq \max_y |\Delta\gamma^n|$ is exactly 1.00. As expected, the convergence of the slip error is hence also first order.

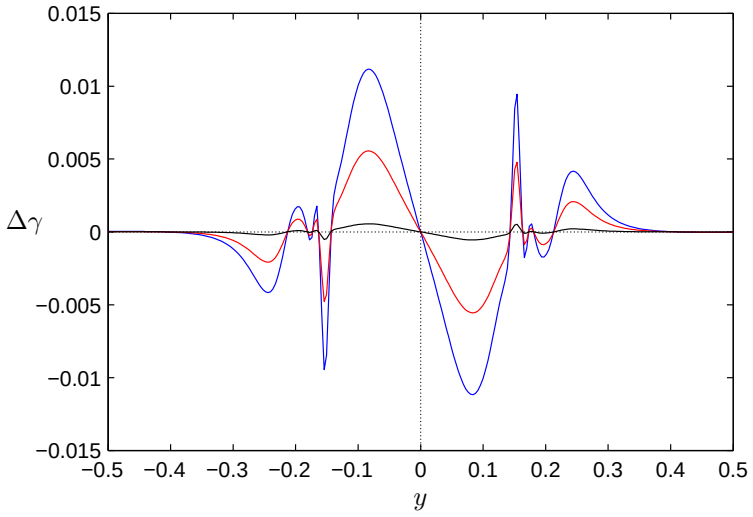


Figure 3.13: *Vortex sheet $\Delta\gamma^n$ observed at the beginning of the time step on the right wall ($x = L/2$) at $t = 0.348$ (maximum enstrophy) for the case at $Re = 1000$ with a no-slip condition and computed using DRK2-END-PW-NS on a 512×512 grid; the order of the sheet with respect to the time step Δt is examined: $\Delta t = 2 \cdot 10^{-4}$ (blue curve), $\Delta t = 1 \cdot 10^{-4}$ (red curve) and $\Delta t = 1 \cdot 10^{-5}$ (black curve).*

Evolution of the vortex sheet inside a time step

Next, we analyze the evolution of the vortex sheet during the time step computation in Fig. 3.14. The vortex sheet $\Delta\gamma^n$ at the beginning of the time step is compared with those observed at the end of the RK2 predictor and of the RK2 corrector. Moreover, a comparison is performed between DRK2-END-PW-NS and DRK2-CSUB-PW-NS.

At $t^{n+\frac{1}{2}}$, the predictor vortex sheet for DRK2-CSUB-PW-NS corresponds to $\Delta\gamma^{n+\frac{1}{2},*}$ and is associated to $\omega^{n+\frac{1}{2},*}$, i.e. it is precisely this sheet that is diffused at $t^{n+\frac{1}{2}}$ during the near-wall diffusion step, in order to yield $\omega^{n+\frac{1}{2}}$. In the case of DRK2-END-PW-NS, the predictor vortex sheet is $\Delta\gamma^{n+\frac{1}{2}}$ and it results from $\omega^{n+\frac{1}{2}}$. The sheet $\Delta\gamma^{n+\frac{1}{2}}$ is not used for any operation here, as the wall contribution is only computed at the end of the time step. The corrector sheet is $\Delta\gamma^{n+1,*}$ for both algorithms, and it is used for the near-wall diffusion step in both cases. Note that the remaining sheet $\Delta\gamma^{n+1}$ after the diffusion of $\Delta\gamma^{n+1,*}$ is then again of the order of $\Delta\gamma^n$.

Interestingly, the vortex sheets from DRK2-END-PW-NS and DRK2-CSUB-PW-NS are nearly indistinguishable. Despite the improved consistency of the time integration scheme (see Section 3.3.1), the slip error $\Delta\gamma^n$ observed when using DRK2-CSUB-PW-NS is not reduced compared to DRK2-END-PW-NS.

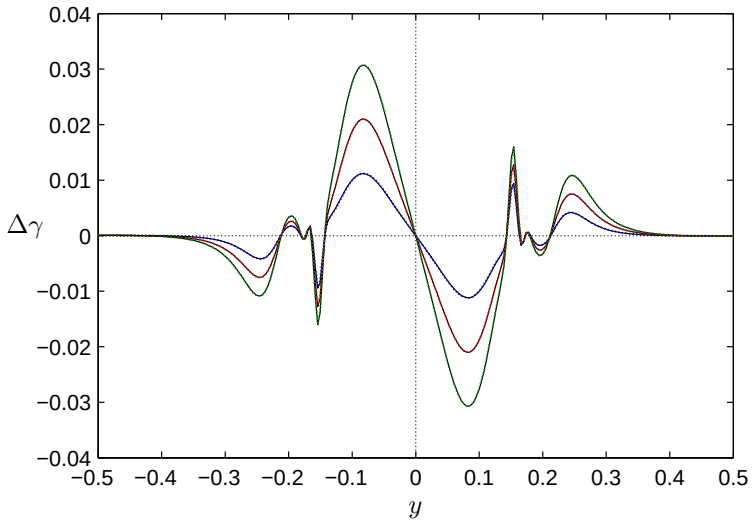


Figure 3.14: *Vortex sheet on the right wall ($x = L/2$) at $t = 0.348$ (maximum enstrophy) using $\Delta t = 2{\cdot}10^{-4}$; DRK2-CSUB-PW-NS (solid black curves) and DRK2-END-PW-NS (dotted colored curves); $\Delta\gamma^n$ at the beginning of the time step (blue), at the end of the RK2 predictor step (red; $\Delta\gamma^{n+\frac{1}{2}}$ for DRK2-END-PW-NS and $\Delta\gamma^{n+\frac{1}{2},*}$ before the wall diffusion step for DRK2-CSUB-PW-NS) and at the end of the RK2 corrector step (green; $\Delta\gamma^{n+1,*}$ before the near-wall diffusion step).*

One may further observe that the evolution of the vortex sheet from t^n to t^{n+1} seems to be linear in time. Fig. 3.15 confirms this observation by showing the evolution of the vortex sheet as a function of the time at $y = -8.40 \cdot 10^{-2}$, i.e. where $|\Delta\gamma|$ is maximum. A polynomial expression of $\Delta\gamma(t)$ is also provided (quadratic interpolation). Indeed, the second order coefficient is small compared to the first order coefficient (the ratio is approximately $1.6 \cdot 10^{-3}$).

Computing the wall flux using an iteration

All previous no-slip enforcing algorithms evaluate the wall flux based on the residual tangential slip velocity. Consider for example the DRK2-END-PW-NS scheme. The wall contribution is computed at the end of the time step and the vortex sheet to be diffused is $\Delta\gamma^{n+1,*}$. By construction, the following field is admissible

$$\omega^{n+1,*}(\mathbf{x}) + \oint_{\partial\Omega} \Delta\gamma^{n+1,*}(\mathbf{x}')\delta(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}\,.$$

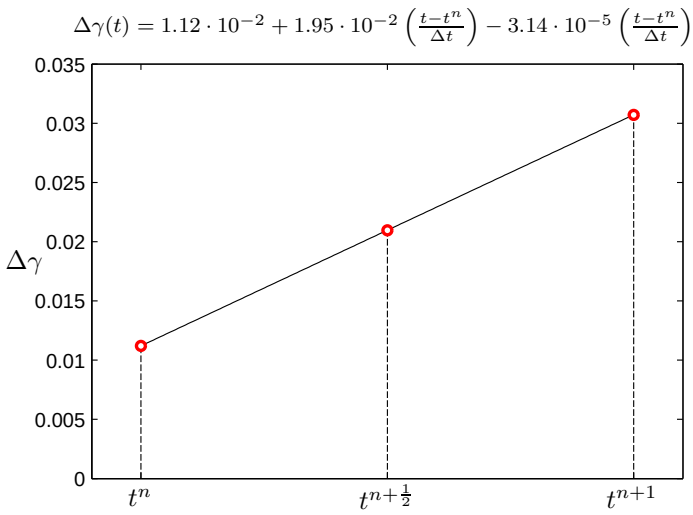


Figure 3.15: *Vortex sheet on the right wall ($x = L/2$ and $y = -8.40 \cdot 10^{-2}$) at $t = 0.348$ (maximum enstrophy) using DRK2-END-PW-NS and $\Delta t = 2 \cdot 10^{-4}$; the vortex sheet is shown before the near-wall diffusion step for the RK2 predictor ($\Delta\gamma^{n+\frac{1}{2},*}$) and the RK2 corrector ($\Delta\gamma^{n+1,*}$); the solid black curve corresponds to a quadratic interpolation (see above the figure for the expression of the polynomial).*

Yet, the diffusion of the vortex sheet slightly alters the vorticity distribution and the resulting field ω^{n+1} is no longer admissible, in the sense that it presents a slip error $\Delta\gamma^{n+1}$ (which is nevertheless smaller than $\Delta\gamma^{n+1,*}$).

An alternative criterion consists in requiring that the vorticity field *after* the diffusion of the vorticity flux is admissible. Stated equivalently, we seek a vortex sheet $\delta\gamma^{n+1,*} \neq \Delta\gamma^{n+1,*}$ such that the field ω^{n+1} resulting from the diffusion of the flux $\delta\gamma^{n+1,*}/\Delta t$ is admissible. Note that, as a consequence, the field combining $\omega^{n+1,*}(\mathbf{x})$ and $\delta\gamma^{n+1,*}$ is not admissible. This implicit problem can be solved by a simple iteration.

Starting with the vorticity field $(\omega^{n+1,*})^{(1)} \triangleq \omega^{n+1,*}$ and the associated vortex sheet $(\Delta\gamma^{n+1,*})^{(1)}$ (step $k = 1$), the following sequence of operations is performed while $\|(\Delta\gamma^{n+1,*})^{(k)}\|_\infty$ is above a prescribed tolerance:

1. Compute the wall flux $(\Delta\gamma^{n+1,*})^{(k)}/\Delta t$.

2. Evaluate the associated wall contribution $(\omega_w)^{(k+1)}$ determined by the diffusion of this flux (e.g. by using the explicit formulas PW [102]).

3. Add this contribution to the vorticity field: $(\omega^{n+1,*})^{(k+1)} \triangleq (\omega^{n+1,*})^{(k)} + (\omega_w)^{(k+1)}$.

4. Compute the vortex sheet $(\Delta\gamma^{n+1,*})^{(k+1)}$ associated to $(\omega^{n+1,*})^{(k+1)}$ and increment the iteration index ($k \leftarrow k+1$).

The previous approach can be recast into this formalism, as it consists in performing only the first iteration. If the procedure convergences, the sought vortex sheet is, by linearity of the diffusion operator,

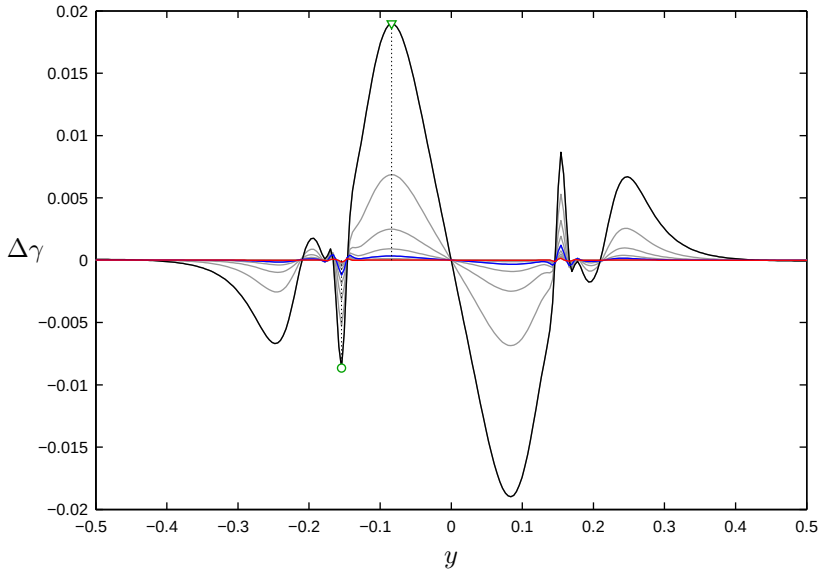
$$\delta\gamma^{n+1,*} = \lim_{q\to\infty} \sum_{k=1}^{q} (\Delta\gamma^{n+1,*})^{(k)} \ .$$

Fig. 3.16 shows the convergence of the iteration for the case at $Re = 1000$ using DRK2-END-PW-NS. For the present computation, the iteration is performed $N_{\text{it}} = 10$ times at every time step. Interestingly, one may observe in Fig. 3.16(b) that the convergence is slower when the second derivative of $\Delta\gamma^{n+1,*}$ is larger in absolute value. While using a smaller time step does not affect the convergence rate, one may nevertheless observe that the error level is reduced.

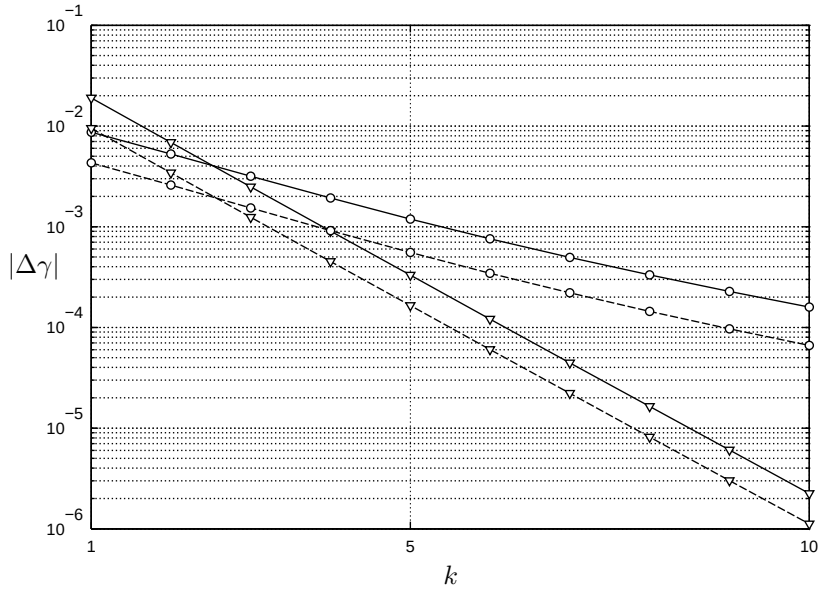
The above observations may be explained by the fact that the diffusion of the vortex sheet spreads the vorticity inside the flow domain. Canceling the peaks of the vortex sheet is more difficult, since the vorticity flux coming from the peak yields a smooth(er) surrounding vorticity field, due the diffusion process. The size of the surrounding area that is affected by the peak diffusion scales with $\sqrt{\nu\Delta t}$. Hence, the smaller the time step, the smaller the diffusion area and the lower the remaining vortex sheet after the diffusion.

Fig. 3.17 compares the vortex sheet $(\Delta\gamma^{n+1,*})^{(1)}$ and the "total" vortex sheet $\delta\gamma^{n+1,*}$ leading to an admissible vorticity field ω^{n+1}. Keep in mind that $\delta\gamma^{n+1,*}$ does not have a real physical meaning apart from the flux it is related to. Both sheets differ the most at the peaks, according to the previous discussion. Indeed, a higher flux is required locally in order to cancel out the sharp peaks.

In Fig. 3.18, the application of this iteration using $N_{\text{it}} = 10$ is also compared to the classical case $N_{\text{it}} = 1$ for the DRK2-END-PW-NS integration scheme. As expected, the initial vortex sheet (the slip error) is nearly completely canceled for the case with $N_{\text{it}} = 10$. Moreover, the vortex sheet observed at the end of the RK2 predictor and corrector are significantly lower for the case with $N_{\text{it}} = 10$ (the vortex sheet $(\Delta\gamma^{n+1,*})^{(1)}$ is shown for the corrector, whereas for the predictor it corresponds to $\Delta\gamma^{n+\frac{1}{2},*}$, as no wall contribution is computed inside the predictor and hence no iteration is required).



(a)



(b)

Figure 3.16: *Iteration over the vortex sheet* $\Delta\gamma^{n+1,*}$ *using DRK2-END-PW-NS so as to obtain* $(\Delta\gamma^{n+1,*})^{(k)} \to 0$ *when* $k \to \infty$ *(here* $k = 1, \ldots, 10$*): (a)* $(\Delta\gamma^{n+1,*})^{(k)}$ *as a function of* y *using* $\Delta t = 2 \cdot 10^{-4}$ *(*$k = 0$ *in solid black,* $k = 5$ *in solid blue,* $k = 10$ *in solid red and other values of* k *in solid grey); (b) convergence of* $|\Delta\gamma|$ *at* $y = -8.40 \cdot 10^{-2}$ *(*$\triangledown$*) and at* $y = -1.54 \cdot 10^{-1}$ *(*$\circ$*) using* $\Delta t = 1 \cdot 10^{-4}$ *(dotted lines) and* $\Delta t = 2 \cdot 10^{-4}$ *(solid lines).*

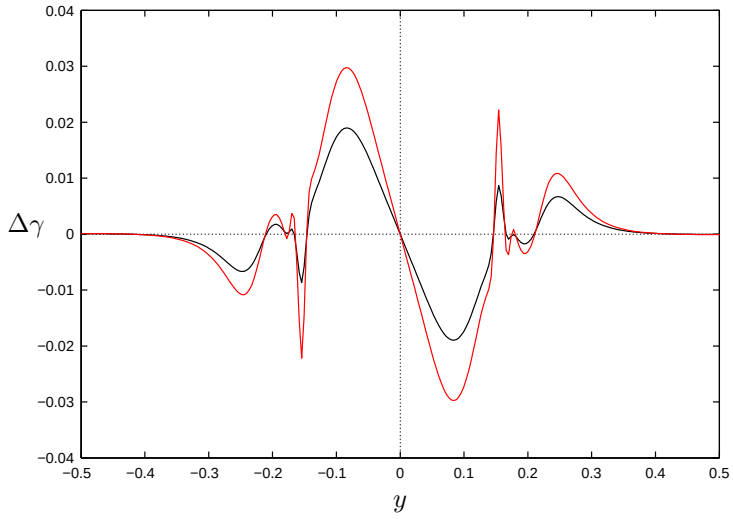


Figure 3.17: *Vortex sheet obtained by using the iteration* $(\Delta\gamma^{n+1,*})^{(k)}$ *and using DRK2-END-PW-NS with* $\Delta t = 2 \cdot 10^{-4}$ *and* $N_{it} = 10$*: actual slip error measured before the near-wall diffusion step* ($(\Delta\gamma^{n+1,*})^{(1)}$ *in solid black) and total vortex sheet to be diffused in order to obtain an admissible vorticity field* ω^{n+1} ($\delta\gamma^{n+1,*} \simeq \sum_{k=1}^{10}(\Delta\gamma^{n+1,*})^{(k)}$ *in solid red).*

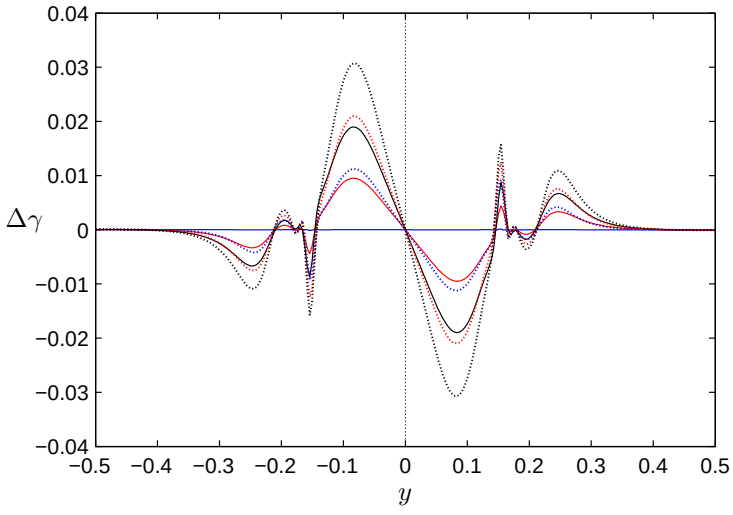


Figure 3.18: *Vortex sheet obtained by using the iteration* $(\Delta\gamma^{n+1,*})^{(k)}$ *and using DRK2-END-PW-NS with* $\Delta t = 2 \cdot 10^{-4}$*:* $N_{it} = 1$ *(dotted lines) and* $N_{it} = 10$ *(solid lines);* $\Delta\gamma^n$ *at the beginning of the time step (blue), at the end of the RK2 predictor step* $\Delta\gamma^{n+\frac{1}{2},*}$ *(red) and at the end of the RK2 corrector step* $(\Delta\gamma^{n+1,*})^{(1)}$ *(black).*

The time convergence study results for DRK2-END-PW-NS shown in Table 3.6 reveal that the convergence rate is first order, as expected (note that using DRK2-CSUB-PW-NS in conjunction with the iteration does not improve the convergence rate). However, compared to the results from Table 3.2 ($N_{\text{it}} = 1$), the error level at $t = 0.348$ is slightly reduced here (with $N_{\text{it}} = 10$). Indeed, it corresponds to the dipole collision, i.e. when the wall flux plays the most important role.

t	Δt	ϵ_2	order	ϵ_∞	order
$t = 0.25$	2×10^{-4}	3.0936×10^{-5}		1.6102×10^{-3}	
	1×10^{-4}	1.4425×10^{-5}	1.10	7.5541×10^{-4}	1.09
$t = 0.348$	2×10^{-4}	2.3433×10^{-4}		1.5253×10^{-2}	
	1×10^{-4}	1.1005×10^{-4}	1.09	7.1842×10^{-3}	1.09

Table 3.6: *Time convergence study on a* 512×512 *grid for the algorithm DRK2-END-PW-NS with* $N_{it} = 10$ *and for the case at* $Re = 1000$.

One must keep in mind that this scheme is computationally very expensive, as a Poisson solution is required at every iteration in order to evaluate the vortex sheet. Yet, it sheds some light in the process of computing a consistent wall flux for the no-slip enforcement.

Perspectives

Despite the fact that the above iteration has been designed to provide an exact no-slip condition (after the convergence of the iteration), the associated flux does not necessarily represent a higher order approximation of the actual vorticity flux. Indeed, ensuring that the vorticity field remains admissible does not guarantee that we will obtain a high order solution of the Navier-Stokes equations. Hence, the task of determining an appropriate expression for the flux is not completed yet.

One possibility that would improve the above iteration procedure consists in iterating over the whole time step so as to yield an equivalent vortex sheet (and hence an associated flux) that cancels the slip error at the end of the time step, while also accounting for the advection. Yet, this would come at en even higher cost compared to the above methodology.

For the time being, no solution was found for the computation of a wall flux that yields a second order time integration algorithm enforcing the no-slip condition.

Another line of inquiry consists in computing a flux that explicitly considers the evolution of the vortex sheet during the time step, by trying for example to account for the second order term (see Fig. 3.15). The objective is thus to obtain a slip error $\Delta\gamma^n$ that scales like Δt^2.

Following this idea, a methodology based on a Richardson extrapolation was tested. It consists in assuming that the flux computed at the end of the RK2 predictor step ($\Delta\gamma^{n+\frac{1}{2},*}/(\frac{1}{2}\Delta t)$) and at the end of the RK2 corrector ($\Delta\gamma^{n+1}/\Delta t$) have the same temporal leading error term. The recombination $(4\Delta\gamma^{n+\frac{1}{2},*} - \Delta\gamma^{n+1,*})/\Delta t$ then leads to a second order approximation of the flux at t^n. However, using this expression is unstable, even for very small time steps.

Since all approaches are first order in time, one may equivalently consider directly using the vortex sheets $\Delta\gamma^n$ and $\Delta\gamma^{n+\frac{1}{2}}$ for the flux evaluation, as they are available without additional Poisson solution. This allows to save 2 Poisson solutions from the 4 that are currently required. The interpretation of the vortex sheet as a flux correcting a slip error in the framework of the bulk-wall decomposition is then less straightforward, though, as everything is shifted in time.

3.4 Results for the dipole flow in a box at $Re = 1000$

In this section, some results are provided for the case at $Re = 1000$, using the DRK2-CSUB-PW-NS algorithm, thus with a no-slip condition at the cavity walls. For the following simulations, the time step is adaptive so as to satisfy $|\omega|_{\max}\Delta t < 0.25$, $\|\mathbf{S}\|_{\max}\Delta t < 0.25$ ($\mathbf{S} \triangleq \frac{1}{2}(\nabla\mathbf{u} + (\nabla\mathbf{u})^T)$ is the strain rate tensor) and $\nu\Delta t/h^2 \leq 0.2$. The redistribution frequency is also adaptive, i.e. the particles are redistributed when $\|\mathbf{S}\|_{\max}\Delta t^r \geq 0.1$, where Δt^r is the elapsed time since the last redistribution.

Fig. 3.19 shows the evolution of the dipole flow between $t = 0$ and $t = 1$, on a 512×512 grid. The collision of the vortex dipole with the right wall occurs at around $t \simeq 0.35$, while the impact of the weaker dipole, which is moving to the left, only happens after $t = 1$ and is not displayed here. After the impingement

on the right, the secondary vorticity generated at the wall rolls up with the dipole vorticity and in a second phase, some filamentary structures from the secondary vorticity remaining at the wall are attracted by the vortex cores between $t = 0.6$ and $t = 0.8$. From then on, a quasi equilibrium is reached (at least on the right side) and the diffusion becomes dominant.

Next, the vorticity iso contours are compared for two different grid resolutions ($N = 512$ and 1024) in Fig. 3.20. Up to $t = 0.6$, the results agree quite well, but later on, some significant differences appear in the roll-up phase. The advection of the filamentary structure is less well captured for the case with $N = 512$. Yet, for the same case, the nearly stationary vorticity core in the right corner of the figure (at around $(0.9, 0.1)$) is well captured.

Fig. 3.21 shows the spacial convergence of the kinetic energy E and the enstrophy $\mathcal{E}$ defined by

$$E \triangleq \frac{1}{2}\int_\Omega (u^2 + v^2)\, d\mathbf{x}$$
$$\mathcal{E} \triangleq \frac{1}{2}\int_\Omega \omega^2\, d\mathbf{x}\,.$$

Again, the results are shown for different grid resolutions ($N = 256$, $N = 512$, $N = 1024$ and $N = 2048$) and a comparison is made with reference results obtained on a 721×721 grid using a pseudo-spectral Chebyshev method [43] (note that the Chebyshev collocation points provide a better grid resolution near the wall and, hence, less points are required compared to a uniform grid). The coarsest case $N = 256$ is clearly under resolved, for both the energy and the enstrophy. As to the energy, one can observe that it is already converged for the case with $N = 512$. Regarding the enstrophy, the convergence is much slower, especially at the peak. Even the most refined case $N = 2048$ provides a peak value that slightly differs from the reference curve.

Fig. 3.22 further shows the mesh Reynolds number $Re_h \triangleq |\omega|_{\max} h^2/\nu$ for the different cases studied here above. A well-resolved simulation is characterized by $Re_h = \mathcal{O}(1)$. Only the most refined case with $N = 2048$, and in a lesser extent also with $N = 1024$, actually fulfill this criterion. The highest mesh Reynolds number is observed at the enstrophy peak, during the primary collision (at $t \simeq 0.35$), as expected.

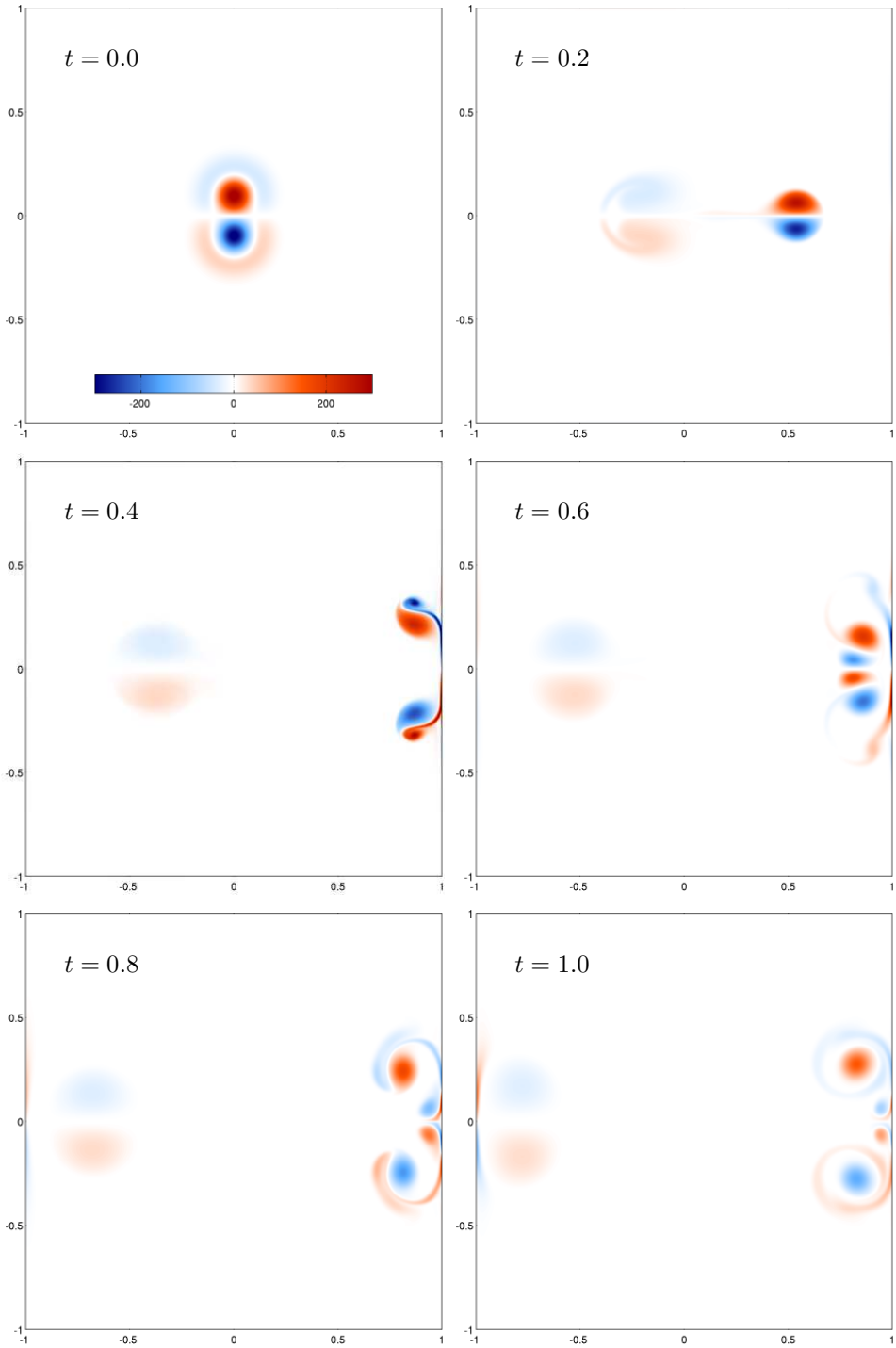


Figure 3.19: *Time evolution of the vorticity for the dipole flow inside a cavity at* $Re = 1000$*; results are obtained on a* 512×512 *grid, using DRK2-CSUB-PW-NS with an adaptive time step* Δt*.*

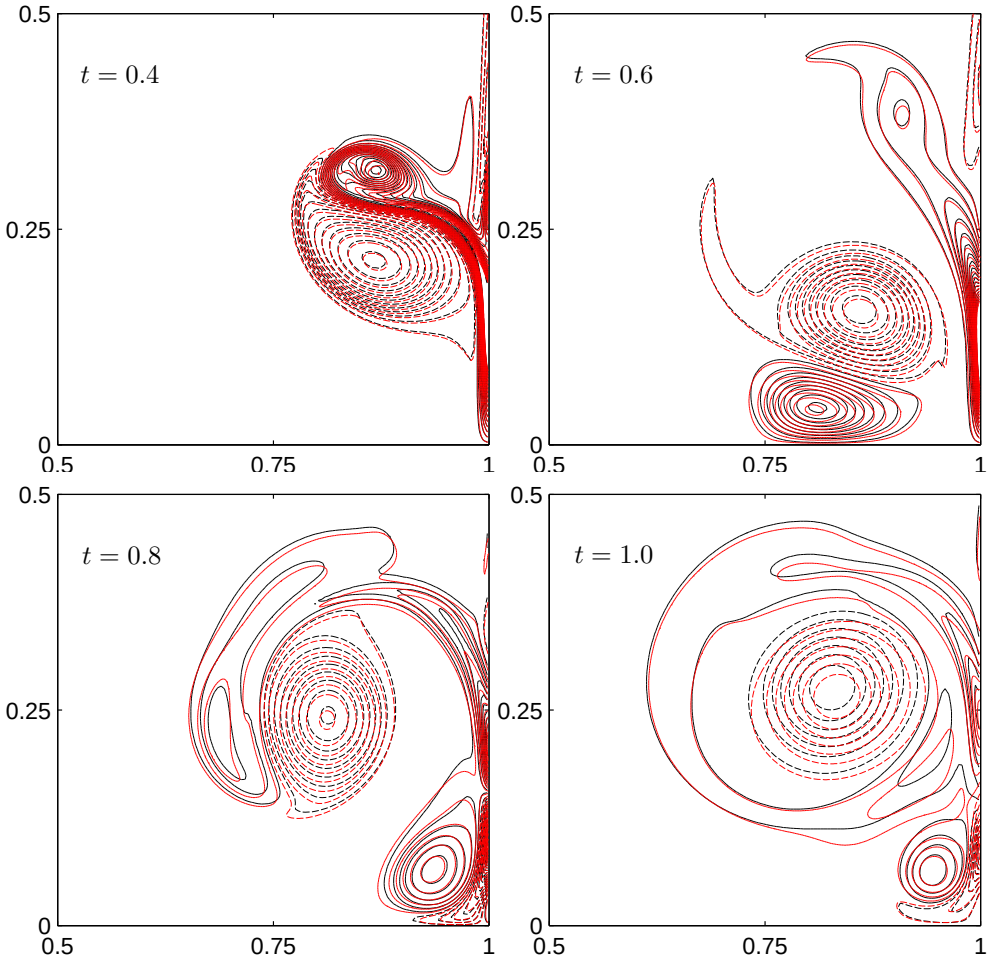
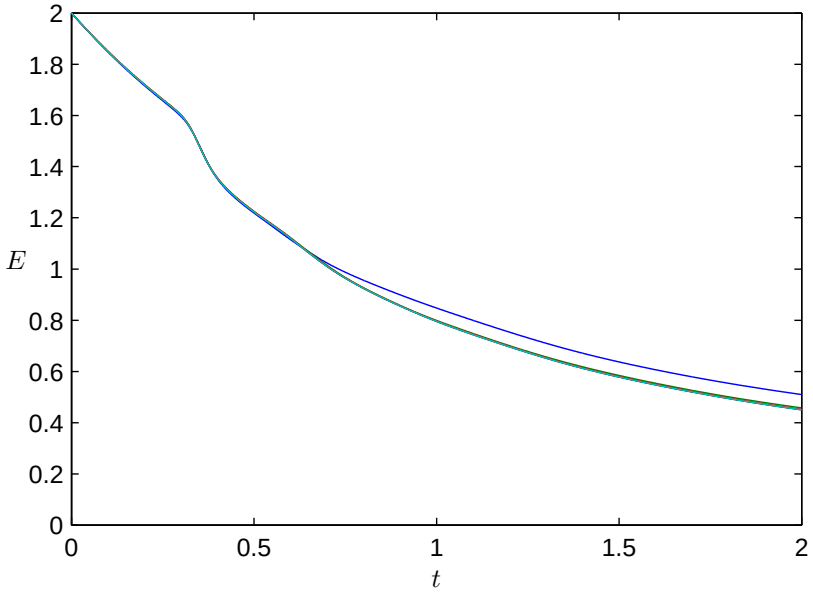
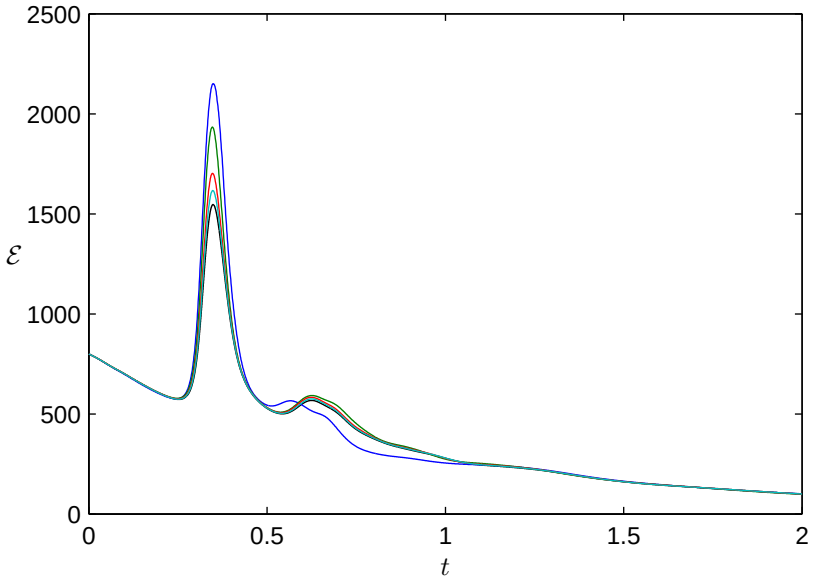


Figure 3.20: *Zoom of the vorticity iso contours (with $\omega = \ldots, -30, -10, 10, 30, \ldots$) in the upper right quarter of the domain for the dipole flow inside a cavity at $Re = 1000$; comparison between $N = 512$ (black contours) and $N = 1024$ (red contours), positive valued contours are represented by dashed contours, whereas negative valued contours by solid contours.*



(a)



(b)

Figure 3.21: *Spatial convergence of the global diagnostics as a function of the time for the dipole flow in a cavity at $Re = 1000$: (a) kinetic energy and (b) enstrophy; $N = 256$ (blue line), $N = 512$ (green line), $N = 1024$ (red line), $N = 2048$ (cyan line) and reference (black line).*

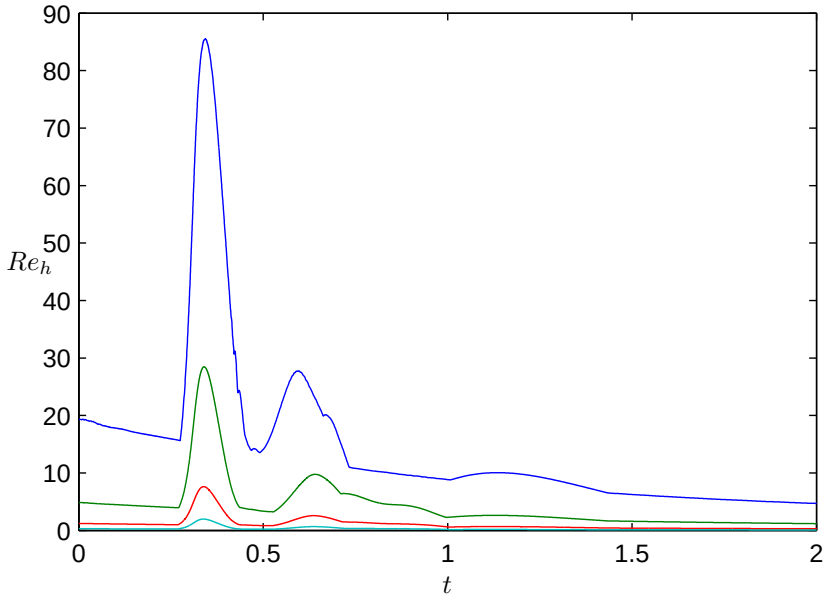


Figure 3.22: *Mesh Reynolds number as a function of the time for the dipole flow in a cavity at* $Re = 1000$*:* $N = 256$ *(blue line),* $N = 512$ *(green line),* $N = 1024$ *(red line) and* $N = 2048$ *(cyan line).*

Chapter 4

Development of an unbounded immersed interface Poisson solver for vortex particle-mesh methods

The content of this chapter was submitted in September 2013 and accepted in March 2014 for publication as an article [88] in the journal *Computers & Fluids*. It is entitled "*An immersed interface solver for the 2-D unbounded Poisson equation and its application to potential flow*"; the authors are Y. Marichal, P. Chatelain and G. Winckelmans. The article is here mostly reproduced as it was submitted.
Part of this work was also published in *Procedia IUTAM* in the framework of the "IUTAM Symposium on Particle Methods in Fluid Mechanics" in October 2012 (Copenhagen), see [87].

Abstract This paper presents a novel algorithm to solve the 2-D potential flow past complex geometries with circulation in unbounded domain and in the presence of a given vorticity field. It is based on a Poisson solver that combines

two components: the immersed interface method to enforce the boundary condition on each inner boundary and the James-Lackner algorithm to compute the outer boundary condition consistent with the unbounded domain solution. The algorithm is here based on second order finite differences and it requires solely 1-D stencil corrections; this makes the immersed interface part of the present method easily extendable to higher dimensional problems. The treatment of the outer boundaries requires an iterative boundary potential method. The algorithm is validated, by means of grid convergence studies, on the flow past multiple bodies. The results confirm the second order accuracy everywhere. The algorithm is also self consistent as "all is done on the grid" (thus without using a vortex panel boundary element method in addition to the grid). For cusped airfoils, a consistent way to enforce the *Kutta-Joukowsky condition* is also presented. The present algorithm constitutes a crucial building block towards an immersed interface-enabled vortex particle-mesh method for the computation of unsteady viscous flows, with boundary layers, detached shear layers and wakes. A possible extension to 3-D problems is also briefly discussed.

4.1 Introduction

The study and the development of solution techniques for Poisson equations is a recurring research topic as they appear in many areas of mathematical and computational physics, e.g. electromagnetism, continuum mechanics and theoretical physics. This wide range of applications has given rise to many different solution techniques. We here consider two aspects of the problem that still remain a challenge today: taking into account irregular interior boundary geometries and providing outer boundary conditions that are compatible with the solution of the equation in an unbounded domain.

These two key components can also be found in a more specific context, in the framework of Computational Fluid Dynamics for the simulation of the flow past bluff bodies in an unbounded domain (external flow aerodynamics). Moreover, in incompressible fluid dynamics, one is always constrained to solve at least one Poisson equation per time step and obtaining its solution represents the most expensive computational step. The choice of the present application, namely potential flow in the presence of a given vorticity field, follows this observation and is motivated by the fact that it represents one of the computational steps required for the simulation of unsteady bluff body flows using

a viscous vortex particle method combined with a vortex panel method [58] (boundary element method), as explained for example in [72, 102].

Whether for Poisson equations, for the Navier-Stokes equations in fluid dynamics, or for other types of PDE's, great efforts have been made in order to incorporate irregular boundary geometries inside the so-called structured grid methods (finite difference methods, spectral methods, etc.).

One of the first attempts to achieve this goal in the context of fluid dynamics was undertaken by Peskin [101]. It is considered to be the pioneering work for a class of methods known as the immersed boundary methods [100, 92]. This class of methods provides a discrete representation of the singular source term acting at the irregular boundary which is immersed inside the computational domain. Hence, considering the flow past moving bodies which are either rigid or even deforming is greatly simplified as the grid must not be adapted to fit the boundaries.

Based on a similar approach and following the same goal, Brinkman-type penalization methods have also emerged [3]. The latter approach can also be applied in combination with different kinds of discretization methods, i.e. spectral methods [66], finite differences [3] or vortex particle methods [32, 109, 50]. However, the regularization of the singular source term over a few grid cells entails a smearing of the solution near the interface, as has been shown for example in [80], and can lead to a loss of accuracy [125]. For high Reynolds number flows, this can be problematic because the boundary layers may not be captured properly. Therefore, methods capturing sharp interfaces have been developed, such as the ghost-cell approach [85], the cut-cell method [24] or the hybrid Cartesian/immersed boundary method (HCIB [54, 51]).

In the same spirit, immersed interface methods have appeared in the literature [77] as a consistent way to take into account possible jumps of the unknown at the interface, e.g. by modifying the finite difference stencil in the vicinity of the interface. The original stencil correction technique [77] uses multiple-dimensional Taylor series. This can however lead to stability issues in the resulting linear system resolution and requires among others a careful choice of the stencil nodes [78]. Other methods use instead one-dimensional Taylor series (*dimension splitting* approach, see [129, 80, 21]). The method developed here is based on the latter approach from [80], as the stencil correction procedure is more easily applicable to higher dimensional problems: for each grid direction, the stencil corrections are derived at the intersections of the in-

terface with the different grid axes along the corresponding direction, according to the prescribed boundary condition. The present approach thus provides a special treatment for grid nodes close to the interface. This feature is shared by all sharp-interface capturing methods. The cut-cell method modifies the grid cell geometry near the interface, the ghost-cell approach extends the solution across the boundary and the HCIB method interpolates the solution on the interior grid nodes closest to the interface using the solution and the boundary condition.

The other key component considered here is the unbounded outer boundary condition. The most natural way to take this into account for a Poisson equation is to perform the convolution of the source term with the free space Green's function, either through direct summation and ideally by fast summation (fast multipole method in two [55] and three dimensions [6]). Another class of methods is based on fast Fourier transforms [60, 17] but, as the immersed interface approach requires local modifications of the spatial differential operator, it is hardly applicable here.

We follow a different approach based on the James-Lackner algorithm [62, 74], which has been further improved in [90] and which additionally remains compatible with mesh refinement techniques. The solution procedure is based on two problems, the first one being obtained by imposing homogeneous Dirichlet conditions on the outer boundary and the second problem computes correction charges at the outer boundary which result in an inhomogeneous Dirichlet condition being consistent with the unbounded character of the solution. Miller [91] extended the method to include some irregular interior boundaries held at a fixed potential. The presence of interior boundaries with unknown surface charges results in a method which is intrinsically iterative.

The present approach combines the work of Linnick and Fasel [80] and Miller [91] and generalizes the algorithm to allow the computation of potential flow past multiple bodies accounting for a given compact vorticity field. In this case, the streamfunction is the superposition of a function linear in space (free stream flow field) and an unbounded solution of the Poisson equation. The streamfunction solution is constant in the interior boundaries but the value of this constant is not known a priori. This value is determined by a supplementary constraint about the circulation of the flow around each solid body.

In this chapter, we thus propose a second order finite difference method to compute the solution of a two-dimensional Poisson equation in an unbounded domain with interior boundaries of complex geometry.

The underlying objective of the present work is to integrate the resulting approach within a vortex particle method [34, 132]. Vortex particle methods perform very well for unbounded vortical flows but accounting for solid bodies remains difficult. Penalization methods have been used (as mentioned above [32, 109, 50]). A different technique consists in combining the Poisson solver with a boundary element method to account for the presence of the walls, either by combining it with a vortex panel method [72, 102, 103, 83], either by computing equivalent sources of velocity potential [35, 104]. This procedure allows to recover from a given vorticity field a velocity field that also respects the no-through flow condition at the surface of the body. In the specific context of vortex particle-mesh (VPM) methods, relying simultaneously on particles and on a grid [23, 28, 83], the present approach is a more consistent alternative to the combination of the finite difference Poisson solver with the boundary element method, as it preserves the order of convergence up to the wall.

In Section 4.2, the governing equations for the elliptic problem are given. Section 4.3 is devoted to the methodology description: in Section 4.3.1, the immersed interface approach is detailed in order to take into account the interior boundaries with prescribed outer boundary conditions; in Section 4.3.2, the iterative boundary potential method is detailed so as to obtain the correct outer boundary conditions; in Section 4.3.3, the global algorithm is given, and Section 4.3.4 briefly presents a possible extension of the approach to three-dimensional problems. Section 4.4 is devoted to the validation of the methodology for several potential flow problems.

Results are first compared with the analytical solution for the flow past a cylinder. The convergence behavior of the approach is assessed and the error value is compared with that obtained using a vortex panel method. A convergence study is also performed for the prediction of the added mass coefficient of an elliptical cylinder. Next, the order of convergence is assessed for the flow past an airfoil with a cusped trailing edge. This case requires the development of a supplementary equation to enforce the *Kutta-Joukowsky condition*. Finally, the ability of the method to take into account multiple bodies as well as more general geometries is also illustrated and validated.

4.2 Problem statement

In many applications of computational fluid dynamics, the solution of a Poisson equation is required. In particular, the operation of computing a velocity field $\mathbf{u}$ associated to a given vorticity field $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is required when working with the velocity-vorticity formulation of the Navier-Stokes equations for incompressible flows ($\nabla \cdot \mathbf{u} = 0$)

$$\frac{D\boldsymbol{\omega}}{Dt} \triangleq \frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\nabla \mathbf{u}) \cdot \boldsymbol{\omega} + \nu \nabla^2 \boldsymbol{\omega} \qquad \nabla^2 \boldsymbol{\Psi} = -\boldsymbol{\omega} \,,$$

with ν the kinematic viscosity of the fluid. Indeed, the velocity field $\mathbf{u}$ can be linked to the vorticity $\boldsymbol{\omega}$ through the above Poisson equation for the stream-function $\boldsymbol{\Psi}$, as $\mathbf{u} = \nabla \times \boldsymbol{\Psi}$ and $\nabla \cdot \boldsymbol{\Psi} = 0$ (Lorenz' gauge).

The flow past a non-moving body with boundary $\partial\Omega_b$ is sketched in Fig. 4.1, in the 2-D case where $\boldsymbol{\Psi} = \Psi\, \hat{\mathbf{e}}_z$ and $\boldsymbol{\omega} = \omega\, \hat{\mathbf{e}}_z$. The flow domain is then $\Omega_f \triangleq \mathbb{R}^2 / \Omega_b$ and the boundary conditions are $\lim_{|\mathbf{x}|\to\infty} \mathbf{u} = \mathbf{U}_\infty$ (with $\mathbf{U}_\infty$ a constant free stream flow) and $\mathbf{u} = 0$ on $\partial\Omega_b$ (no-slip condition). The translation of the no-slip condition into vorticity formulation is not straightforward. Usually, the Poisson equation is solved with a *no-through flow* condition on $\partial\Omega_b$. This is actually the core of the present work and we refer to the problem as finding the *potential flow* that cancels the through flow induced by the vorticity field (see also Section 3.2.1). The potential velocity field however still presents a residual tangential slip velocity at the wall and the way to enforce a no-slip condition based on this is further detailed in [72, 102] and Section 3.3.

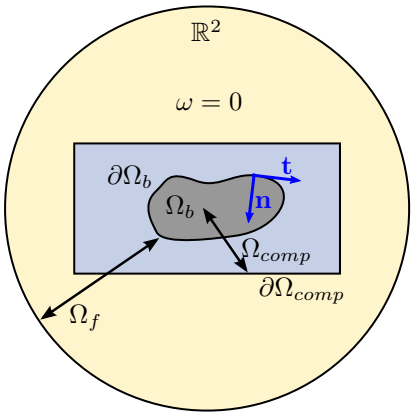


Figure 4.1: *Sketch of the different domains.*

By abusing the notation slightly, we call $\mathbf{u}$ the potential velocity that satisfies the no-through flow condition $\mathbf{u}\cdot\mathbf{n}=0$ on $\partial\Omega_b$, with $\mathbf{n}$ the outward pointing normal of Ω_f. In 2-D, this is equivalent to saying that the streamfunction $\Psi=\overline{\Psi}$ is constant along the boundary $\partial\Omega_b$, with $\overline{\Psi}$ a priori unknown. The slip velocity generated by the latter condition can be seen as an infinitely thin vortex sheet γ (interior boundary charge). The degree of freedom associated with the unknown $\overline{\Psi}$ is filled by prescribing the circulation $\Gamma \triangleq \oint_{\partial\Omega_b}\gamma(\mathbf{x}')\,d\mathbf{x}'$.

Moreover, we assume that the vorticity field is compact and completely included inside the computational domain Ω_{comp} (with $\Omega_b\subset\Omega_{comp}$), as shown in Fig. 4.1, and we extend the solution to Ω_b with $\Psi=\overline{\Psi}$. By means of the decomposition $\Psi \triangleq \Psi_b+\Psi_\infty$ with $\Psi_\infty=(\mathbf{U}_\infty\times\mathbf{x})\cdot\hat{\mathbf{e}}_z$ the free stream contribution, we can formally write the set of equations for the unknown body contribution Ψ_b in Ω_{comp}

$$\begin{aligned}
\nabla^2\Psi_b &= -\omega-\oint_{\partial\Omega_b}\gamma(\mathbf{x}')\,\delta(\mathbf{x}-\mathbf{x}')\,d\mathbf{x}' \quad \text{in } \Omega_{comp} \\
\Psi_b &= \overline{\Psi}-\Psi_\infty \quad \text{on } \partial\Omega_b \text{ such that } \oint_{\partial\Omega_b}\gamma(\mathbf{x}')\,d\mathbf{x}'=\Gamma \\
\Psi_b &= \Psi-\Psi_\infty \quad \text{on } \partial\Omega_{comp}\,,
\end{aligned} \tag{4.1}$$

with $\overline{\Psi}$ and the outer boundary condition Ψ on $\partial\Omega_{comp}$ both a priori unknown. Using the free space Green's function for the 2-D Poisson equation $G(\mathbf{x})=\frac{1}{2\pi}\log(|\mathbf{x}|/L)$ (L is a reference length), the extended solution in $\mathbb{R}^2$ reads

$$\Psi(\mathbf{x})=\Psi_\infty(\mathbf{x})-\int_{\Omega_{comp}}\omega(\mathbf{x}')\,G(\mathbf{x}-\mathbf{x}')\,d\mathbf{x}'-\oint_{\partial\Omega_b}\gamma(\mathbf{x}')\,G(\mathbf{x}-\mathbf{x}')\,d\mathbf{x}'\,, \tag{4.2}$$

where γ is part of the solution and thus also a priori unknown.

The problem statement is written here in the context of the particular case of potential flows. It nevertheless generalizes to a broader spectrum of applications, where the total boundary charge is imposed, contrary to the enforcement of a constant Dirichlet boundary condition on the inner boundary $\partial\Omega_b$.

4.3 Methodology

The computation of the solution to the problem stated in Section 4.2 requires the following two components: an immersed interface approach (IIM) for the inner boundary condition and a boundary potential method (BPM) for the

unbounded outer boundary condition. The potential flow application requires a decomposition of the solution in a free stream component Ψ_∞ and a body contribution Ψ_b, in order to account for the unbounded character of the global solution. Hence, we have to provide an appropriate treatment of an inner boundary condition which is a linear function with unknown constant level. The two approaches IIM and BPM are detailed respectively in Sections 4.3.1 and 4.3.2. The global algorithm is then given in Section 4.3.3.

4.3.1 Immersed interface approach for the interior boundary

As an example, let us first consider a one-dimensional function $f(x) \triangleq f^{(0)}(x) \in C_\infty(\mathbb{R}\backslash\{x_\alpha\})$ and some underlying grid points x_q. $f(x)$ and its derivatives $f^{(k)}(x)$ may be discontinuous in the immersed interface $x_\alpha \in [x_i, x_{i+1})$, as can be seen in Fig. 4.2.

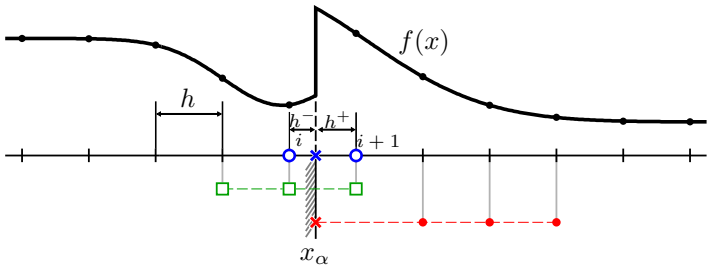


Figure 4.2: *Sketch of the 1D immersed interface problem: position of the immersed interface ($x = x_\alpha$, blue cross); irregular points i and $i+1$ affected by the correction term J_α (blue circles); stencil nodes for the uncorrected second derivative at x_i (green squares); stencil nodes for the correction terms J_α^+ and J_α^- (red bullets) and boundary condition (red cross).*

Following [80] or [129] and using generalized Taylor series that are valid across the interface, a corrected finite difference scheme can still be written at x_i, without affecting the order of accuracy. For the second derivative of f, we have

$$\begin{aligned} f^{(2)}(x_i) &= R_{-1}\, f(x_{i-1}) + R_0\, f(x_i) + R_{+1}\left(f(x_{i+1}) - J_\alpha^+\right) + \mathcal{O}(h^2) \\ J_\alpha^+ &= [f^{(0)}]_\alpha + \frac{h^+}{1!}[f^{(1)}]_\alpha + \frac{(h^+)^2}{2!}[f^{(2)}]_\alpha + \frac{(h^+)^3}{3!}[f^{(3)}]_\alpha\,, \end{aligned}$$

with $h^+ \triangleq x_{i+1} - x_\alpha$ and $[f^{(k)}]_\alpha = f^{(k)}(x_\alpha^+) - f^{(k)}(x_\alpha^-)$ the jump in the k^{th} derivative of $f(x)$. For a uniform grid with spacing $h \triangleq x_{i+1} - x_i$, the

coefficients $\{R_{-1}, R_0, R_{+1}\}$ correspond to the classical centered second order scheme $\{1, -2, 1\}/h^2$. The corrected scheme at x_{i+1} reads

$$\begin{aligned} f^{(2)}(x_{i+1}) &= R_{-1}\left(f(x_i) - J_\alpha^-\right) + R_0\, f(x_{i+1}) + R_{+1}\, f(x_{i+2}) + \mathcal{O}(h^2) \\ J_\alpha^- &= -[f^{(0)}]_\alpha + \frac{h^-}{1!}[f^{(1)}]_\alpha - \frac{(h^-)^2}{2!}[f^{(2)}]_\alpha + \frac{(h^-)^3}{3!}[f^{(3)}]_\alpha \ , \end{aligned}$$

with $h^- \triangleq x_\alpha - x_i$. The nodes i and $i+1$ are called *irregular* in the immersed interface literature because the uncorrected stencil crosses the interface in both cases and hence requires the addition of the correction terms J_α^+ and J_α^-. All nodes whose stencil does not cross the interface are called *regular*.

The correction terms J_α^+ and J_α^- provide a mean to enforce the boundary condition at $x = x_\alpha$. Considering the target problem in 2-D, we may identify one of both regions $x < x_\alpha$ and $x > x_\alpha$ as the body domain Ω_b, depending on the spatial configuration of the computational domain. Without loss of generality, $x < x_\alpha$ will here represent Ω_b. Inside Ω_b, f is assumed to be known and thus also $f^{(k)}(x_\alpha^-)$. However outside of Ω_b, f is part of the solution. The derivatives $f^{(k)}(x_\alpha^+)$ are therefore computed using one-sided finite differences, only taking into account stencil nodes inside Ω_f $(x \geq x_\alpha)$ and the Dirichlet boundary condition f_α

$$\begin{aligned} f^{(k)}(x_\alpha^+) = (S_0^k)_\alpha\, f_\alpha + (S_0^k)_2\, f(x_{i+2}) + (S_0^k)_3\, f(x_{i+3}) \\ + (S_0^k)_4\, f(x_{i+4}) + \mathcal{O}(h^{4-k}) \ , \end{aligned}$$

which is in agreement with the required accuracy of the corrected scheme. The point x_{i+1} has not been included in the scheme for numerical stability reasons (see Fig. 4.2), as discussed in [80]. For more details about the notations and the computation of these 1-D schemes, see Appendix A.

Two-dimensional operators, such as the Laplacian $\nabla^2(\cdot)$, are handled by correcting the schemes for the derivatives along the different grid directions individually (*dimension splitting* approach), i.e. $\partial^2 f/\partial x^2$ and $\partial^2 f/\partial y^2$ are treated like one-dimensional stencils. The derivative jumps required by the correction terms are then computed at the intersections of the interface with the grid lines, which are called *control points* in the immersed interface literature. The correction actually widens the stencil compared to the classical 5-point scheme but the system that has to be solved is still linear, as expected. The extent of the correction stencils implies some restrictions on the geometry of

the boundary as the one-sided stencil may not cross the boundary again. For smooth boundaries, it can be prevented by grid refinement.

Similarly to the one-dimensional case, the solution is assumed to be known inside Ω_b. Indeed, according to Eq. (4.1), we have $\Psi_b = \overline{\Psi} - \Psi_\infty$ inside the body (keep in mind that the value of $\overline{\Psi}$ is not known a priori). The solution is therefore continuous but its derivatives are not.

Hence, the corrected scheme introduces the supplementary unknown $\overline{\Psi}$ and adds the terms containing Ψ_∞ to the right hand side of the linear system. As a consequence, one has to complete the system by imposing the total boundary charge, i.e. the circulation $\oint_{\partial\Omega_b} \gamma(\mathbf{x}')\ d\mathbf{x}' = \Gamma$. Note that the vortex sheet γ is part of the solution and is not computed explicitly during the solution procedure, as it is implicitly taken into account by the derivative jumps of Ψ_b (the value of γ can be obtained as a post processing step, though; see Chapter 7).

Yet, prescribing the circulation is carried out straightforwardly after realizing that the correction terms actually behave like an additional bulk vorticity field [14]. Indeed, if one gathers all the correction terms $J_{\alpha,k}$ (i.e. $J^+_{\alpha,k}$ or $J^-_{\alpha,k}$) that have to be applied at a certain grid point $\mathbf{x}_{ij}$ near the interface and removes the contribution of Ψ_∞, one can write the following equation

$$\left(\nabla^2\Psi\right)^h_{ij} = -\omega_{ij} - \sum_k \left(-R_k J_{\alpha,k}\right) \triangleq -\omega_{ij} - \left(\omega_\gamma\right)_{ij} \ , \tag{4.3}$$

with $\left(\nabla^2\Psi\right)^h_{ij}$ the classical 5-point stencil for the Laplacian. ω_γ can be equivalently seen as a discrete projection of the singular vortex sheet γ onto the grid nodes. It is only non-zero in the vicinity of the body interface and it plays a role similar to the sources of potential velocity computed in [35] and [104]. In those works, the local accuracy is limited to first order for the no-through flow condition enforcement, even though it achieves second order, once incompressibility and the no-slip condition are taken into account. In our case, these additional source terms are designed in a fashion that is consistent with the numerical discretization scheme and hence it preserves the second order accuracy up to the wall.

For a uniform and isotropic grid, the circulation constraint is then:

$$\Gamma = \oint_{\partial\Omega_b} \gamma(\mathbf{x}')\ d\mathbf{x}' = \int_{\Omega_{comp}} \omega_\gamma(\mathbf{x}')\ d\mathbf{x}' \simeq \sum_{i,j} \left(\omega_\gamma\right)_{ij} h^2 \ , \tag{4.4}$$

with the definition $\omega_\gamma(\mathbf{x}) \triangleq \oint_{\partial\Omega_b} \gamma(\mathbf{x}')\ \delta(\mathbf{x}-\mathbf{x}')\ d\mathbf{x}'$ in the continuous case.

This approach shows that it can be easily generalized for the flow past multiple bodies because we introduce one equation and one unknown per body, similarly to [14].

The resulting system is solved using the library HYPRE [46, 45]; in particular the GMRES solver is combined with an algebraic multigrid preconditioning. The tool developed in this section hence allows to solve the problem Eq. (4.1), provided the outer boundary condition on $\partial\Omega_{comp}$ is known. Obtaining this boundary condition is the subject of the next section.

4.3.2 James-Lackner algorithm for the outer boundary

The aim of this Section is to compute a field with outer boundary conditions that are compatible with the solution of a Poisson equation in unbounded domain. The algorithm presented here follows that of Miller [91] and is based on the ideas introduced by James [62] and Lackner [74]. If we consider again the potential flow problem Eq. (4.1), the difference with Miller's problem equations comes from the boundary condition at the interior boundary $\partial\Omega_b$. Miller imposes an a priori known constant Dirichlet condition whereas, in our case, we enforce the circulation. The modifications of the algorithm are given hereafter.

In the spirit of the James-Lackner algorithm, let us introduce the additional decomposition

$$\Psi_b \triangleq \Psi_0 + \delta\Psi_b \ ,$$

where Ψ_0 is the solution in Ω_{comp} of the following problem with homogeneous outer Dirichlet boundary conditions

$$\begin{aligned} \nabla^2\Psi_0 &= -\omega - \oint_{\partial\Omega_b} \gamma_0(\mathbf{x}')\ \delta(\mathbf{x}-\mathbf{x}')\ d\mathbf{x}' \quad \text{in} \ \ \Omega_{comp} \\ \Psi_0 &= \overline{\Psi}_0 - \Psi_\infty \ \ \text{on} \ \ \partial\Omega_b \ \ \text{such that} \ \ \oint_{\partial\Omega_b} \gamma_0(\mathbf{x}')\ d\mathbf{x}' = \Gamma \\ \Psi_0 &= 0 \quad \text{on} \ \ \partial\Omega_{comp} \ . \end{aligned} \tag{4.5}$$

This problem can be solved straightforwardly using the tools developed in Section 4.3.1, keeping in mind that both γ_0 and $\overline{\Psi}_0$ are part of the solution. However, even if the vortex sheet γ_0 generated on $\partial\Omega_b$ has the correct integral value Γ, it is not equal to the solution γ, due to the fact that the outer boundary conditions are different from those of the problem Eq. (4.1) (for the same reason

we also have $\overline{\Psi}_0 \neq \overline{\Psi}$). By the minimum-maximum principle of the Laplace equation, we also have $\Psi_0 = 0$ outside of Ω_{comp}. As a consequence, an artificial vortex sheet γ_{comp} appears on $\partial\Omega_{comp}$ and the solution of problem Eq. (4.5), also valid in $\mathbb{R}^2$, reads

$$\Psi_0(\mathbf{x}) = -\int_{\Omega_{comp}} \omega(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' - \oint_{\partial\Omega_b} \gamma_0(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' - \oint_{\partial\Omega_{comp}} \gamma_{comp}(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' \,.$$

In order to understand the nature of γ_{comp}, we can rewrite the latter equation for $\mathbf{x}$ outside of Ω_{comp} (or equivalently on $\partial\Omega_{comp}$)

$$\oint_{\partial\Omega_{comp}} \gamma_{comp}(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' = -\int_{\Omega_{comp}} \omega(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' - \oint_{\partial\Omega_b} \gamma_0(\mathbf{x}') G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' \,.$$

This shows that the contributions of ω and γ_0 to the far-field solution Ψ are recovered by the artificial vortex sheet γ_{comp}. The only far-field contribution that is missing is the vortex sheet $\gamma - \gamma_0$ that complements γ_0 in order to yield the vortex sheet solution γ.

Indeed, considering now Eq. (4.2) and the definition of Ψ_b, the required correction inside Ω_{comp} reads

$$\delta\Psi_b(\mathbf{x}) = \oint_{\partial\Omega_{comp}} \gamma_{comp}(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' - \oint_{\partial\Omega_b} (\gamma-\gamma_0)(\mathbf{x}')\, G(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}' \,.$$

By definition, $\delta\Psi_b = \overline{\Psi} - \overline{\Psi}_0$ on $\partial\Omega_b$ and the total circulation of the correction sheet $(\gamma - \gamma_0)$ equals zero.

The first term of the correction $\delta\Psi_b$ is easily accounted for. Indeed, the value of γ_{comp} can be deduced from Ψ_0, using Green's third identity written in Ω_{comp}, as shown in [91]:

$$\gamma_{comp}(\mathbf{x}) = \nabla\Psi_0(\mathbf{x}) \cdot \mathbf{n} = \frac{\partial\Psi_0}{\partial n} \,,$$

with $\mathbf{n}$ the normal pointing outwards of Ω_{comp}.

The second term in $\delta\Psi_b$, consisting of the vortex sheet correction $(\gamma - \gamma_0)$, is more difficult to handle. In absence of body in the domain, this term simply vanishes and we recover the original two-step James-Lackner algorithm [62, 74].

In the presence of a body, we need to iterate because the vortex sheet γ is part of the final solution Ψ_b, as explained in [91]:

$$\begin{aligned} \Psi_b^{(k+1)}(\mathbf{x}) &= \Psi_0(\mathbf{x}) + \delta\Psi_b^{(k)} \\ \delta\Psi_b^{(k)} &\triangleq \oint_{\partial\Omega_{comp}} \gamma_{comp}(\mathbf{x}')\; G(\mathbf{x}-\mathbf{x}')\; d\mathbf{x}' \\ &\quad - \oint_{\partial\Omega_b} (\gamma^{(k)} - \gamma_0)(\mathbf{x}')\; G(\mathbf{x}-\mathbf{x}')\; d\mathbf{x}' \; . \end{aligned} \tag{4.6}$$

In Section 4.3.3, we explain how to perform this iteration and how to evaluate the contribution of $(\gamma^{(k)} - \gamma_0)$ in the framework of the immersed interface approach. Briefly, we need to evaluate the convolutions in $\delta\Psi_b^{(k)}$ (see Eq. (4.6)) on the outer boundary $\partial\Omega_{comp}$, based on Ψ_0 and the previous solution $\Psi_b^{(k)}$. Once the outer boundary condition is known ($\Psi_b^{(k+1)} = \delta\Psi_b^{(k)}$ on $\partial\Omega_{comp}$), we can compute the next solution $\Psi_b^{(k+1)}(\mathbf{x})$ inside Ω_{comp}. Note that the convolutions from Eq. (4.6) are not explicitly evaluated inside Ω_{comp}, as the following equivalent Poisson problem is solved instead, using the tools from Section 4.3.1:

$$\begin{aligned} \nabla^2\Psi_b^{(k+1)} &= -\omega - \oint_{\partial\Omega_b} \gamma^{(k+1)}(\mathbf{x}')\; \delta(\mathbf{x}-\mathbf{x}')\; d\mathbf{x}' \quad \text{in} \;\; \Omega_{comp} \\ \Psi_b^{(k+1)} &= \overline{\Psi}^{(k+1)} - \Psi_\infty \;\; \text{on} \;\; \partial\Omega_b \;\; \text{such that} \;\; \oint_{\partial\Omega_b} \gamma^{(k+1)}(\mathbf{x}')\; d\mathbf{x}' = \Gamma \\ \Psi_b^{(k+1)} &= \delta\Psi_b^{(k)} \quad \text{on} \;\; \partial\Omega_{comp} \; . \end{aligned} \tag{4.7}$$

Again, one should observe here that it is not required to know $\gamma^{(k+1)}$ (nor $\overline{\Psi}^{(k+1)}$) in order to solve Eq. (4.7), as it is obtained as a part of the solution, along with $\Psi_b^{(k+1)}$ (imposing the outer boundary condition, the circulation and the vorticity is sufficient).

As a side note, the current approach enforcing the circulation amounts to impose the value of the monopole, leaving the iteration on the higher order multipole terms. No convergence break down was observed, even when the outer boundary of the computational domain is close to the body. This is in contrast with the under relaxation required when trying to impose $\overline{\Psi}$, as referred in [91].

4.3.3 Algorithm

Consider a uniform 2-D node-centered grid defined over the rectangular domain $[x_0, x_N] \times [y_0, y_M]$ with $N+1$ points in the x direction and $M+1$

points in the y direction. Thus $\Delta x = (x_N - x_0)/N$ and $\Delta y = (y_M - y_0)/M$. The one-dimensional explicit stencil for the second derivative at x is then $\{R^x_{-1}, R^x_0, R^x_{+1}\} = \{1, -2, 1\}/(\Delta x)^2$; the stencil for y is similar. The algorithm to solve the potential flow problem Eq. (4.1) is following [91], taking into account the modifications introduced in Section 4.3.2. There are essentially three types of operations involved in this algorithm (see also Fig. 4.3):

- $\Psi = solve(\Psi_{comp}, -\omega, \Gamma)$: The immersed interface tools of Section 4.3.1 are used to obtain the discrete solution of $\nabla^2\Psi = -\omega$ with prescribed circulation Γ and satisfying the outer boundary conditions $\Psi(\mathbf{x}) = \Psi_{comp}(\mathbf{x})$ on $\partial\Omega_{comp}$.

- $\gamma_{bdy} = extractCharge(\Psi)$: This operation extracts a boundary charge γ_{bdy} from a discrete field Ψ.

- $\Psi_{comp} = conv(\gamma_{bdy})$: This operation computes a new outer boundary condition on $\partial\Omega_{comp}$ by means of a convolution between the free space Green's function and the boundary charge γ_{bdy} (see Eq. (4.6)).

The algorithm is then performed as follows:

1. **Homogeneous Dirichlet solution** : Solve the problem Eq. (4.5) for Ψ_0 with homogeneous outer Dirichlet boundary conditions on $\partial\Omega_{comp}$:

   $$\Psi_0 = solve(0, -\omega, \Gamma)$$

2. **Outer sheet** : Compute the artificial source $\gamma_{comp} = \partial\Psi_0/\partial n$ on $\partial\Omega_{comp}$. This is done using a one-sided fourth order finite difference stencil, as in [90] ('$FD4$' in Fig. 4.3) :

   $$\gamma_{comp} = extractCharge(\Psi_0)$$

3. **Initial BC correction** : This is the first step of the iterative approach ($k = 0$). The vortex sheet γ_{comp} induces a new outer boundary condition and, as can be seen in Eq. (4.6), it is given by

   $$\forall \mathbf{x} \in \partial\Omega_{comp} : \delta\Psi_b^{(0)}(\mathbf{x}) \triangleq \oint_{\partial\Omega_{comp}} \gamma_{comp}(\mathbf{x}')\, G(\mathbf{x} - \mathbf{x}')\, d\mathbf{x}' \ .$$

   According to the above observation made about γ_{comp}, this step computes the outer boundary condition induced by the far-field contributions from ω and γ_0. One should also observe that the second convolution from

Eq. (4.6) cannot yet be evaluated, as only γ_0 is available at this point. The corresponding algorithmic operation is then

$$\delta\Psi_b^{(0)} = conv(\gamma_{comp})$$

Uniform intensity panels in conjunction with fast multipole summation techniques are used for the computation of the convolution ('*panels*' in Fig. 4.3). The panels are centered on the grid nodes, i.e. for the upper boundary $y = y_M$ at the grid node x_i we have a panel defined between $x_i - \Delta x/2$ and $x_i + \Delta x/2$ with an intensity given by $(\partial\Psi_0/\partial n)_{i,M}$.

We now start the iteration with $k = 1$.

4. **Non-homogeneous Dirichlet solution** : Eq. (4.7) is now solved with the outer boundary condition based on the vortex sheet extracted from the previous solution. The algorithmic operation in Fig. 4.3 is

$$\Psi_b^{(k)} = solve(\delta\Psi_b^{(k-1)}, -\omega, \Gamma)$$

5. **Inner sheet** : A new vortex sheet $(\gamma^{(k)} - \gamma_0)$ is generated on $\partial\Omega_b$ and, following the observation made in Eq. (4.3), it is evaluated by using the equivalent bulk field $\omega_{\gamma^{(k)}-\gamma_0}$ extracted from $\Psi_b^{(k)} - \Psi_0$ ('*bulk sheet*' in Fig. 4.3). It can be computed efficiently as it only involves the irregular points of the domain

$$\omega_{\gamma^{(k)}-\gamma_0} = extractCharge(\Psi_b^{(k)} - \Psi_0)$$

6. **BC correction** : The new boundary charge on $\partial\Omega_b$ requires an update of the outer boundary condition and the convolution in Eq. (4.6) is computed using fast multipole summation techniques with singular particles to represent the bulk field $\omega_{\gamma^{(k)}-\gamma_0}$ ('*particles*' in Fig. 4.3). We thus evaluate the following equation on $\partial\Omega_{comp}$:

$$\delta\Psi_b^{(k)}(\mathbf{x}) = \delta\Psi_b^{(0)}(\mathbf{x}) - \oint_{\partial\Omega_b} (\gamma^{(k)} - \gamma_0)(\mathbf{x}')\, G(\mathbf{x} - \mathbf{x}')\, d\mathbf{x}' \,.$$

The operation in Fig. 4.3 is

$$\delta\Psi_b^{(k)} = \delta\Psi_b^{(0)} - conv(\omega_{\gamma^{(k)}-\gamma_0})$$

7. **Convergence check** : When the error $\|\Psi_b^{(k)} - \Psi_b^{(k-1)}\|_2$ is lower than a prescribed tolerance, the procedure stops ('*convergence*' in Fig. 4.3), else $k := k + 1$ and we go back to step 4.

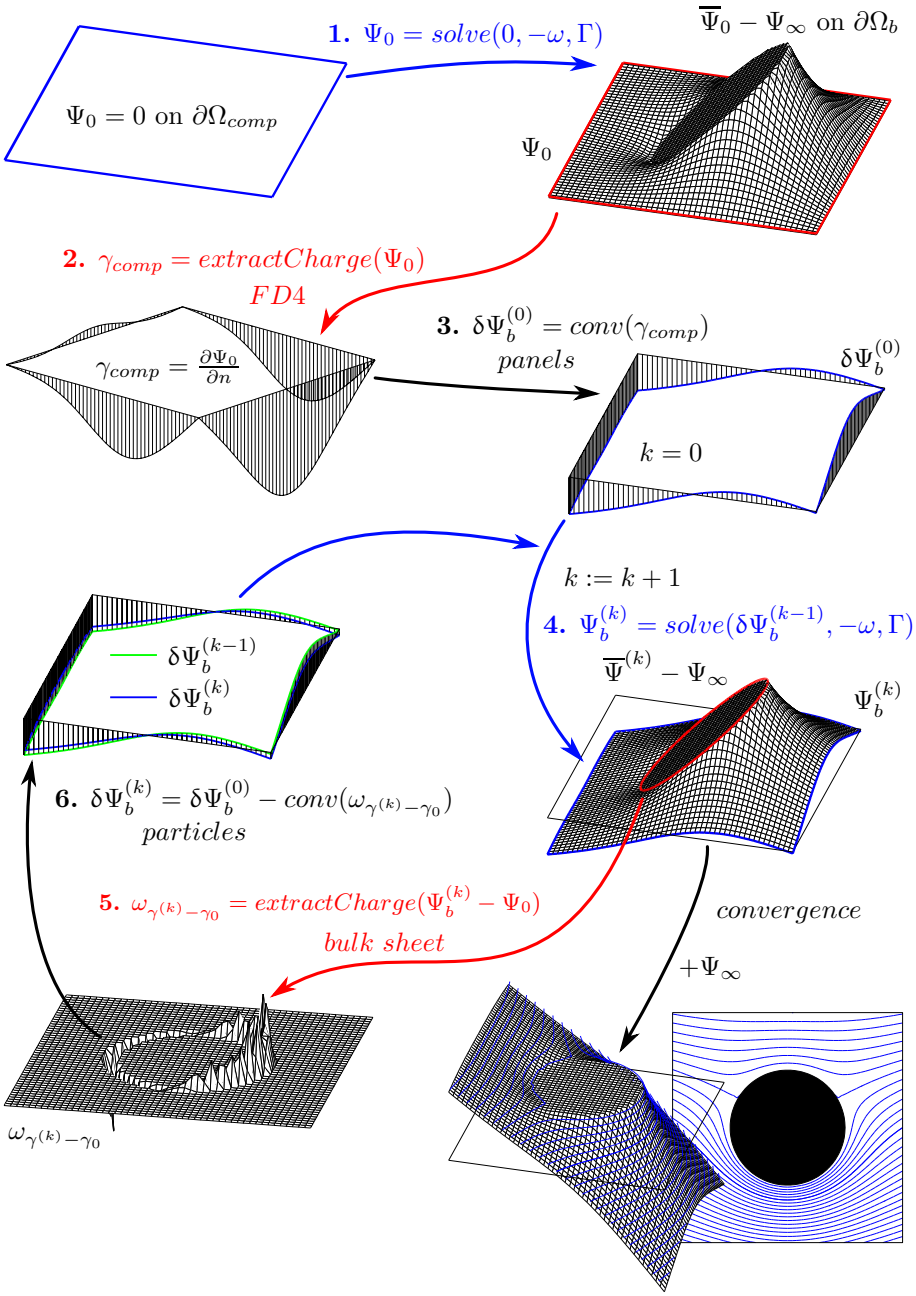


Figure 4.3: *Sketch of the different computational steps of the algorithm.*

4.3.4 Possible extension to 3-D problems

In the present section, we discuss the modifications required for an extension to three dimensions. We consider a simply connected domain (i.e. without holes). Hence the potential flow solution does not have circulation. We can still use the Helmholtz decomposition for the velocity field

$$\mathbf{u} = \nabla \times \mathbf{\Psi} - \nabla\phi \; .$$

The first term is used to account for the contribution of free vorticity $\boldsymbol{\omega}$ within the flow, if there is any; it is obtained by solving

$$\nabla^2 \mathbf{\Psi} = -\boldsymbol{\omega} \; ,$$

with unbounded flow condition on the outer boundary of the computational domain. The second term is used to enforce the no-through flow condition at the wall; it is obtained by solving

$$\nabla^2 \phi = 0$$

with the boundary condition

$$\mathbf{u} \cdot \mathbf{n} = (\nabla \times \mathbf{\Psi}) \cdot \mathbf{n} - \frac{\partial \phi}{\partial n} = \mathbf{u}_{wall} \cdot \mathbf{n} \; .$$

This, in fact, is the procedure that was followed by [104]. The present methodology could be used to solve the second problem. This scalar equation for ϕ is the counterpart of the scalar problem we solved in 2-D (for Ψ). However, imposing a Neumann boundary condition differs from the present 2-D approach enforcing a Dirichlet condition. The way to deal with this issue is explained in Chapter 5, where the aim is to impose a flux condition for a diffusion equation. Since we use 1-D correction stencils for the immersed interface part of the method, the extension to 3-D would be otherwise relatively simple.

4.4 Numerical results

The present approach is validated for 2-D potential flows. In the framework of sharp-interface capturing methods, a similar study has already been performed in [12, 11], where the unsteady 3-D inviscid flow around self-propelled fish-like

bodies was considered, using the HCIB method in velocity-pressure formulation from [54]. This methodology provides a specific treatment for inviscid flows, handling the no-through flow condition using an extrapolation of the tangential velocity.

First, we present the results for the potential flow past a cylinder, for which the analytical solution is known. A convergence study is performed. The method is also compared to the vortex panel method. Then, the added mass coefficient of an elliptical cylinder is computed and the order of convergence of the error is assessed for different aspect ratios of the cylinder. The flow past a cusped airfoil is also considered and we develop an implementation of the *Kutta-Joukowsky condition* that is consistent with the present framework. Finally, the method is illustrated and validated in the case of the flow past multiple bodies. For the remainder of the text we call Ψ_e the exact solution of the problem.

4.4.1 Grid convergence study for the potential flow with circulation past a cylinder

The streamfunction Ψ_e for the potential flow with circulation Γ past a cylinder of radius R (diameter D) is given by

$$\Psi_e(r,\theta) = U_\infty \left(1 - \frac{R^2}{r^2}\right) r\sin(\theta-\alpha) - \frac{\Gamma}{2\pi}\log\left(\frac{r}{R}\right) \ ,$$

where U_∞ is the free stream velocity magnitude with an angle of attack α with respect to the x-axis. We set here $\Gamma = 0.5\ (4\pi U_\infty R)$ and $\alpha = 0^\circ$ (see Fig. 4.4). The solution is computed on the domain $[-D;D]\times[-D;D]$, using a grid $(N+1)\times(N+1)$, thus defining a mesh size $h \triangleq \Delta x = \Delta y = 2D/N$. As the streamfunction is defined up to a constant, we compute the error field $\epsilon \triangleq (\Psi - \overline{\Psi}) - \Psi_e$. We consider the norms $\epsilon_\infty \triangleq \|\epsilon\|_\infty \triangleq \max_{\Omega_{comp}} |\epsilon|$ and $\epsilon_2 \triangleq \|\epsilon\|_2 \triangleq (1/D^2 \int_{\Omega_{comp}} \epsilon^2 \ d\mathbf{x})^{1/2}$. For two different mesh sizes h_{n-1} and h_n, the observed convergence order for a given norm $\|\cdot\|_q$ is then $(r_q)_n \triangleq \log((\epsilon_q)_n/(\epsilon_q)_{n-1})/\log(N_{n-1}/N_n)$. Results for the convergence study are given in Table 4.1. It can be seen that the accuracy of the method is asymptotically second order in both norms.

In order to allow a comparison with a vortex panel method, we use panels of approximate length h with uniform and linear intensity discretization, which are respectively first and second order methods. For panels with uniform inten-

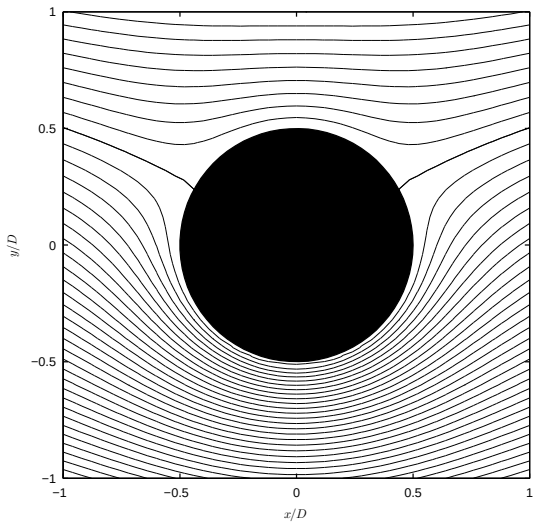


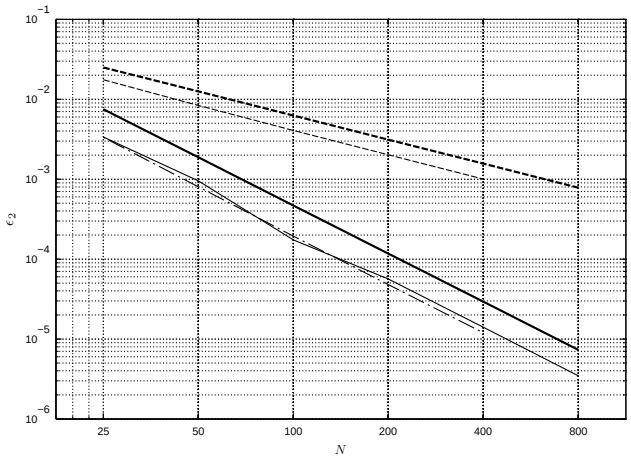
Figure 4.4: *Computed streamlines for the flow past a cylinder with* $\Gamma = 0.5\ (4\pi U_\infty R)$*, using* $N = 50$*. We show, on purpose, a case with low numerical resolution. Yet, it is seen that the numerical solution is already quite good.*

n	N	ϵ_∞	r_∞	ϵ_2	r_2
1	25	4.196586×10^{-3}		3.367888×10^{-3}	
2	50	1.309620×10^{-3}	1.68	9.520453×10^{-4}	1.82
3	100	3.471269×10^{-4}	1.92	1.740897×10^{-4}	2.45
4	200	9.194994×10^{-5}	1.92	5.641887×10^{-5}	1.63
5	400	2.380289×10^{-5}	1.95	1.402193×10^{-5}	2.01
6	800	6.161394×10^{-6}	1.95	3.464592×10^{-6}	2.02

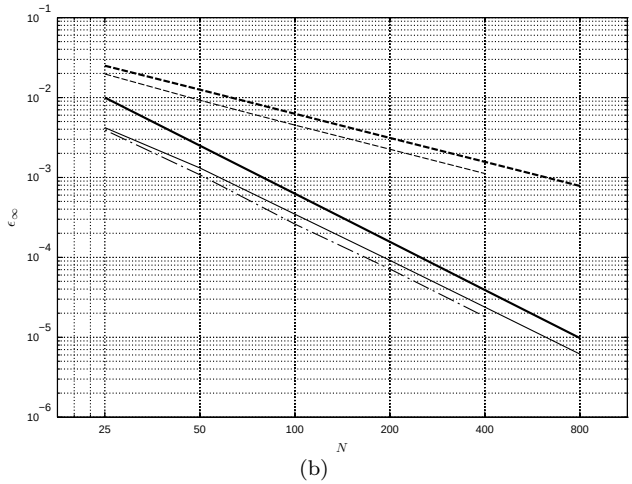
Table 4.1: *Grid convergence study for the potential flow around a cylinder on a grid* $N \times N$ *using the present approach.*

sity, we enforce zero tangential velocity in the center and just below the panel. For panels with linearly varying intensity, we enforce zero normal velocity in the center of each panel. In addition, we supply the constraint about the global circulation. The panel solution provides the spatial evolution of the intensity along the panels and, as a post processing step, the associated solution field is computed on the same grid as was previously defined for the present method. Again the error is computed by subtracting from the solution the stream function averaged over all panels. Grid convergence results for both uniform and

linear panels, as well as for the present method, are given in Fig. 4.5(a) (ϵ_2) and in Fig. 4.5(b) (ϵ_∞). The level of the error for the present approach is found to be similar to that obtained using linear panels. We stress again that the advantage here is that the representation of the boundary is consistent with the underlying finite difference stencil.



(a)



(b)

Figure 4.5: *Grid convergence study on ϵ_2 (a) and ϵ_∞ (b) for the potential flow with circulation past a cylinder: present approach (thin solid), uniform panels (thin dashed), linear panels (thin dash-dotted), first order slope (thick dashed) and second order slope (thick solid).*

4.4.2 Added mass for an elliptical cylinder without circulation

Another interesting point is the comparison of the numerically computed added mass with its theoretical value in the case of a non-lifting elliptical cylinder. According to [110], when $\Gamma = 0$, the linear impulse per unit length of the fluid can be expressed as

$$\begin{aligned} \mathbf{I}_f = \rho \int_{\Omega_f} \mathbf{u}\ d\mathbf{x} &= \rho \int_{\mathbb{R}^2} \mathbf{u}\ d\mathbf{x} - \rho \int_{\Omega_b} \mathbf{u}\ d\mathbf{x} \\ &= \rho \int_{\mathbb{R}^2} \mathbf{x} \times (\omega\ \hat{\mathbf{e}}_z)\ d\mathbf{x} - M\ \mathbf{U}_b\ , \end{aligned}$$

with ρ the fluid density, $M \triangleq \rho \int_{\Omega_b} d\mathbf{x}$ the mass per unit length of the "pseudo fluid inside the body" and $\mathbf{U}_b$ the body velocity. The last equation assumes that we work in a frame of reference that is stationary with respect to the fluid and that the pseudo fluid inside the body has a velocity equal to $\mathbf{U}_b$ ($\omega = 0$ inside Ω_b, as the body does not rotate here).

More precisely, for a cylinder that is initially at rest, we consider the increment of the fluid linear impulse per unit length $\delta\mathbf{I}_f$ due to an impulsive acceleration $\delta\mathbf{U}_b$ towards the left (here, the notation $\delta(\cdot)$ must not be confused with that used for the decomposition $\Psi_b \triangleq \Psi_0 + \delta\Psi_b$ from Section 4.3.2). The acceleration generates a vortex sheet increment ($\delta\gamma\ \hat{\mathbf{e}}_z$), which induces a velocity field $\delta\mathbf{u}$. We look for the so-called added mass per unit length M_a of the fluid next to the wall that virtually increases the mass M_b (per unit length) of the body during the acceleration. If $\mathbf{F}$ is the force per unit length accelerating the body and the surrounding fluid during a small time increment δt, we have, if $\delta\mathbf{I}_f$ is assumed to be aligned with $\delta\mathbf{U}_b$

$$\mathbf{F}\delta t = M_b\ \delta\mathbf{U}_b + \delta\mathbf{I}_f \triangleq (M_b + M_a)\ \delta\mathbf{U}_b\ .$$

And thus

$$M_a\ \delta\mathbf{U}_b = \delta\mathbf{I}_f = \rho \oint_{\partial\Omega_b} \mathbf{x} \times (\delta\gamma\ \hat{\mathbf{e}}_z)\ d\mathbf{x} - M\ \delta\mathbf{U}_b\ .$$

The added mass coefficient can then be computed by further assuming that $\delta\gamma\ \hat{\mathbf{e}}_z$ is proportional to $\delta\mathbf{U}_b$. For the numerical computation, we use again the bulk vorticity field increment ω_γ (see Eq. (4.3)) and we set $\delta\mathbf{U}_b = -\delta U_b\ \hat{\mathbf{e}}_x$ ($\alpha = 0°$). For an elliptical cylinder aligned with the reference axes and with

semi axis R_x and R_y, the theoretical added mass coefficient for an acceleration along the x-axis is $M_a^{th} = \rho\pi R_y^2$ [95].

The computational domain is $[-L;L] \times [-L;L]$ and the grid $(N+1) \times (N+1)$. The error $\epsilon_a \triangleq |M_a - M_a^{th}|$ and its convergence order $(r_a)_n \triangleq \log((\epsilon_a)_n/(\epsilon_a)_{n-1})/\log(N_{n-1}/N_n)$ are compared for different aspect ratios in Fig. 4.6 ($D_x/D_y = 1/4, 1/2, 1, 2, 4$). We can again observe a second order convergence for this diagnostic but the level of the error increases with decreasing ratio D_x/D_y. Two extrema appear in the vortex sheet at the top and at the bottom of the ellipse precisely where the local curvature increases. As expected, a decreasing ratio D_x/D_y makes it more challenging to capture the increasingly curved vortex sheet.

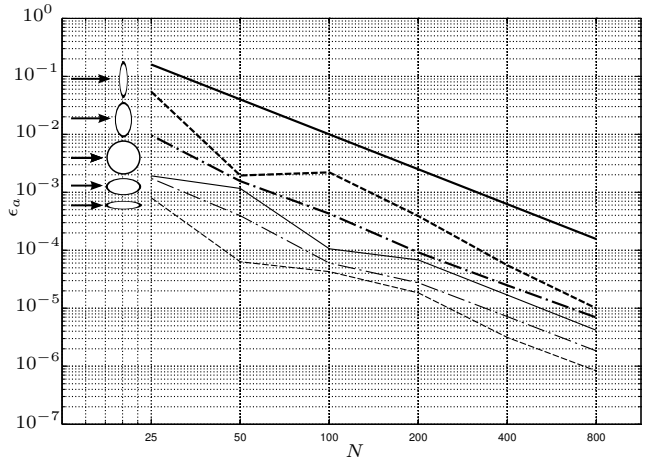


Figure 4.6: *Grid convergence study on the added mass error ϵ_a for an elliptical cylinder: $D_x/D_y = 1/4$ (thick dashed), $1/2$ (thick dash-dotted), 1 (thin solid), 2 (thin dash-dotted), 4 (thin dashed) and second order slope (thick solid).*

4.4.3 Grid convergence study for the potential flow with circulation past an airfoil

We consider here a symmetric Joukowsky airfoil. The conformal mapping that maps the circle of radius R (diameter D) in the complex plane $Z \triangleq X + iY$ onto the airfoil in the plane $z \triangleq x + iy$ is

$$z(Z) = Z + \frac{(R-\varepsilon)^2}{(Z-\varepsilon)} \, .$$

The streamfunction of the flow past the airfoil with an angle of attack α is $\Psi_e(x,y) = im\{F(Z(z))\}$, with

$$F(Z) = U_\infty \left(Ze^{-i\alpha} + \frac{R^2}{Z} e^{i\alpha} \right) + \frac{\Gamma}{2\pi i} \log\left(\frac{Z}{R}\right) \ .$$

The angle of attack is $\alpha = 10^\circ$. The domain is a square defined by $[-L;L] \times [-L;L]$ (with a computational grid $(N+1)\times(N+1)$). The chord is given by $c/L = 1.2$ and the thickness parameter by $\varepsilon/c = 0.035$. The radius R can be deduced from ε. The airfoil and the streamlines of the flow are displayed in Fig. 4.7.

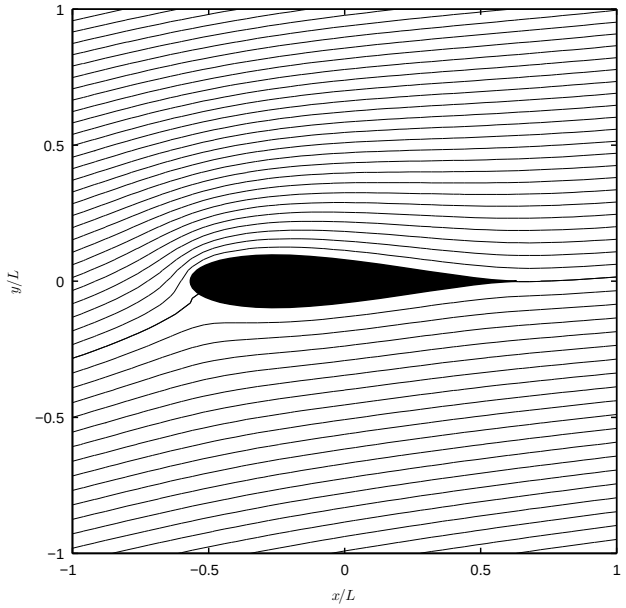


Figure 4.7: *Computed streamlines for the flow past an airfoil with $\alpha = 10^\circ$, using $N = 50$ and imposing numerically the Kutta-Joukowsky condition (with a 2-D stencil of order $p = 1$). The method is seen to perform already quite well despite the low numerical resolution.*

One important feature of the Joukowsky airfoil is the cusp at its trailing edge. Setting the circulation to $\Gamma_e = -4\pi U_\infty R \sin\alpha$ ensures that the velocity at the trailing edge remains finite (*Kutta-Joukowsky condition*). The velocity vector is then also aligned with the trailing edge. Any other choice of Γ leads to an infinite velocity at this point, or equivalently $|\partial\Psi/\partial n| \to \infty$ with $\mathbf{n}$ a vector aligned with the trailing edge. This observation also holds for trailing edges with a non-zero angle.

Numerically, as we do not know the circulation a priori, we have to replace Eq. (4.4) and write a new equation that eventually enforces the *Kutta-Joukowsky condition*:

$$\left.\frac{\partial\Psi}{\partial n}\right|_{TE}^{up} + \left.\frac{\partial\Psi}{\partial n}\right|_{TE}^{down} = 0 \,, \tag{4.8}$$

where the subscript TE refers to the trailing edge and where the superscripts *up* and *down* stand for an evaluation of the normal derivative on the two respective sides of the trailing edge (see Fig. 4.8; contrary to previously, the normal vectors are here defined to be pointing outward of Ω_b).

Eq. (4.8) means that the tangential velocity components above and below the trailing edge are equal and thus aligned with the trailing edge (note that this is also what is done in vortex panel methods). As the present approach is iterative (the linear solver and the boundary potential method are both iterative), the circulation changes at each iteration step and eventually adopts the correct value, after global convergence. Before reaching convergence, the intermediate solutions provide an infinite perpendicular velocity component at the trailing edge, yet the tangential velocity component remains regular on both sides of the trailing edge and Eq. (4.8) is valid. The circulation Γ satisfying Eq. (4.8) can be computed afterwards using Eq. (4.4).

Eq. (4.8) is discretized using two-dimensional finite difference stencils (for the two normal derivatives on both sides of the trailing edge) because $\mathbf{x}_{TE}$ does not necessarily lie on a grid segment or a grid node. The stencils for the first derivatives can be computed at an arbitrary order p by means of two-dimensional Taylor series. Therefore N_p different grid nodes ($N_p \triangleq \frac{1}{2}(p+1)(p+2)-1$) have to be chosen in addition to the value $\overline{\Psi}$ at the trailing edge point $\mathbf{x}_{TE}$ (this ensures that the envelope of the data locations is convex for the interpolation procedure). The coefficients of the stencil are then obtained as the solution of a linear system that is computed using Taylor series around $\mathbf{x}_{TE}$ and evaluated at the given grid node locations. The observation made previously about the singularity leads to choose grid nodes in the region just upstream of $\mathbf{x}_{TE}$, in order not to include the singularity. However, the choice of the N_p grid nodes is constrained by the fact that the coefficient matrix should not be rank deficient. For high values of p, choosing the nearest nodes to $\mathbf{x}_{TE}$ often leads to rank deficiency. The current procedure to compute a valid stencil consists in generating random sets of N_p grid nodes and checking the rank of the associated matrix (for a set of valid stencils, the one minimizing

the leading error term is chosen). Note that one-dimensional stencils could also have been used, by adopting a strategy similar to the one that will be developed in Chapter 5.

As an example, two of the resulting stencils for $p = 2$ and $p = 4$ are shown in Fig. 4.8. Fig. 4.7 shows the solution obtained using this procedure with $p = 1$ on a grid $N = 50$. Even if both the resolution and the order of the equation to enforce the *Kutta-Joukowsky condition* are quite low, one can observe that the computed streamlines are nevertheless already quite close to those of the analytical solution.

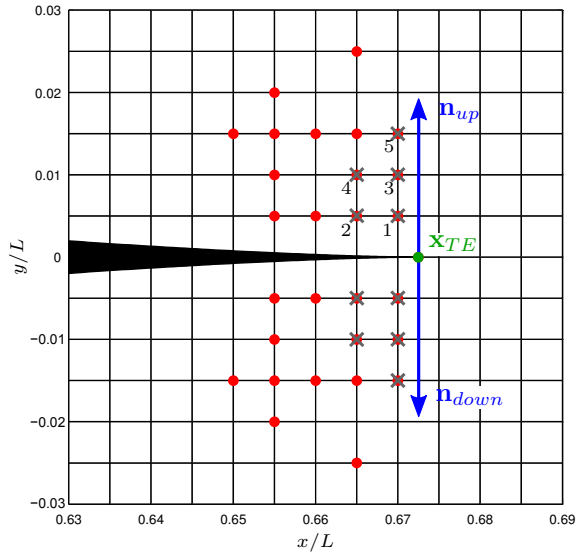
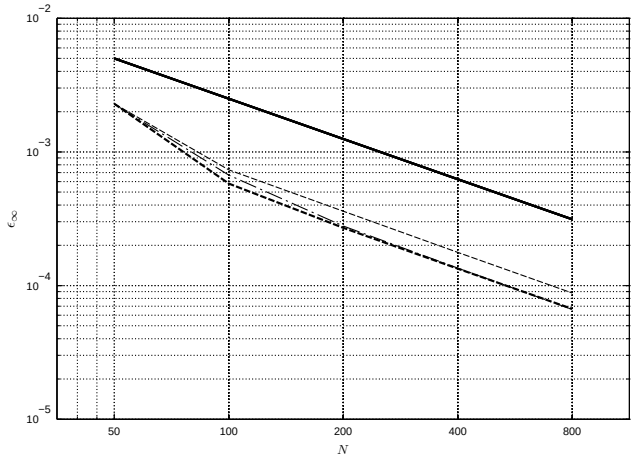


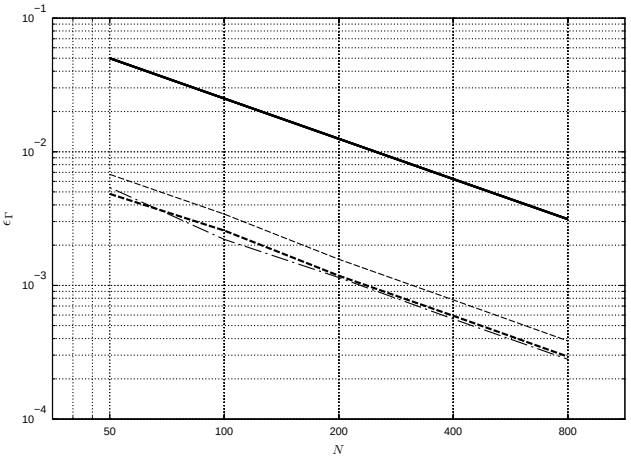
Figure 4.8: *Stencil used for the discretization of Eq. (4.8) (zoom on the trailing edge area). Grid nodes used for $p = 2$ (crosses) and for $p = 4$ (bullets). As an illustration, the $p = 2$ weights to compute the derivative $\frac{\partial}{\partial y}(\cdot)$ are $w_{1\ldots5} = \{0, -3, \frac{1}{2}, \frac{9}{2}, -\frac{1}{2}\}/\Delta y$ and $w_{TE} = -\sum_{k=1}^{5} w_k$.*

The ability of the method to capture this geometric feature, as well as to predict the correct circulation is tested here using a grid convergence study. The trailing edge is placed exactly on a grid line ($y = 0$) to ensure that irregular points are generated in that area. Moreover, for the computations at different resolutions N, we slightly translate the airfoil along the x-axis in order to keep $\mathbf{x}_{TE}$ equidistant from two consecutive grid nodes on the axis $y = 0$. This allows to keep exactly the same stencil for every resolution. Results are shown for ϵ_∞ in Fig. 4.9(a) and for $\epsilon_\Gamma \triangleq |\Gamma - \Gamma_e|$ in Fig. 4.9(b), also comparing different values for the order p of the stencil. Γ is simply computed as a post processing

step using Eq. (4.4). The accuracy of the method is here only first order for both errors, even for higher order stencils (increasing p does not improve the asymptotic convergence rate). This is because the immersed interface stencil corrections associated to the nodes next to the trailing edge are not computed accurately. Indeed, as long as the circulation is not equal to Γ_e, the gradient of the solution is infinite at $\mathbf{x}_{TE}$ and so is the jump in the first derivative, which is clearly not taken into account in the immersed interface approach. The trailing edge would need a special treatment in order to improve the convergence order. We leave this as a subject for further investigation.



(a)



(b)

Figure 4.9: *Grid convergence study on ϵ_∞ (a) and ϵ_Γ (b) for the potential flow with circulation past an airfoil: $p = 4$ (thick dashed), $p = 2$ (thin dashed), $p = 1$ (thin dash-dotted) and first order slope line (thick solid).*

4.4.4 Potential flow past multiple bodies

We now consider the flow with an angle of attack $\alpha = 10^\circ$ past three bodies defined on a domain $[-L; L] \times [-L; L]$, and a grid $(N+1) \times (N+1)$. The bodies are a pentagonal shape (body 1), a triangular shape (body 2) and an ellipse (body 3). Body 1 has a parametric polar representation with a radius given by $r(\theta) = R_{p,1}(1+0.035\sin(5(\theta-15^\circ)))$, θ being the angle defined around the point $(0.3L; 0.45L)$ and $R_{p,1}/L = 0.3$. Its circulation is $\Gamma_1 = 0.5$ $(4\pi U_\infty R_{p,1})$. The polar equation for body 2 is $r(\theta) = R_{p,2}(1+0.1\sin(3(\theta-50^\circ)))$, θ being the angle defined around the point $(0.4L; 0.4L)$ and $R_{p,2}/L = 0.2$ $(\Gamma_2 = -4\pi U_\infty R_{p,2})$. The ellipse is centered in $(-0.4L; -0.3L)$, it is tilted by an angle of 45° in the clockwise sense and has a semi-major axis $R_{maj}/L = 0.4$ and a semi-minor axis $R_{min}/L = 0.2$. The ellipse is non lifting $(\Gamma_3 = 0)$. Even if some parts of the body geometries are not convex, it is nevertheless guaranteed here that the one-sided correction stencils at the different irregular points do not cross the boundary. The streamlines of the potential flow are given in Fig. 4.10.

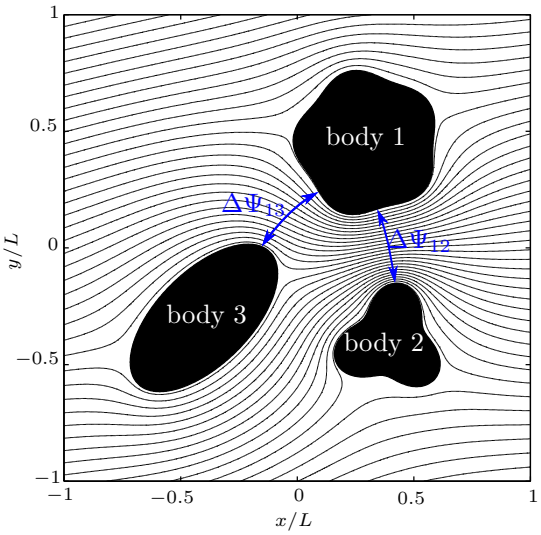


Figure 4.10: *Computed streamlines for the flow past multiple bodies, using* $N = 400$.

In order to validate the results, the mass flow rates between the bodies are computed: $\Delta\Psi_{12} \triangleq \overline{\Psi}_1 - \overline{\Psi}_2$ and $\Delta\Psi_{13} \triangleq \overline{\Psi}_1 - \overline{\Psi}_3$ ($\overline{\Psi}_m$ is the constant streamfunction value for body m, see also Fig. 4.10). These values are compared for different grid sizes N with the reference results obtained using a highly refined vortex panel method (approximately 4500 linear panels in total, the

equivalent of twice the mesh resolution of the finest grid used for the present approach computation). A mesh convergence study is performed in Fig. 4.11 and one can observe again a second order convergence for the errors $\epsilon_{f,12} \triangleq |\Delta\Psi_{12} - \Delta\Psi_{12}^{ref}|$ and $\epsilon_{f,13} \triangleq |\Delta\Psi_{13} - \Delta\Psi_{13}^{ref}|$.

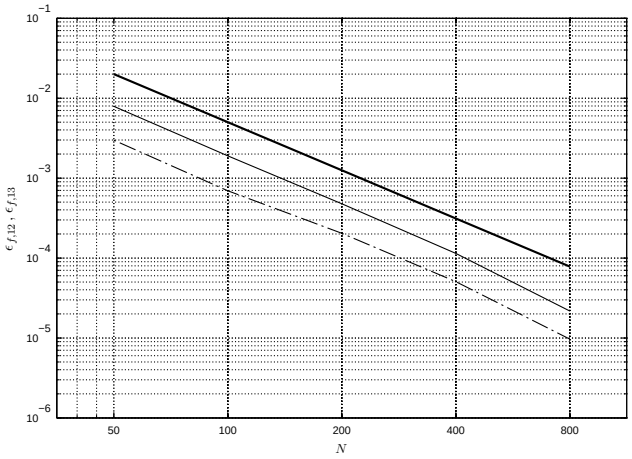


Figure 4.11: *Grid convergence study on mass flow rates for the potential flow with multiple bodies: $\epsilon_{f,12}$ (thin solid), $\epsilon_{f,13}$ (thin dashed) and second order slope (thick solid).*

4.5 Conclusion

A two-dimensional second order finite difference solver has been presented for Poisson equations in unbounded domain and including irregular boundaries. The method has been developed with the specific goal of computing the potential flow around multiple bodies with compact vorticity support. A grid convergence study has been performed based on the analytical solution of the potential flow past a cylinder. This study confirms the claimed accuracy. The prediction of the added mass coefficient of an elliptical cylinder leads to the same conclusion concerning the order of convergence. A comparison with the vortex panel method has also been performed (uniform and linear panels) and the observed level of the error norms is found to be similar to those obtained with linear panels. The method has also been validated for the flow past multiple bodies. The previously mentioned cases allow the imposition of an arbitrary circulation for the flow around each body. For the flow past cusped airfoils, the circulation that satisfies the *Kutta-Joukowsky condition* is unique. There-

fore, a way to enforce the *Kutta-Joukowsky condition* has been developed, that leads, at convergence, to the proper circulation. However, the global method is then only first order accurate, because the intermediate steps do not have the proper circulation. Finally, the procedure for a possible extension to solving 3-D potential flow problems was also given.

The aim of this work is to integrate this type of solver inside a viscous vortex particle-mesh solver, as will be seen in Chapter 7. In this method, one step consists in solving a Poisson equation in order to recover, from a given vorticity field, the unbounded velocity field that respects a no-through flow condition at the wall. One way to do this is to combine a multipole-based vortex panel solver with a fast finite difference Poisson solver, as done in [83]. One of the drawbacks is then the lack of full consistency between the vortex panel solver and the finite difference discretization. The present approach treats the wall condition in a fashion that is more consistent with the finite difference stencil. Furthermore, the solution would then no longer be a combination of two different approaches, as everything would be computed on the grid. The accurate evaluation of the vorticity flux emanating from the wall and its diffusion into the surrounding fluid (see [102]), all also done on the grid, is explained in Chapter 5. The required modifications for the underlying particle-mesh interpolation are given in Chapter 6 and the immersed interface-enabled vortex particle-mesh solver is detailed and validated in Chapter 7.

Chapter 5

Development of an immersed interface parabolic solver

In the immersed interface method developed in this thesis, the diffusive term of the Navier-Stokes equations is also computed on the grid, using jump-corrected finite differences, similarly to what was done for the Poisson solver. The direction-splitting approach we have used so far easily takes into account Dirichlet-type boundary conditions (see Chapter 4). The methodology has to be extended to allow for Neumann boundary conditions, so as to be able to apply a no slip boundary condition at the wall. Indeed, this condition requires the enforcement of a flux, which is proportional to the remaining tangential slip velocity after the convective part of the time-stepping (see Chapter 3).

The aim of this chapter is first to develop a stencil for Neumann boundary conditions, and secondly, to study its stability properties in the framework of the diffusion equation, in one and two dimensions.

5.1 One-dimensional case

The diffusion equation for $\omega(x,t)$ in one dimension and defined in the domain $x \in [a,b]$ reads

$$\frac{\partial \omega}{\partial t} = \nu \frac{\partial^2 \omega}{\partial x^2} \ ,$$

with prescribed boundary conditions at the two boundaries a and b.

The discretization of the second derivative in space is performed using second order centered finite differences. The boundary conditions are easily enforced if the underlying grid is defined in a way that the boundaries a and b coincide with grid nodes. However, if the latter condition is not fulfilled, an immersed interface stencil has to be provided and the tools developed in Section 4.3.1 have to be reused.

Let us assume that the grid is defined as $x_k = x_0 + k \cdot h$ $(k = 0, ..., N)$ and $x_N = b$ (the right boundary coincides with the last grid node), whereas the left boundary $a \in (x_i, x_{i+1}]$ is immersed in the grid $(x_0 < a)$, as shown in Fig. 5.1. For $x < a$, we simply set $\omega(x,t) = 0$, $\forall t$ (this is equivalent to solving the diffusion equation in the domain $[x_0, a)$ with homogeneous Dirichlet boundary conditions). Without loss of generality, we also set the right boundary condition to $\omega(b,t) = \omega_b(t)$. A Dirichlet-type boundary condition at $x = a$ will be expressed as $\omega(a,t) = \omega_a^{(0)}(t)$ and a Neumann-type BC as $-\nu\, \omega^{(1)}(a,t) = q_a(t)$, or equivalently as $\omega_a^{(1)}(t) \triangleq -q_a(t)/\nu$. In order to make the link with the target problem in fluid dynamics we wish to solve, the domain $x < a$ represents the body, where the vorticity ω is supposed to be zero (if the body is at rest) and $x \geq a$ represents the flow domain. The equation is solved in the whole domain $[x_0, x_N = b]$ (flow and body), just as was the case for the Poisson solver.

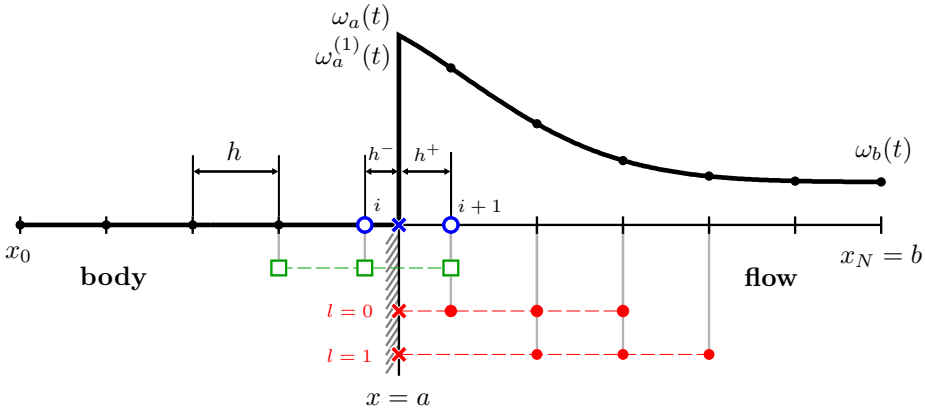


Figure 5.1: *Sketch of the 1D immersed interface problem : position of the immersed interface ($x = a$, blue cross); irregular points i and $i+1$ affected by the correction term (blue circles); stencil for the second derivative at x_i that crosses the interface (green squares); two possible stencils for the correction terms ($l = 0$ or $l = 1$, red bullets).*

Knowing that the solution is discontinuous at $x = a$, we choose the following jump-corrected scheme for the discretization of the second derivative at x_i

$$\begin{aligned} \omega_i^{(2)} &= \frac{1}{h^2}\left(\omega_{i-1} - 2\,\omega_i + (\omega_{i+1} - J_a^+)\right) + \mathcal{O}(h^2) \\ J_a^+ &= [\omega^{(0)}]_a + \frac{h^+}{1!}[\omega^{(1)}]_a + \frac{(h^+)^2}{2!}[\omega^{(2)}]_a + \frac{(h^+)^3}{3!}[\omega^{(3)}]_a \ , \end{aligned} \tag{5.1}$$

with $\omega_i^{(k)} \triangleq \omega^{(k)}(x_i)$, $\omega_i = \omega_i^{(0)} \triangleq \omega(x_i)$ and $h^+ \triangleq x_{i+1} - a$. For the sake of clarity, the notation for the time variable t is omitted here. The second derivative at x_{i+1} is handled similarly using $h^- \triangleq a - x_i$ (instead of h^+) at J_a^-, as discussed in Chapter 4 and [80]. The terms J_a^+ and J_a^- provide us a mean to enforce the boundary condition at $x = a$. The derivative jumps can be expressed as

$$[\omega^{(k)}]_a = \omega^{(k)}(a^+) - \omega^{(k)}(a^-) = \omega^{(k)}(a^+) \ ,$$

since $\omega^{(k)}(a^-) = 0$ (the solution inside the body is zero). If the body was on the right hand side ($x > a$), we would have had $[\omega^{(k)}]_a = -\omega^{(k)}(a^-)$. The derivatives of ω are computed using one-sided finite differences, only taking into account the solution on the flow side ($x \geq a$) and the boundary condition. We thus have

$$\omega^{(k)}(a^+) = (S_q^k)_a\ \omega_a^{(q)} + \sum_{p=1}^{3}(S_q^k)_p\ \omega_{i+p+l} + \mathcal{O}(h^{4-k}) \qquad \forall k = 0, ..., 3 \ ,$$

with $q = 0$ (Dirichlet) or $q = 1$ (Neumann) and l is a shift parameter (see Fig. 5.1). When $k = q$, the coefficients simplify to $(S_q^q)_a = 1$ and $(S_q^q)_p = 0$, by definition. For a more detailed discussion about this type of scheme, including their computation and the associated leading error term analysis, refer to Appendix A.

The time derivative is handled using a classical ODE time integration scheme; e.g., when using the first order Euler scheme (= Runge-Kutta scheme of first order)

$$\omega_i^{n+1} = \omega_i^n + \nu\Delta t\left(\omega_i^{(2)}\right)^n \ . \tag{5.2}$$

The classical stability analysis based on Fourier modes (e.g., Von Neumann stability analysis) can be applied when the grid is uniform and the solution

periodic. Moreover, the finite difference discretization scheme must remain identical for all grid points. The stability criterion for the discretized Eq. (5.2) without interface is a well-known result:

$$r \triangleq \frac{\nu \Delta t}{h^2} \leq \frac{1}{2} ,$$

with r the Fourier number. The presence of the interface invalidates this approach, as the stencil is modified near the interface. As a consequence, we have to consider different tools. In the following analysis, we will focus on the single step Runge-Kutta schemes of order p.

The matrix stability analysis is based on the study of the whole discretized linear system, including the boundary conditions. Performing first the space discretization of Eq. (5.1), we obtain the following linear ordinary differential equation (the time is still continuous)

$$\frac{d\mathbf{U}}{dt} = \nu \left(\frac{\mathbf{B}}{h^2}\right) \mathbf{U} ,$$

with $\mathbf{B}/h^2$ the discretized second derivative in space, including the correction terms, and $\mathbf{U}(t)$ the vector of the unknowns $\omega_i(t)$ and of the prescribed boundary conditions at x_0, a and $b = x_N$. For this analysis, the boundary conditions are assumed to be constant in time.

The time integration can then be written in the form of an amplification matrix $\mathbf{A}$:

$$\mathbf{U}^n = \mathbf{A}\mathbf{U}^{n-1} = \cdots = \mathbf{A}^n \mathbf{U}^0 ,$$

where $\mathbf{U}^n$ is the solution vector at time t^n.

An exact time integration over one time step Δt is given by

$$\mathbf{A} = \exp(r\mathbf{B}) \triangleq \lim_{N \to \infty} \sum_{p=0}^{N} \frac{1}{p!} (r\mathbf{B})^p ,$$

with $\mathbf{B}^0 = \mathbf{I}$ the identity matrix. The p^{th} order Runge-Kutta scheme is simply a truncation of the aforementioned exponential

$$\mathbf{A} = \mathbf{I} + (r\mathbf{B}) + \frac{1}{2}(r\mathbf{B})^2 + ... + \frac{1}{p!}(r\mathbf{B})^p .$$

The approximation is defined to be Lax-stable for $\tau > 0$ ($\Delta t \leq \tau$ and $0 \leq n\Delta t \leq T$), if the following condition holds

$$\forall n \ : \ \|\mathbf{A}^n\| \leq 1 \ .$$

A necessary condition for this is

$$\rho(\mathbf{A}) \triangleq \max_k |\lambda_k(\mathbf{A})| \leq 1 \ , \tag{5.3}$$

with $\rho(\mathbf{A})$ the spectral radius of the matrix $\mathbf{A}$ and $\lambda_k(\mathbf{A})$ its eigenvalues. This condition is sufficient if the matrix $\mathbf{A}$ is normal ($\mathbf{A}\ \mathbf{A}^T = \mathbf{A}^T\mathbf{A}$).

The eigenvalues of $\mathbf{A}$ can be linked to those of $\mathbf{B} = \mathbf{V}\mathbf{\Lambda}_B\mathbf{V}^{-1}$ (we assume $\mathbf{B}$ is diagonalizable), where $\mathbf{\Lambda}_B$ is the diagonal matrix of the eigenvalues $\lambda_k(\mathbf{B})$. Indeed, we can write

$$\begin{aligned}
\mathbf{A} &= \mathbf{V}\,\mathbf{I}\,\mathbf{V}^{-1} + r\,\mathbf{V}\mathbf{\Lambda}_B\mathbf{V}^{-1} + \frac{r^2}{2}\,\mathbf{V}\mathbf{\Lambda}_B^2\mathbf{V}^{-1} + ... + \frac{r^p}{p!}\,\mathbf{V}\mathbf{\Lambda}_B^p\mathbf{V}^{-1} \\
&= \mathbf{V}\left[\mathbf{I} + r\,\mathbf{\Lambda}_B + \frac{r^2}{2}\,\mathbf{\Lambda}_B^2 + ... + \frac{r^p}{p!}\,\mathbf{\Lambda}_B^p\right]\mathbf{V}^{-1} \\
&= \mathbf{V}\mathbf{\Lambda}_A\mathbf{V}^{-1} \ .
\end{aligned}$$

Hence, the eigenvalues of $\mathbf{A}$ satisfy

$$\begin{aligned}
\lambda_k(\mathbf{A}) &= \phi\big(r\ \lambda_k(\mathbf{B})\big) \\
\phi(z) &\triangleq 1 + z + \frac{1}{2}\,z^2 + ... + \frac{1}{p!}\,z^p \ .
\end{aligned}$$

The necessary condition for stability is then

$$r \leq r_{\max} \triangleq \sup_{r^* \in \mathscr{D}} r^* \ , \tag{5.4}$$

with $\mathscr{D} \triangleq \{r > 0 \ \text{ s.t. } \ \rho(\mathbf{A}) \leq 1\} = \{r > 0 \ \text{ s.t. } \ \forall k \ : \ r\ \lambda_k(\mathbf{B}) \in \mathscr{S}\}$ and $\mathscr{S} \triangleq \{z \in \mathbb{C} \ \text{ s.t. } \ |\phi(z)| \leq 1\}$ the region of stability associated to the time integration scheme (here the Runge-Kutta scheme of order p).

Note that for any function $\phi(z)$ defined on the spectrum of $\mathbf{B}$ (i.e. $\phi(\lambda_k(\mathbf{B}))$ exists for all eigenvalues of $\mathbf{B}$, assuming again $\mathbf{B}$ is diagonalizable), the eigenvalues of $\phi(\mathbf{B})$ are simply $\phi(\lambda_k(\mathbf{B}))$, see [59]. The analysis is hence applicable to other time integration schemes, such as for example implicit Runge-Kutta methods. In particular, $\phi(z)$ is a rational function for Lobatto IIIB methods

(e.g. the second order scheme leads to $\phi(z) = (1 + z/2)/(1 - z/2)$). The latter methods are unconditionally stable thanks to their implicit nature and they thus do not require a further analysis ($|\phi(z)| \leq 1$ for $z \in \mathbb{C}$ with $\Re\{z\} \leq 0$).

The uncorrected centered finite difference scheme provides real eigenvalues $\lambda_k(\mathbf{B})$. However, the addition of the correction terms due to the presence of the interface may generate complex eigenvalues $\lambda_k(\mathbf{B})$, and the stability criterion for the explicit Euler scheme ($p = 1$) then simplifies as follows

$$r \leq r_{\max} \triangleq \min_k \left[-\frac{2\,\Re\{\lambda_k(\mathbf{B})\}}{|\lambda_k(\mathbf{B})|^2} \right] .$$

As an example, Fig. 5.2 shows the position of $r_{\max}\ \lambda_k(\mathbf{B})$ for all possible h^+/h together with the Euler stability region (here $N = 50$).

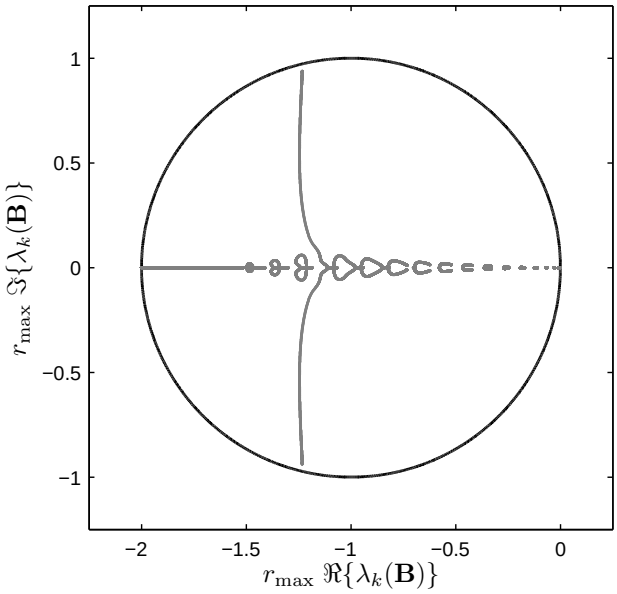


Figure 5.2: *Position of $r_{\max}\ \lambda_k(\mathbf{B})$ for all possible h^+/h together with the Euler stability region ($N = 50$). The immersed BC is Dirichlet with $l = 1$.*

The stability criteria as a function of h^+/h, for $N = 200$, are shown in Fig. 5.3. These criteria are obtained using a bisection method on $\rho(\mathbf{A}) - 1$, which is a continuous function of r (but not differentiable) for all $\lambda_k(\mathbf{B})$ such that $\Re\{\lambda_k(\mathbf{B})\} < 0$. The domain is defined as $x_0 = -L + h^+$ and $b = x_N = L + h^+$ and the interface is at $a = 0$ (N is even). Fig. 5.3 compares the stability criteria of different Runge-Kutta integration schemes ($p = 1, 2, 3$ and 4) with

a Dirichlet BC and a Neumann BC at $x = a$. It also shows the influence of the stencil shifting parameter l used in the one-sided stencil to compute the different derivatives of the solution. Table 5.1 provide the most severe criterion for each case (worst possible h^+/h). The case *Dirichlet* with $l = 0$ is seen to be unconditionally unstable for small values of h^+/h, independently of the order p. This is not a surprise because the system to obtain the coefficients of the one-sided finite differences becomes singular as $h^+/h \to 0$. The latter is not acceptable because the intersection of the grid and the interface is arbitrary in general, leading to all possible values of h^+/h. The found criteria are not too restrictive, compared to the classical limits obtained by the Von Neumann analysis without interface. The results appear to be independent of N, at least for N not too small ($N = 50$ and $N = 200$ provide the same criteria to within 10 digits).

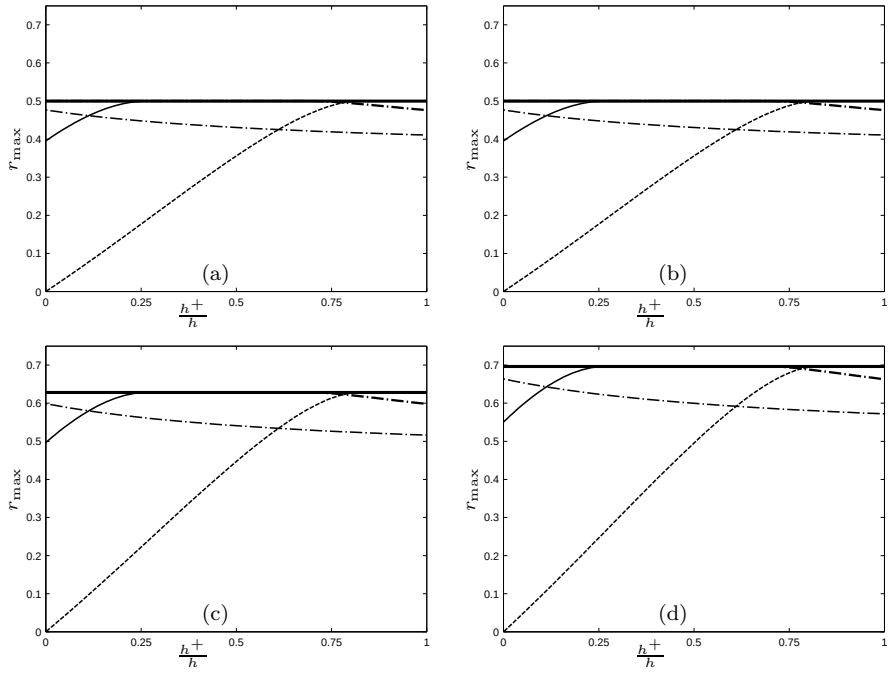


Figure 5.3: *Maximum Fourier number obtained by matrix stability analysis in the one-dimensional case, on a grid with $N = 200$. Runge-Kutta scheme of order $p = 1$ (Euler) (a), $p = 2$ (b), $p = 3$ (c) and $p = 4$ (d). Immersed boundary condition type: Neumann with $l = 0$ (thin solid), with $l = 1$ (thin dash-dotted); Dirichlet with $l = 0$ (thin dashed), with $l = 1$ (thick dash-dotted); classical stability criterion without interface (thick solid).*

BC type	Euler	RK2	RK3	RK4
Neumann ($l = 0$)	0.3954	0.3954	0.4967	0.5506
Neumann ($l = 1$)	0.4108	0.4108	0.5161	0.5721
Dirichlet ($l = 0$)	0.0	0.0	0.0	0.0
Dirichlet ($l = 1$)	0.4761	0.4761	0.5981	0.6630
No interface	0.5000	0.5000	0.6282	0.6963

Table 5.1: *Stability criteria $r_{\max}$ for the different Runge-Kutta integration schemes and for different types of boundary conditions ('no interface' stands for the Von Neumann stability criterion without immersed interface).*

As a matter of fact, the matrix $\mathbf{A}$ is non-normal here and the computed criteria are actually not sufficient. Yet, some numerical experiments that are not reported here have shown that the stability criteria obtained using Eq. (5.4) provide a "reasonable" value for $r_{\max}$, though. Indeed, a computation performed using a time step such that r is slightly greater than $r_{\max}$ ($r - r_{\max} \simeq 10^{-4}$) becomes unstable after several thousands of time steps, whereas choosing $r = r_{\max}$ leads to a numerical solution that remains stable.

In order to provide a sufficient condition, one may alternatively study the ϵ-pseudospectrum $\Lambda_\epsilon(r\mathbf{B})$ of $r\mathbf{B}$ ($\Lambda_\epsilon(\mathbf{E})$ is the set of all ϵ-pseudo-eigenvalues $z \in \mathbb{C}$ such that z is an eigenvalue of $\mathbf{E} + \delta\mathbf{E}$ for some $\delta\mathbf{E}$ with $\|\delta\mathbf{E}\| \leq \epsilon$). According to Reddy and Trefethen [107], the discretization is Lax-stable, except for an algebraic factor, if and only if all ϵ-pseudo eigenvalues of $r\mathbf{B}$ lie within a distance $\mathcal{O}(\epsilon)$ of the stability region $\mathscr{S}$ of the time integration scheme.

Note that, if Eq. (5.3) is satisfied, the non-normality of $\mathbf{A}$ may solely induce some transient growth of $\|\mathbf{A}^n\|$, depending on the initial condition. Asymptotically (for n large), this norm converges to zero, as $\|\mathbf{A}^n\| \leq \rho^n(\mathbf{A})\kappa(\mathbf{V})$, with $\kappa(\mathbf{V})$ the condition number of $\mathbf{V}$.

5.2 Two-dimensional case

The diffusion equation for $\omega(\mathbf{x}, t)$ in two dimensions reads

$$\frac{\partial \omega}{\partial t} = \nu \nabla^2 \omega = \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) .$$

Considering the direction-splitting approach from Chapter 4, one must compute the correction terms J_α^+ and J_α^- (just as in the one-dimensional case, see Eq. (5.1)) at the control point $\mathbf{x}_\alpha$, i.e. at the intersection between the interface and a grid line. The required derivatives $\left(\partial^k \omega / \partial x^k\right)_\alpha$ are therefore evaluated along the grid line on which the control point is defined.

Enforcing a Neumann condition in a splitting approach becomes more difficult in two dimensions, as the first derivatives in the two directions are now coupled :

$$q_\omega \triangleq -\nu \frac{\partial \omega}{\partial n} = -\nu \left(n_x \, \frac{\partial \omega}{\partial x} + n_y \, \frac{\partial \omega}{\partial y} \right) ,$$

with $\mathbf{n} \triangleq (n_x, n_y)$ the normal vector to the body boundary. Two-dimensional Taylor series could be used to develop stencils that evaluate all derivatives appearing in J_α^+ and J_α^- as a function of the imposed normal derivative. Yet, the choice of the stencil points is not trivial, as it may lead to rank deficiency of the coefficient matrix (see Section 4.4.3), making the approach not robust. As a consequence, we use again one-dimensional finite differences

$$\left(\frac{\partial^k \omega}{\partial x^k} \right)_\alpha = (S_q^k)_\alpha \left(\frac{\partial^q \omega}{\partial x^q} \right)_\alpha + \sum_{p=1}^{3} (S_q^k)_p \, \omega_{i+p+l,j} + \mathcal{O}(h^{4-k}) , \tag{5.5}$$

where the one-sided stencil is written here for a control point lying on a grid line $y = y_j$ ($\mathbf{x}_\alpha = (x_\alpha, y_j)$ where x_α is between x_i and x_{i+1}). The basic problem is that we know neither ω_α, nor $(\partial \omega / \partial x)_\alpha$. One could imagine replacing $(\partial \omega / \partial x)_\alpha$ by

$$\left(\frac{\partial \omega}{\partial x} \right)_\alpha = \frac{1}{n_x} \left(\left(\frac{\partial \omega}{\partial n} \right)_\alpha - n_y \left(\frac{\partial \omega}{\partial y} \right)_\alpha \right) .$$

However, this approach is geometrically not robust, as n_x may be equal to zero. It would also leave us with the question of how to compute $(\partial \omega / \partial y)_\alpha$. A different strategy must be adopted. Section 5.2.1 is devoted to addressing that

issue and developing a 2-D immersed interface scheme. Section 5.2.2 studies the stability of this scheme and Section 5.2.3 provides some results on a grid convergence analysis.

5.2.1 Compatible extrapolation scheme for Neumann boundary conditions

The aim of this section is to write a scheme that couples both directions (x and y) without using the information of a neighboring control point, contrary to what is done in Chern and Shu [21].

At this point, we introduce the terminology of *current* and *transverse* direction. When considering a control point that lies on a grid line $y = const$, the current direction is simply the x direction, whereas y is the transverse direction. The roles are inverted if the control point lies on a grid line $x = const$. The current coordinate will be noted as ξ and the transverse coordinate as η. Partial derivatives in these directions will be respectively noted as

$$\begin{aligned} \omega_{\xi}^{(k)} &\triangleq \frac{\partial^k \omega}{\partial \xi^k}(\mathbf{x}_\alpha) \\ \omega_{\eta}^{(k)} &\triangleq \frac{\partial^k \omega}{\partial \eta^k}(\mathbf{x}_\alpha) \ . \end{aligned}$$

This terminology was not needed in the case of a Dirichlet condition, as the derivatives were only computed in the current direction.

The following system provides the ingredients to enforce a flux condition

$$\begin{cases} \omega_\alpha & = S_\alpha^\xi \ \omega_\xi^{(1)} + \displaystyle\sum_{p=1}^{3} S_p^\xi \ \omega_{\xi,p} + \mathcal{O}(h^4) \\ \omega_\alpha & = S_\alpha^\eta \ \omega_\eta^{(1)} + \displaystyle\sum_{p=1}^{3} S_p^\eta \ \omega_{\eta,p} + \mathcal{O}(h^4) \\ \left(\dfrac{\partial \omega}{\partial n}\right)_\alpha & = n_\xi \ \omega_\xi^{(1)} + n_\eta \ \omega_\eta^{(1)} \ , \end{cases} \tag{5.6}$$

where $\omega_{\xi,p}$ correspond to the values of the solution at grid points lying on the current grid line (here, we define again a stencil shifting parameter l_ξ that determines the distance in number of grid points of the first stencil point to the first grid point in the flow). The $\omega_{\eta,p}$ are solution values of some points lying on parallel grid lines, but not necessarily associated to grid points, as can be seen in Fig. 5.4. S_*^ξ and S_*^η are simplified stencil notations for $(S_1^0)_*$, respectively in the current and transverse direction (see Appendix A).

The criterion to select whether the transverse stencil points are chosen for η increasing or for η decreasing (respectively upwards or downwards in Fig. 5.4) is based on the underlying level set value of the adjacent grid points.

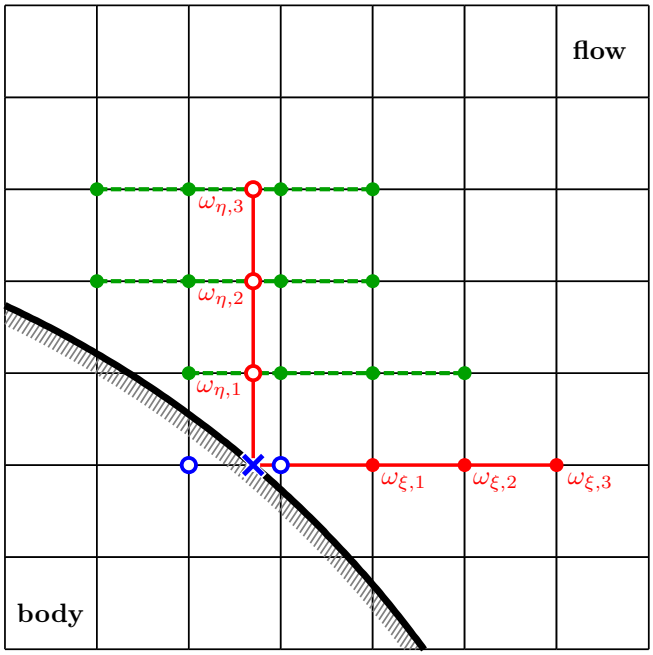


Figure 5.4: Sketch of the "compatible extrapolation" scheme to compute ω_α, $\omega_\xi^{(1)}$ and $\omega_\eta^{(1)}$: control point (blue cross); irregular points affected by the corrections of the control point (blue circles); stencil points for the current direction ($\omega_{\xi,p}$ with $l_\xi = 1$, red bullets); stencil points for the transverse direction ($\omega_{\eta,p}$, red circles); stencil points required for the interpolation of $\omega_{\eta,p}$ (green bullets).

The idea behind this scheme is to write two one-sided schemes for the wall value ω_α, one in the current direction and one in the transverse direction. Both schemes are respectively written as a function of $\omega_\xi^{(1)}$ and $\omega_\eta^{(1)}$, which are still unknown at this stage, as is ω_α. The matching condition for those two evaluations of ω_α is provided by the Neumann boundary condition, which then closes the system. We call this scheme *compatible extrapolation* because it adjusts the first derivatives in order to match the value at the wall, using the provided normal derivative.

The 3-by-3 system of Eq. (5.6) can be solved for ω_α, $\omega_\xi^{(1)}$ and $\omega_\eta^{(1)}$. The next step consists in plugging the result into the one sided stencil Eq. (5.5) for the computation of the jumps needed by the correction terms J_α. We have the

choice to use ω_α or $\omega_\xi^{(1)}$ as the wall information (using both values inside the correction terms is also possible). Using $\omega_\xi^{(1)}$ and $l = 0$ for Eq. (5.5) leads to a lower leading error term in practice. The explicit solution of the latter system yields the following expressions

$$\begin{aligned} \omega_\xi^{(1)} &= \frac{1}{\mathcal{D}} \left(n_\eta \, S^\eta - n_\eta \, S^\xi + S_\alpha^\eta \, \left(\frac{\partial \omega}{\partial n} \right)_\alpha \right) + \mathcal{O}(h^3) \\ \omega_\alpha &= S_\alpha^\xi \, \omega_\xi^{(1)} + S^\xi + \mathcal{O}(h^4) \, , \end{aligned} \tag{5.7}$$

with $\mathcal{D} \triangleq n_\xi S_\alpha^\eta + n_\eta S_\alpha^\xi$ the determinant, $S^\xi \triangleq \sum_{p=1}^3 S_p^\xi \; \omega_{\xi,p}$ and $S^\eta \triangleq \sum_{p=1}^3 S_p^\eta \; \omega_{\eta,p}$. The scheme is still linear, as expected, and the coefficients are only geometry dependent. The order of the error term for $\omega_\xi^{(1)}$ is based on the observation that $\mathcal{D} = \mathcal{O}(h)$ (see Appendix A). The order of both error terms are also consistent with the precision required by the correction terms in order to ensure a local second order scheme.

Numerically, the system is well-posed because the value of the determinant $\mathcal{D}$ never goes to zero if the current stencil shift is equal to $l_\xi = 1$, as represented in Fig. 5.5. Indeed, $|S_\alpha^\xi|/h$ only depends on h^+/h and $|S_\alpha^\eta|/h$ is a constant. All possible geometrical configurations can be covered by letting vary the angle $\theta \triangleq \arctan(n_\eta/n_\xi)$ between $-\pi$ and π. Fig. 5.5 show the behavior of $|\mathcal{D}|$ as a function of h^+/h and θ ($l_\xi = 0$ in Fig. 5.5(a) and $l_\xi = 1$ in Fig. 5.5(b)).

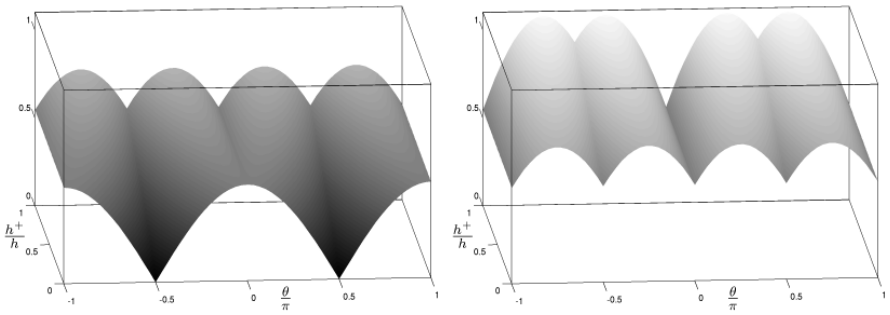


Figure 5.5: *Behavior of $|\mathcal{D}|$ as a function of $\theta \triangleq \arctan(n_\eta/n_\xi)$ and h^+/h for different grid shifting parameters : (a) $l_\xi = 0$ and (b) $l_\xi = 1$.*

We also stress that this scheme easily generalizes to three dimensions. Indeed, the system remains closed, since one wall evaluation would simply be added in the third grid direction, using the associated partial derivative as the fourth unknown.

Fig. 5.4 also shows that the values $\omega_{\eta,p}$ have to be interpolated from adjacent grid point values. The approach followed here consists in taking 4 stencil nodes $\omega_{\eta,p,s}$ along the current direction (green bullets in Fig. 5.4). This ensures a fourth order interpolation (see Appendix A), which is required for the second order precision of the correction terms

$$\omega_{\eta,p} = \sum_{s=1}^{4} T_s^0 \, \omega_{\eta,p,s} + \mathcal{O}(h^4) \, .$$

The stencil points are chosen so as to be as much as possible centered around the interpolation point, yet remaining outside of the body. In some cases, where the interface becomes perpendicular to the current direction, the envelope of the transverse interpolation nodes may result in an extrapolation instead of an interpolation of $\omega_{\eta,p}$. Nevertheless, in practice, this issue does not appear to be problematic in terms of precision and stability of the scheme.

5.2.2 Stability analysis

The stability of this scheme is studied, as in the one-dimensional case. We can compute the eigenvalues associated with the amplification matrix of the spatial derivative discretization, including the correction terms near the interface.

The discretization is performed on a $(N+1) \times (N+1)$ grid defined in the domain $[-D; D] \times [-D; D]$ with a Dirichlet condition on the outer boundary (the cells are thus isotropic and $\Delta x = \Delta y = h = 2D/N$). The inner boundary is a circle of diameter $D = 2R$, with R its radius. Contrary to previously, we may not isolate the influence of the correction associated to one specific control point here. Therefore, we conduct the stability analysis by letting vary the position $\mathbf{x}_c$ of the circle center with respect to the grid and we monitor the associated distribution of the geometric parameter h^*. Similarly to the one-dimensional case with h^+, h^* is defined as the distance between the control point $\mathbf{x}_\alpha$ and the grid point on the flow side affected by the correction coming from $\mathbf{x}_\alpha$. Three different representative positions $\mathbf{x}_c$ are considered here, as shown in Fig. 5.6. The latter are assumed to be the "worst cases".

Results of the stability analysis are shown for different resolutions N in Table 5.2, when enforcing a Dirichlet condition on the circle boundary, and in Table 5.3, when enforcing a Neumann condition using the compatible extrapolation scheme. The stability analysis is performed for the three positions $\mathbf{x}_c$ and the results are shown at the position $\mathbf{x}_c$ leading to the most severe stability criteria.

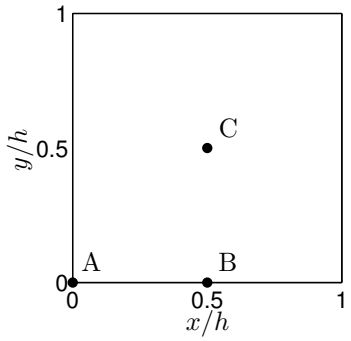


Figure 5.6: *Sketch of the different positions of the center of the circle* $\mathbf{x}_c$ *used for the stability analysis;* $A : \mathbf{x}_c/h = (0,0)$, $B : \mathbf{x}_c/h = (0.5,0)$ *and* $C : \mathbf{x}_c/h = (0.5,0.5)$.

	$\mathbf{x}_c$						
	A	B	C	Euler	RK2	RK3	RK4
$N = 20$	×			0.2499	0.2499	0.3140	0.3481
$N = 30$		×		0.2506	0.2506	0.3148	0.3489
$N = 40$	×			0.2483	0.2483	0.3119	0.3457
$N = 50$		×		0.2503	0.2503	0.3145	0.3486
No interface				0.25	0.25	0.3141	0.3482

Table 5.2: *Stability criteria* $r_{\max}$ *obtained for a Dirichlet boundary condition* ($l = 1$) *applied at the circle interface. The "*$\times$*"-sign indicates the position of* $\mathbf{x}_c$ *that leads to these criteria, at different resolutions N ('no interface' stands for the 2-D Von Neumann stability criterion without immersed interface).*

	$\mathbf{x}_c$						
	A	B	C	Euler	RK2	RK3	RK4
$N = 20$	×			0.2127	0.2127	0.2672	0.2962
$N = 30$		×		0.2293	0.2293	0.2881	0.3193
$N = 40$	×			0.2280	0.2280	0.2865	0.3176
$N = 50$		×		0.2217	0.2217	0.2786	0.3088
No interface				0.25	0.25	0.3141	0.3482

Table 5.3: *Stability criteria* $r_{\max}$ *obtained for a Neumann boundary condition* ($l = 0$) *applied at the circle interface, using the compatible extrapolation scheme. The "*$\times$*"-sign indicates the position of* $\mathbf{x}_c$ *that leads to these criteria, at different resolutions N ('no interface' stands for the 2-D Von Neumann stability criterion without immersed interface).*

These results certainly do not cover all possible configurations but they essentially show that using an immersed interface correction in two dimensions gives rise to less severe stability criteria compared to that of the one dimensional case. Moreover, the criteria do not differ much from those obtained using the Von Neumann analysis, valid for cases without interface.

One may also notice that the compatible extrapolation scheme for the Neumann immersed interface BC is sensibly less stable than the Dirichlet BC immersed interface scheme, though not significantly. Remarkably, both types of boundary conditions share the same location $\mathbf{x}_c$ (for all considered grid resolutions N), where the scheme leads to the most severe stability criterion. It is also worth mentioning that the design of a Neumann enforcing scheme that is based on extrapolation of the tangential derivative produces eigenvalues with $\Re\{\lambda_k(\mathbf{B})\} > 0$, which is unconditionally unstable. The compatible extrapolation however removes this undesired feature by reducing the level of extrapolation while providing an additional information at the wall (the flux), turning the extrapolation into an interpolation. The distributions of h^*/h related to the most unstable location $\mathbf{x}_c$ at different grid resolutions N (according to Tables 5.2 and 5.3) are given in Fig. 5.7.

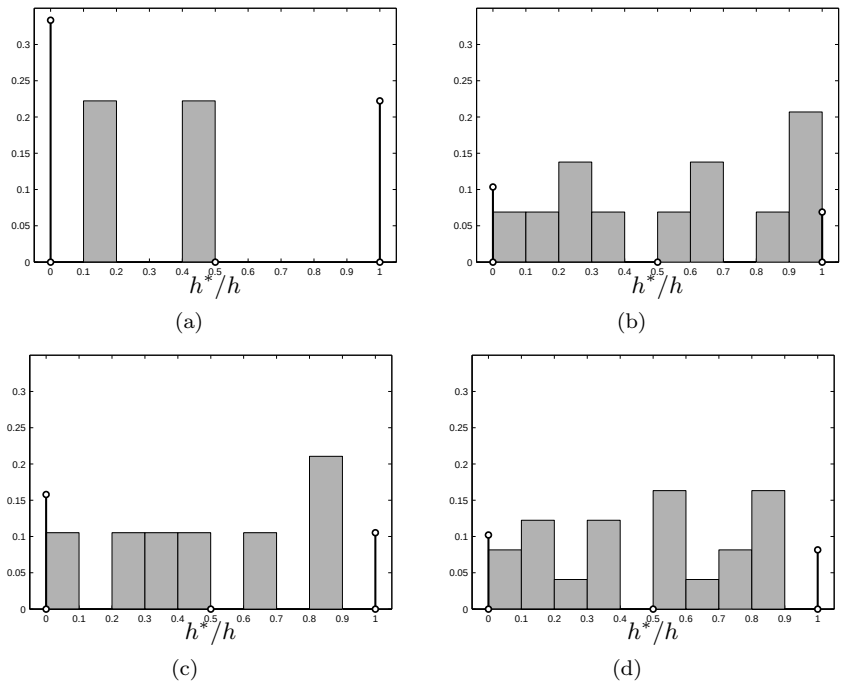


Figure 5.7: *Normalized distribution of h^*/h for (a) $N = 20$ ($\mathbf{x}_c$ at A), (b) $N = 30$ ($\mathbf{x}_c$ at B), (c) $N = 40$ ($\mathbf{x}_c$ at A) and (d) $N = 50$ ($\mathbf{x}_c$ at B). The open circles represent the pointwise distribution of the particular configurations $h^*/h = 0,\ 0.5$ and 1.*

One may also compare in Fig. 5.8 the eigenvalues $\lambda_k(\mathbf{B})$ obtained when enforcing either a Dirichlet condition, either a Neumann condition for the most unstable configuration at $N = 50$. The Dirichlet condition seems to produce eigenvalues with higher imaginary parts.

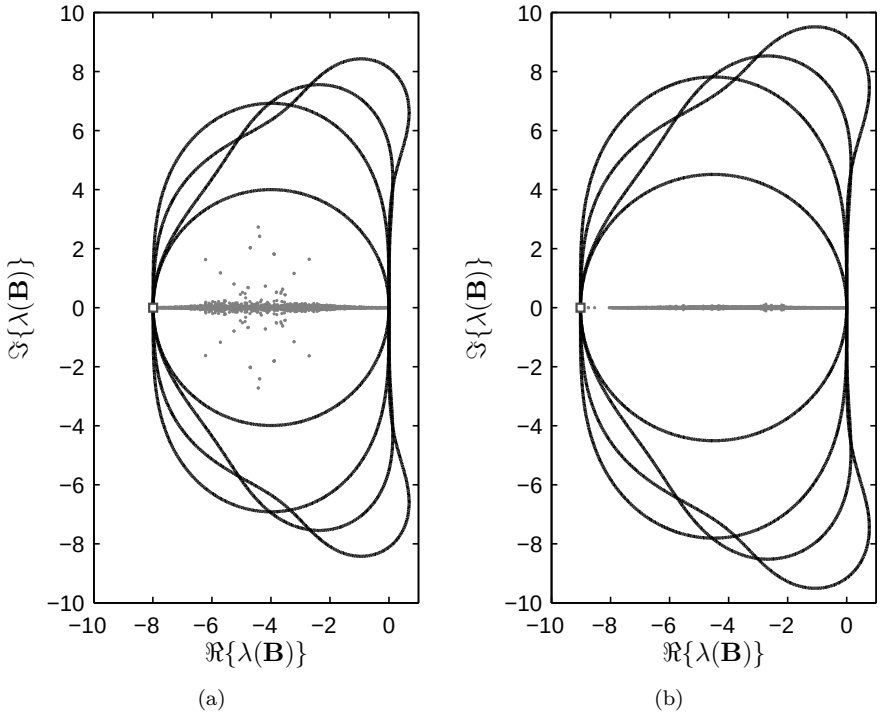
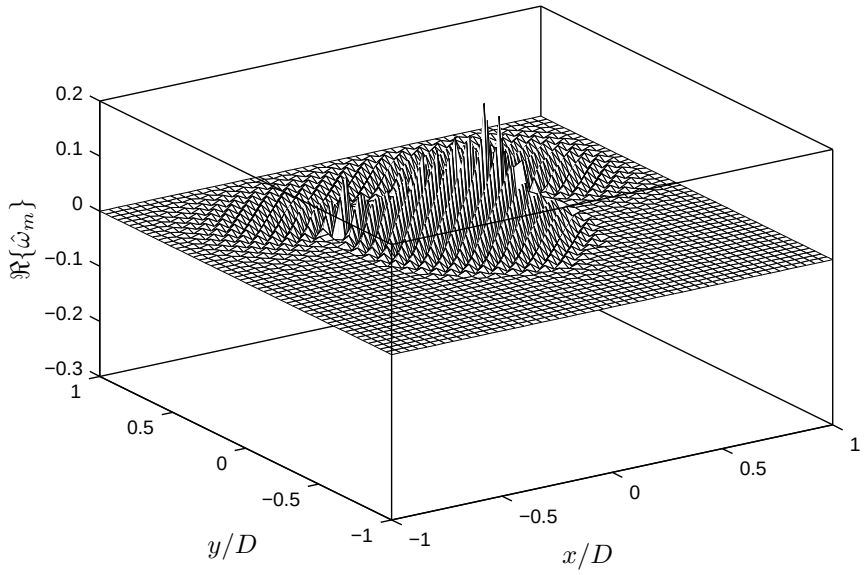
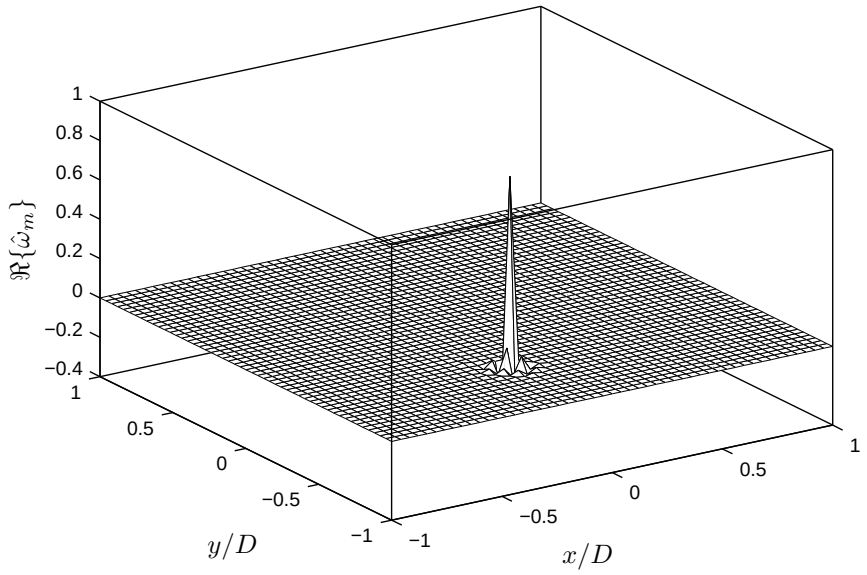


Figure 5.8: *Eigenvalues $\lambda_k(\mathbf{B})$ obtained when enforcing (a) a Dirichlet condition and (b) a Neumann condition at position $\mathbf{x}_c/h = (0.5, 0)$ and for $N = 50$. The square represents the most unstable mode $\lambda_m(\mathbf{B})$ corresponding to $\rho(\mathbf{A})$. The figure also shows the associated stability regions for some Runge-Kutta time integration schemes up to order 4 and scaled by the respective r_{max}.*

The associated most unstable modes $\hat{\omega}_m$ corresponding to $\rho(\mathbf{A})$ (the square in Fig. 5.8) are shown in Fig. 5.9. Again, the Dirichlet and Neumann condition enforcement noticeably differ in their nature. One may easily distinguish the circular immersed interface for the Dirichlet case whereas, for the Neumann condition, only one particular stencil point is seen to contribute to the mode.



(a)



(b)

Figure 5.9: *Real part of the most unstable mode $\hat{\omega}_m$ obtained when enforcing (a) a Dirichlet condition and (b) a Neumann condition at position $\mathbf{x}_c/h = (0.5, 0)$ and for $N = 50$.*

5.2.3 Grid convergence study

As a next step, we study the convergence of the developed scheme. The test case we study consists in solving the diffusion of a Gaussian function. The well-known solution of this problem in a 2-D unbounded domain reads

$$\omega(x,y,t) = g(x,y,t) \triangleq \omega_g \, \frac{\sigma^2}{(\sigma^2+4\nu t)} \, \exp\left[-\frac{(x-x_g)^2+(y-y_g)^2}{(\sigma^2+4\nu t)}\right] , \quad (5.8)$$

with $\mathbf{x}_g \triangleq (x_g, y_g)$ the offset of the function with respect to the origin and σ the initial core size of the Gaussian. The problem is solved outside of a circle of radius R, providing the analytical solution as a boundary condition on the circle boundary. The solution to this problem is thus still $g(x,y,t)$ outside of the circle. The initial condition is

$$\omega_0(x,y) \triangleq \omega(x,y,0) = \begin{cases} 0 & \text{if } r < 0 \\ g(x,y,0) & \text{if } r \geq R \end{cases} ,$$

with $r \triangleq |\mathbf{x}-\mathbf{x}_c|$. We also provide the analytical solution on the outer boundary of the computational domain as a Dirichlet condition:

$$\forall (x,y) \in \partial\Omega_{comp} : \;\; \omega(x,y,t) = g(x,y,t) \; .$$

On the inner boundary, on the flow side (i.e. just outside the circle), we either impose a Dirichlet condition or a Neumann condition using the immersed interface schemes that have been developed:

$$\begin{aligned} \forall (x,y) \in \partial\Omega_b \quad &: \quad \omega(x,y,t) = g(x,y,t) \\ &\text{or} \\ \forall (x,y) \in \partial\Omega_b \quad &: \quad \frac{\partial\omega}{\partial n}(x,y,t) = \frac{\partial g}{\partial n}(x,y,t) \; . \end{aligned}$$

Inside the circle, the boundary condition is homogeneous: $\omega(x,y,t) = 0$ for $r < R$ and $t \geq 0$.

The problem is then solved on the grid $(N+1) \times (N+1)$ that has already been defined in the previous section.

The following values have been chosen for the grid convergence study:

$$\begin{aligned} \sigma/R &= 1 \\ \mathbf{x}_c/R &= (2, -6)\ \pi 10^{-3} \\ \mathbf{x}_g/R &= (0.62, -0.8)\ . \end{aligned} \tag{5.9}$$

The center of the circle $\mathbf{x}_c/R$ has an irrational offset with respect to the origin. We hence ensure quasi arbitrary intersections between the interface and the grid and thus avoid any symmetry that could affect the numerical error. The Gaussian center $\mathbf{x}_g/R$ is near the interface so as to provide large tangential gradients of the solution along the interface. The initial condition corresponding to these parameters is shown in Fig. 5.10.

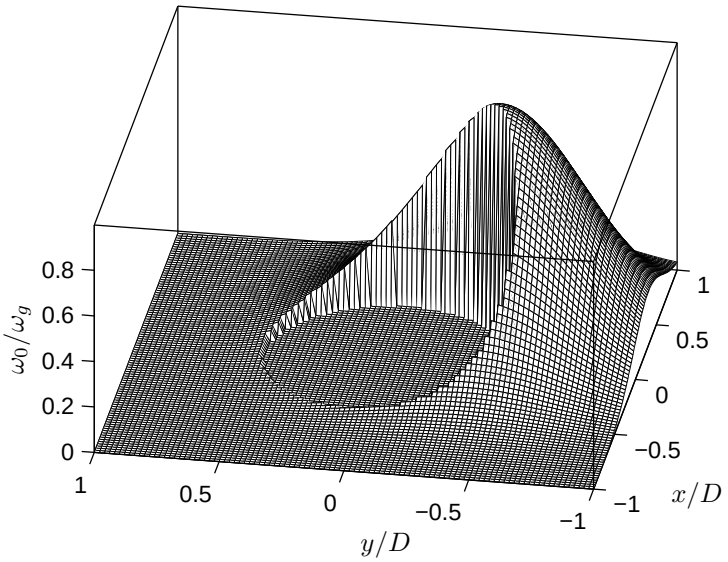


Figure 5.10: *Initial condition $\omega_0(x, y) = \omega(x, y, 0)$ considered for the grid convergence study (here $N = 100$).*

The spatial convergence of the numerical discretization is studied in conjunction with a Euler time integration scheme, similarly to Eq. (5.2). The time step $\nu\Delta t/h_{min}^2 = 0.01$ is chosen to be very small (and identical for all resolutions N) so that the error due to the mesh size h prevails compared to the the error due to the time step (h_{min} is the mesh size of the most refined grid considered in the convergence study).

We also define the following error norms:

$$\begin{aligned} \epsilon_2 &\triangleq \|\epsilon\|_2 \triangleq \frac{h}{\omega_g D}\left(\sum_{i,j}(\tilde{\omega}_{ij}^M - \omega_{ij}^M)^2\right)^{\frac{1}{2}} \\ \epsilon_\infty &\triangleq \|\epsilon\|_\infty \triangleq \frac{1}{\omega_g}\max_{i,j}|\tilde{\omega}_{ij}^M - \omega_{ij}^M| \,, \end{aligned}$$

with $\tilde{\omega}_{ij}^M$ and $\omega_{i,j}^M$ respectively the numerical and analytic solutions at the grid node $\mathbf{x}_{ij}$ at the time $T \triangleq M\Delta t = 0.1\ \sigma^2/(4\nu)$. Fig. 5.11 and Table 5.4 show the grid convergence results. As expected, the second order convergence of the error norms is confirmed for both types of boundary conditions. Moreover, the level of the error is fairly comparable in both cases.

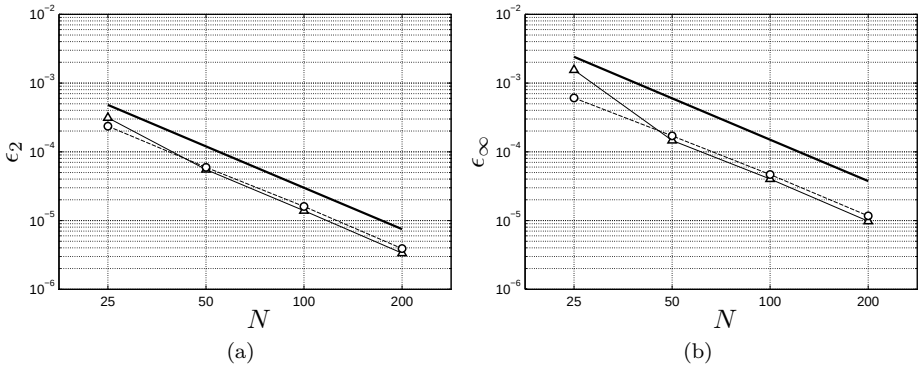


Figure 5.11: *Grid convergence study for the solution of a diffusing Gaussian function, while imposing a Dirichlet condition (thin solid line with "Δ"-signs) and a Neumann condition (thin dashed line with "$\circ$"-signs) on $\partial\Omega_b$: (a) L_2 error norm and (b) L_∞ error norm. The thick solid line shows a second order slope.*

	Dirichlet	Neumann
ϵ_2	2.05	2.03
ϵ_∞	2.04	2.00

Table 5.4: *Observed order of convergence between both most refined grids in the case of a diffusing Gaussian function.*

Chapter 6

Interpolation between particles and grid in the presence of a wall

Vortex particle-mesh methods (VPM), used for the simulation of unsteady vortical flows, are based on a combination of information carried by particles and an underlying grid. This information has to be mapped from one support to another at different steps of the numerical algorithm. The transfer of information from the particles to the grid is conventionally called P2M (*particles-to-mesh*) whereas, the reverse mapping is called M2P (*mesh-to-particles*).

Historically, the former P2M operation was already used in the framework of purely Lagrangian methods, in order to periodically reinitialize the set of particles after a few time steps, using a set of new particles whose positions coincide with the underlying grid nodes. This reinitialization procedure, also known as *redistribution* (or *remeshing*), is required as the Lagrangian particles are subject to distortion by the flow. The absence of redistribution would lead to particle clustering and/or depletion, and thus to a less accurate representation of the vorticity field as time evolves [71]. On the contrary, performing this redistribution too often introduces higher numerical errors. Therefore, the right balance has to be chosen with respect to the redistribution frequency. The present chapter focuses on the spatial accuracy of the particle-mesh interpolation and the effect of the redistribution frequency onto the numerical errors is subsequently studied in Chapter 8.

Considering first the 1-D case, and having an old set of distorted particles (with the associated positions $\tilde{x}_q$ and intensities $\tilde{\alpha}_q \triangleq \int_{\Omega_q} \omega \, dx$), the intensities α_p of the new particles are computed as

$$\alpha_p = \sum_q \tilde{\alpha}_q \, w\left(\frac{x_p - \tilde{x}_q}{\Delta x}\right) , \tag{6.1}$$

with $w(\xi)$ a compact support interpolation kernel, and $x_p = x_0 + p \, \Delta x$ the positions of the new particles coinciding with a uniform grid of spacing Δx. From an algorithmic point of view, we loop over the index q and every old particle gives the fraction $w((x_p - \tilde{x}_q)/\Delta x)$ of its intensity $\tilde{\alpha}_q$ to each new particle at x_p lying in its interpolation range (see Fig. 6.1).

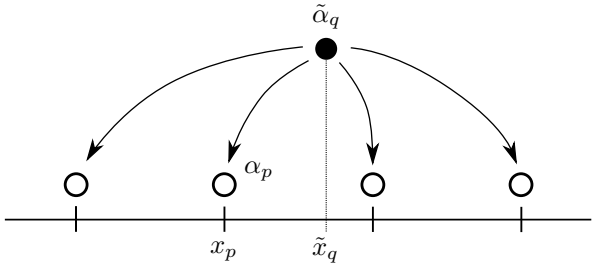


Figure 6.1: *Sketch of the P2M redistribution scheme: old particle at position $\tilde{x}_q$ and with intensity $\tilde{\alpha}_q$ (bullet); new particle at position x_p and with intensity α_p (circles).*

As an example, the third order kernel $w(\xi) = M'_4(\xi) \in C_1(\mathbb{R})$ derived by Monaghan [93] is widely used in the Lagrangian methods community:

$$M'_4(\xi) = \begin{cases} 0 & \text{if } |\xi| > 2 \, , \\ \dfrac{1}{2}(2 - |\xi|)^2(1 - |\xi|) & \text{if } 1 \leq |\xi| \leq 2 \, , \\ 1 - \dfrac{5}{2}|\xi|^2 + \dfrac{3}{2}|\xi|^3 & \text{if } |\xi| \leq 1 \, . \end{cases} \tag{6.2}$$

The redistribution for higher-dimensional problems is obtained by using a tensor product. For instance, in 2-D, the scheme reads:

$$\alpha_p = \sum_q \tilde{\alpha}_q \, w\left(\frac{x_p - \tilde{x}_q}{\Delta x}\right) w\left(\frac{y_p - \tilde{y}_q}{\Delta y}\right) , \tag{6.3}$$

with $\alpha_p \triangleq \int_{\Omega_p} \omega \, dS \triangleq \omega_p S_p$ ($S_p = \Delta x \Delta y = h^2$ for an isotropic grid).

The redistribution is computationally quite efficient, especially for an unbounded flow domain, as it is a local procedure. It conserves moments up to

order 2, by construction (i.e., the integral of the vorticity, the linear impulse and the angular impulse).

However, the addition of a solid wall requires some modifications in order to prevent the creation of new particles outside of the flow domain. Indeed, in 1-D, the M'_4 kernel creates two new particles on both sides of an old particle, which is undesirable in the vicinity of the wall. Moreover, high order kernels like M'_4 provide a poor representation of discontinuous functions (recall that the vorticity is non-zero at the wall and that it vanishes inside the body when at rest), due to the negative lobe of $w(\xi)$ for $|\xi| > 1$.

It should be mentioned that some approaches do not require a particular treatment of the redistribution scheme near the wall. As an example, penalization methods account for the presence of the wall by computing a body force [50]. This regularized body force is computed on the grid so as to approximately enforce the proper velocity at the wall and inside the body. Hence, the flow is extended inside the body and the same redistribution scheme is performed everywhere. The drawback of this approach comes from the regularization of the body force that induces a smearing of the solution field near the wall. In the present approach a "sharper" treatment of the wall is sought for.

Ploumhans et al. [102] use decentered and lower order schemes near the wall, along with a procedure to choose which direction to perform the redistribution first (so as to get the "least decentered" configuration). This approach was shown to work well for purely Lagrangian vortex methods but, as was mentioned in [132], the kernels are very oscillatory and produce negative undershoots in the flow domain of higher amplitude than those of M'_4. Fig. 6.2 shows a redistributed field using the approach from [102]. Even for constant intensity particles (i.e. partition of unity), the obtained solution exhibits large oscillations near the wall. No particles are created in the solid body, but the smoothness of the redistributed field is not guaranteed. The inconsistency between the decentered kernels and the centered kernel used far from the boundaries further accentuates this phenomenon. For vortex particle-mesh methods, the smoothness of the field is crucial, as the spatial operators are computed on the underlying grid, using finite differences.

The approach adopted in Cottet and Poncet [35] lies somewhere in between Gazzola et al. [50] and Ploumhans et al. [102], in the sense that the same interpolation formulas are used everywhere, i.e. also near the wall. Similarly,

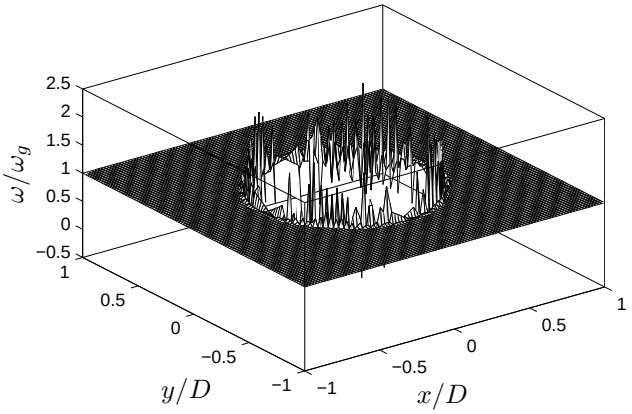


Figure 6.2: *Particle-to-mesh interpolation using the approach from [102]. The numerical setup of the test case, as well as the particle positions are reported in Section 6.1.3.*

the computation of the diffusion term is also performed identically at all particle positions using the PSE scheme [41], regardless of the wall position. Based on the residual slip velocity at the wall, the integral formulas aiming at enforcing the no-slip condition are then used in order to correct the spurious vorticity that was introduced during the interpolation and diffusion steps (the procedure correspond to the no-slip enforcement explained in Chapter 3 and the formulas are similar to those from [102]; see also Appendix E.8).

Another possibility consists in using "image" particles, as was done in [83] and successfully applied in the framework of a 3-D VPM method. The idea behind this approach is to provide a zero-flux condition at the wall by creating particles inside the body whose intensities correspond to the original flow particles and whose positions are symmetric to the original ones relative to the wall. The part of the intensity that is lost inside the body during the redistribution of the original particle is then expected to be recovered by the redistribution of the image particle inside the flow. For arbitrary intersections of the body interface with the grid, this is no longer true and one may rescale the image particle so as to stay conservative. Fig. 6.3 shows the application of this rescaled scheme (similar to [83]); this scheme also implements a "halo" around the body where particles may not exist, as was also done in [83] (typically $0 \leq d < 0.25\ h$ with h the grid size and d the distance to the wall; this halo is used in order to prevent particles from being too close to the wall, as the regularity of the underlying panel method solution may otherwise be affected). The oscillations are still present, but their amplitude is lower than in Fig. 6.2.

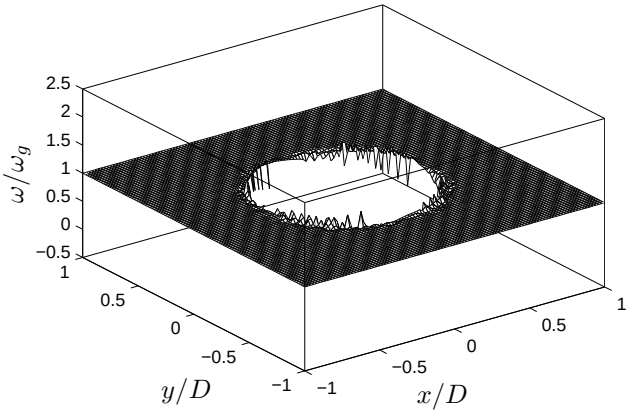


Figure 6.3: *Particle-to-mesh interpolation using an approach similar to [83].*

To a more general extent, the redistributed field resulting from a P2M operation may also lose its smoothness *without* the presence of a wall, when the particles are arbitrarily scattered (with local depletion or clustering). Following [34], the accuracy of the redistribution procedure may be linked to the moment conservation of the kernel. However, only a weak measure of the error is provided (using a test function), which does not give an indication about the local behavior of the redistributed field. In order to maintain its smoothness, the particles should "uniformly" cover the entire vorticity support (for regularized Lagrangian methods this amounts to require that their cores overlap)[1].

By contrast, the M2P procedure (going from a regular grid information to a possibly scattered set of particles) follows the rules of classical interpolation, allowing the analysis of the local error. In this case, the conservation of moments is not addressed, since in a VPM method, this type of operation is only used to interpolate onto the particles the right-hand sides of the evolution equations, namely the velocity and the Laplacian of the vorticity, in 2-D (plus the vortex stretching in 3-D). Yet, it could be argued that the conservation of $\nabla^2\omega$ is important, though. Nevertheless, it is quite natural to favor the local accuracy over the moments conservation here.

We first deal with the mesh-to-particles interpolation in Section 6.1. Based on that, Section 6.2 explains how to adapt the redistribution technique in order to fit inside an immersed interface framework.

[1]In practice, the particles move according to a smooth velocity field and thus their positions remain "organized" (if redistribution occurs regularly). Hence, they are not arbitrarily scattered.

6.1 Mesh-to-particles interpolation (M2P)

Basically, the tools for the M2P interpolation are the same as those used for the P2M approach. As the 2-D interpolation stencil is a composition of one-dimensional schemes (see Eq. (6.3)), we first deal with the 1-D case in Section 6.1.1. The generalization of the approach to 2-D is detailed in Section 6.1.2 and then validated in Section 6.1.3.

6.1.1 One-dimensional case

Similarly to Eq. (6.1), the kernel $w(\xi)$ may also be used to interpolate the value of a function u (rather than the particle-carried quantity $\int_{\Omega_q} u\,dx$) at an arbitrary location x, based on the values $u_j = u(x_j)$ that the function takes at the nodes x_j of a uniform grid with spacing Δx:

$$\tilde{u}(x) = \sum_j u_j \; w\left(\frac{x - x_j}{\Delta x}\right) \;, \tag{6.4}$$

with $\tilde{u}(x)$ the interpolation of $u(x)$. Algorithmically, we loop again over the set of points x where the interpolation of u is required (i.e., at the particles' positions x_p) and we compute Eq. (6.4) by identifying the contributing grid nodes (see Fig. 6.4).

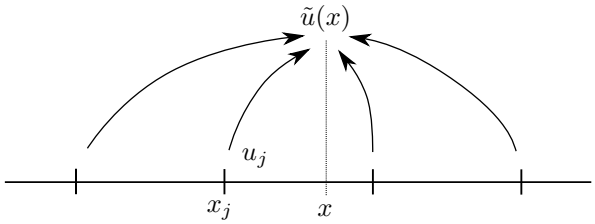


Figure 6.4: *Sketch of the M2P interpolation scheme: interpolation $\tilde{u}(x)$ of u at x, using the node values u_j at x_j.*

The M_4' kernel is third order accurate and thus interpolates exactly quadratic functions in the sense that

$$x^m = \sum_j (x_j)^m \; w\left(\frac{x - x_j}{\Delta x}\right) \quad \text{for } m = 1, 2 \;.$$

This observation gives some hints about a possible extension of the approach to wall-bounded fields using ghost nodes inside the body. The computation

of those ghost nodes follows the lines of classical finite difference methods and the ghost cell approach [85, 126]. We wish to construct an extended field so as to be able to apply the same interpolation scheme anywhere, independently of the wall location. If we compute the values of the ghosts using a quadratic function matching some prescribed information at the wall, we ensure that the interpolated function will also satisfy these conditions at the wall, as the M_4' kernel exactly interpolates polynomials of degree 2.

As can be seen in Fig. 6.5, for a particle lying between the interface x_α and x_{i+1}, we need two ghost points u_{i-1}^* and u_i^* inside the body in order to interpolate the function at the particle position. Assuming that we know the value $u_\alpha \triangleq u(x_\alpha)$ of the function $u(x)$ at the wall and its first derivative $u_\alpha^{(1)} \triangleq u^{(1)}(x_\alpha)$, we may write the following system of equations for u_{i-1}^* and u_i^*

$$\begin{cases} T_0^0\, u_{i-1}^* + T_1^0\, u_i^* &= u_\alpha - T_\beta^0\, u_\beta \\ T_0^1\, u_{i-1}^* + T_1^1\, u_i^* &= u_\alpha^{(1)} - T_\beta^1\, u_\beta\ , \end{cases} \tag{6.5}$$

with $x_\beta \triangleq x_\alpha + \Delta x$ and $u_\beta \triangleq u(x_\beta)$, which is not known yet. The coefficients $T_\cdot^k$ are defined in Appendix A and allow to compute the k^{th} derivative of a function at x_α using the stencil points $\{x_\beta, x_{i-1}, x_i\}$. The reason why the value u_β is chosen and not u_{i+1} stems from considerations about numerical robustness. Choosing u_{i+1} instead of u_β would have lead to a rank deficient system Eq. (6.5) when $x_\alpha \to x_{i+1}$. As to the value u_β, it is obtained by using the scheme

$$u_\beta = \tilde{T}_0^0\, u_{i+1} + \tilde{T}_1^0\, u_{i+2} + \tilde{T}_2^0\, u_{i+3}\ , \tag{6.6}$$

where the coefficients $\tilde{T}_{(\cdot)}^0$, defined in Appendix A, correspond to those of the stencil points $\{x_{i+1}, x_{i+2}, x_{i+3}\}$. Fig. 6.5 sums up the procedure for the 1-D case. The approach eventually consists of a simple extrapolation of the solution from the flow domain, using additionally the function value at the wall and its derivative. The rationale for this procedure relies on the fact that the previously developed tools from Appendix A can be reused.

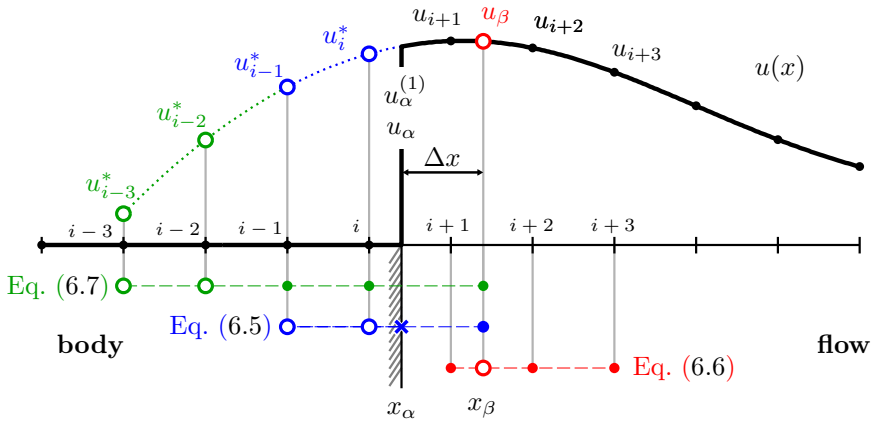


Figure 6.5: *Sketch of the ghost computation for the 1-D M2P interpolation scheme: position of the immersed interface at x_α with the provided wall data u_α and $u_\alpha^{(1)}$ (blue cross); stencil to compute u_β (in red); stencil to compute the near-wall ghosts u_i^* and u_{i-1}^* (in blue); stencil to compute the extended ghosts u_{i-2}^* and u_{i-3}^* for the 2-D case (in green); open circles represent computed data and bullets represent provided data (possibly by a preceding computation).*

6.1.2 Two-dimensional case

The approach may be generalized to two dimensions by considering again the tensor product of the M_4' interpolation kernels, similarly to Eq. (6.3). Former to this 2-D interpolation, the ghosts have to be computed along both grid directions based on the wall information that resides on the control points: the intersection of the body interface and the grid lines, as defined previously. This wall information consists of partial derivatives along these grid lines.

The 2-D case adds two more difficulties compared to the 1-D case. First, the ghost grid field is over-determined, as the application of the previous methodology along both directions may lead to different ghost values for some grid nodes inside the body (see Fig. 6.6). However, we wish to compute this field as a pre-processing step, in order to then apply the interpolation scheme without any particular treatment for the particles near the wall. The uniqueness of the ghost values is thus a prerequisite for this approach and this calls for an adjustment of the algorithm. By contrast, Tseng et al. [126] use multi-dimensional schemes to construct the ghosts, avoiding thus this over-determination. This is hardly applicable here, as the wall information at the control points consists of derivatives along the grid lines and therefore strongly suggests using the decoupled one-dimensional approach.

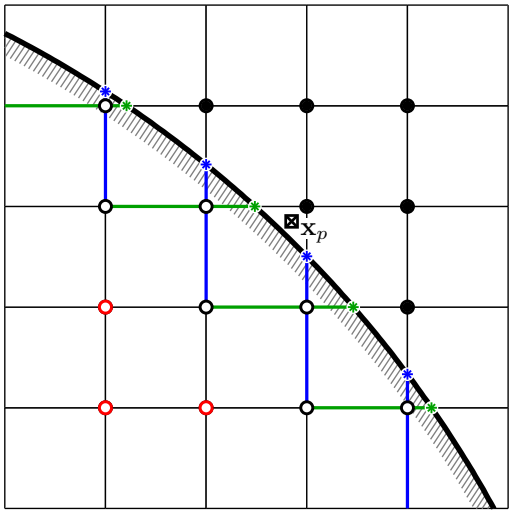


Figure 6.6: *Sketch of the 2-D M2P interpolation scheme: control points (green asterisks for x direction, blue asterisks for y direction); range of the 1-D methodology for the ghost computation and applied in both directions (green and blue lines); particle at $\mathbf{x}_p$ near the wall (black crossed square); flow grid nodes used for the M2P interpolation at $\mathbf{x}_p$ (black bullets); grid ghosts required for the M2P interpolation at $\mathbf{x}_p$ (open circles); ghosts provided by the 1-D methodology (black open circles); ghosts missed by the 1-D methodology (red open circles).*

Secondly, restricting the number of ghosts to two nodes in the x and y directions, as was done for the 1-D case, may lead to "holes" in the ghost distribution required for the 2-D case. More precisely, Fig. 6.6 shows that particles near the wall may still see "empty" ghost nodes in their 4x4 interpolation stencil, which leads to a loss of accuracy.

The second difficulty can be circumvented by further extending the envelope of the ghosts, i.e., by adding two more ghost points, as shown in Fig. 6.5. We thus distinguish between *near-wall* ghosts (u^*_i and u^*_{i-1}) and *extended* ghosts (u^*_{i-2} and u^*_{i-3}). A consistent way to compute the values of the latter in 1-D consists in realizing that we may simply use the same quadratic function $p_2(x)$ determined by Eq. (6.5):

$$u^*_{i-2} = p_2(x_{i-2}) \quad \text{and} \quad u^*_{i-3} = p_2(x_{i-3}) \,, \tag{6.7}$$

or, equivalently, by applying two different $T^0_{(\cdot)}$ schemes using the solution values at the nodes $\{x_\beta, x_{i-1}, x_i\}$. This procedure is clearly a further extrapolation of the ghost values and we do not control their behavior. However, the value at the particle position $\mathbf{x}_p$ may be rewritten as a linear combination of the solution

values in the flow and the wall data, and the envelope of this information is thus a convex hull enclosing the evaluation point $\mathbf{x}_p$.

The over-determination of the ghosts is handled by computing a weighted contribution of the information coming from the different control points. Geometrically, a grid node inside the body may receive at most 4 different ghost values, as the node may be surrounded by 4 control points in the worst case, in 2-D. A distinction is made between near-wall and extended ghosts, giving a higher priority to near-wall ghosts, since the distance to their originating control point is smaller, making the information more reliable (the procedure is detailed hereafter).

Eqs. (6.5)-(6.7) are expanded as a set of coefficients ζ_{ql} (stored at all control points) that are used for the construction of the ghosts. The near-wall ghost correspond to $q = 1, 2$ and the extended ghosts to $q = 3, 4$ (cf. the 1-D case in Fig. 6.5 with u^*_{i+1-q}); the index l loops over all elements required for the ghost computation, namely the wall data and the field value at some grid nodes (including ghost values for the extended ghosts). In 1-D, we have

$$u^*_{i+1-q} = \zeta_{q,1}\ u_\alpha + \zeta_{q,2}\ u^{(1)}_\alpha + \sum_{l=2}^{8} \zeta_{ql}\ u_{i+l-4}\ .$$

These coefficients are then modified according to the 2-D weighting. This can be all performed as a pre-processing step if the body does not move with respect to the grid.

The weighting is computed using an auxiliary scalar grid field χ_{ij}, a mask field that is bookkeeping the number of contributions per ghost grid node. The mask χ_{ij} is first filled with contributions coming from the near-wall ghosts, by looping over the control points. After this loop, the weight for a near-wall ghost q emanating from a particular control point and lying at $\mathbf{x}_{ij}$ may be computed as the arithmetic average $1/\chi_{ij}$ and ζ_{ql} is scaled accordingly. Next, the near-wall ghost coefficients ζ_{ql} $(q = 1, 2)$ are frozen and we repeat the procedure for the extended ghosts ($q = 3$ and $q = 4$). Finally, all ghosts may be computed in a single loop over the control points and by summing all contributions according to the updated coefficients ζ_{ql}.

The weighting approach induces some smoothing of the ghost field. The accuracy of the interpolation may be slightly affected, as well as the enforcement of the wall conditions. This is the price to pay for preserving the uniqueness of the ghost values. As will be shown in the grid convergence study, the order of the error is not affected.

We also point out that the ghost construction implies some restrictions on the body geometry, as on the one hand we need 3 grid points on the flow side of each control point and on the other hand there may still be some "holes" (unassigned ghosts) for non-convex bodies with high local curvature.

6.1.3 Grid convergence study

Similarly to Section 5.2.3, the test case we are validating the present methodology against is the M2P interpolation of a function that is defined outside of a cylinder centered at $\mathbf{x}_c$ and of radius R (and thus of diameter $D = 2R$). In this case, we consider the Laplacian of the Gaussian function $g(x, y, 0)$ (Eq. (5.8)):

$$\nabla^2\omega(x,y) \triangleq \begin{cases} 0 & \text{if } r < R \\ \nabla^2 g(x,y,0) = \dfrac{4\omega_g}{\sigma^2}\left[\dfrac{r_g^2}{\sigma^2} - 1\right]\exp\left(-\dfrac{r_g^2}{\sigma^2}\right) & \text{if } r \geq R \end{cases},$$

with $r \triangleq |\mathbf{x} - \mathbf{x}_c|$ and $r_g \triangleq |\mathbf{x} - \mathbf{x}_g|$ (for the sake of clarity, the time variable $t = 0$ will be omitted from now on). Considering the Laplacian of a Gaussian function stems from the fact that $\nabla^2\omega$ is one of the quantities that have to be interpolated onto the particles in a VPM method. We perform here a comparison between two different setups. In the first setup (**case 1**), the field $(\nabla^2\omega)_{ij}$ and the associated wall data are provided analytically whereas, in the second case (**case 2**), both are computed using (corrected) finite differences based on the prescribed field $\omega_{ij} = g(x_i, y_j)$ and the flux $(\partial\omega/\partial n)_\alpha$ at the control points, as would be the case for a VPM method. The details about these immersed interface computations can be found in Appendix B.

The numerical parameters defining $\nabla^2\omega$ are given in Eq. (5.9). We use again a $(N+1)\times(N+1)$ grid defined in the domain $[-D; D]\times[-D; D]$ (thus again $\Delta x = \Delta y = h$). The extended grid field for the M2P interpolation computed according to Section 6.1.2 and using the analytical field $\nabla^2\omega$ and wall data is shown in Fig. 6.7 for $N = 100$. The ghost field provides indeed a smooth extension across the interface.

In order to validate the interpolation procedure using the ghosts, we define a set of particle positions $\mathbf{x}_p$ resulting from the advection (during one time step Δt_{adv}) of the grid node positions (in the flow domain) with a prescribed velocity field $\mathbf{u}_{\text{adv}} = (\Gamma_{\text{adv}}/2\pi r)\,\hat{\mathbf{e}}_\theta$ corresponding to a purely azimuthal flow with circulation Γ_{adv} (θ is defined with respect to $\mathbf{x}_c$).

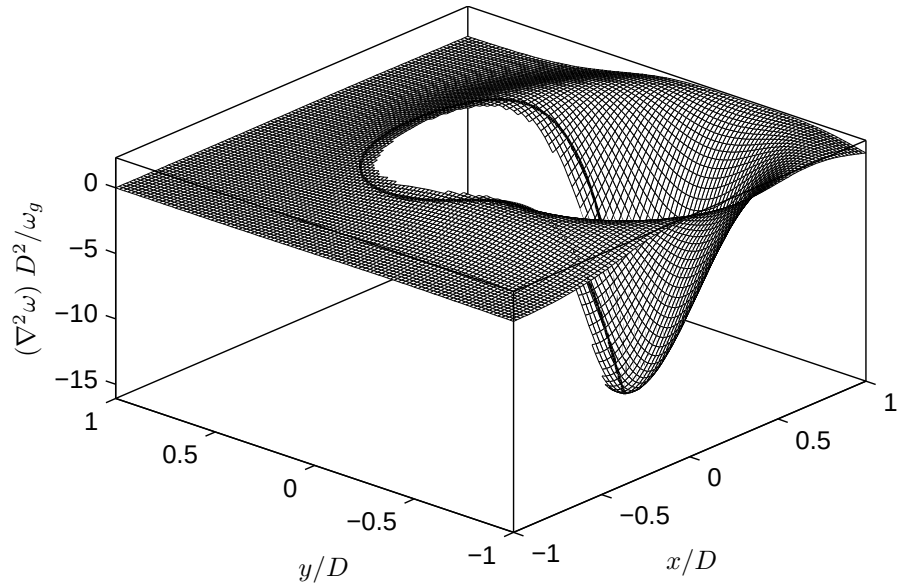


Figure 6.7: *M2P extended field obtained for $\nabla^2\omega$ with $N = 100$. The cylinder boundary is represented by a thick solid line; grid points inside the body which are not ghosts have been omitted.*

The advection time step Δt_{adv} is such that the CFL number

$$\frac{\Delta t_{\text{adv}}}{h}\frac{|\Gamma_{\text{adv}}|}{2\pi R} = 0.5$$

is held fixed for all considered grid resolutions N. The resulting particle positions are shown in Fig. 6.8. We thus ensure that the particle positions with respect to the grid remain similar for all grid resolutions N. Moreover, all particles having a non-zero velocity, none of the particles coincide with grid nodes; otherwise the interpolation would have been trivial at those points and the computed error biased. Moreover, the orientation of the particle position vector with respect to the associated grid node is uniformly distributed over all particles.

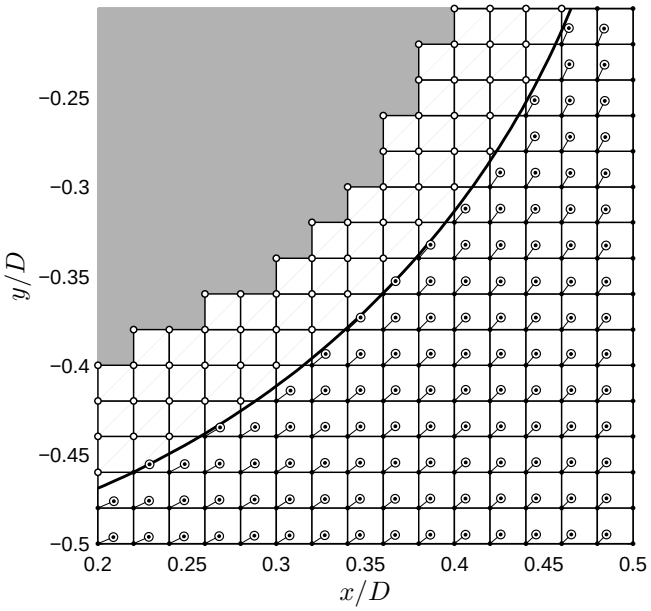
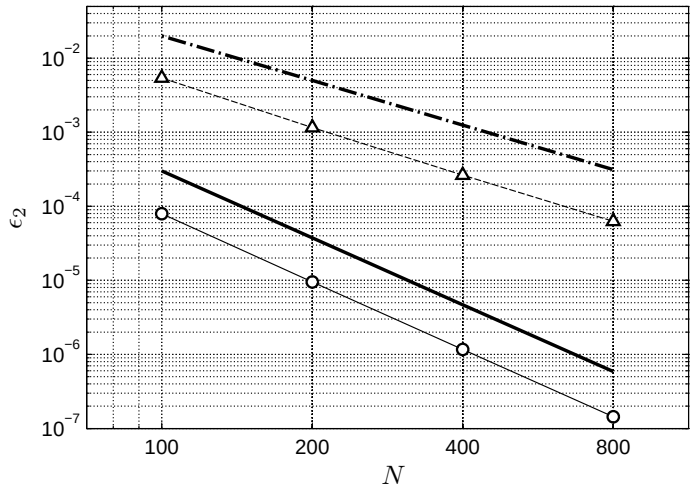


Figure 6.8: *Sketch of the problem setup for the validation of the 2D M2P interpolation scheme in the case of a cylinder and at a grid resolution $N = 100$: M2P ghosts inside the body (open circles); flow grid nodes (bullets); particle positions (dotted circles) with their associated advection path (thin lines).*

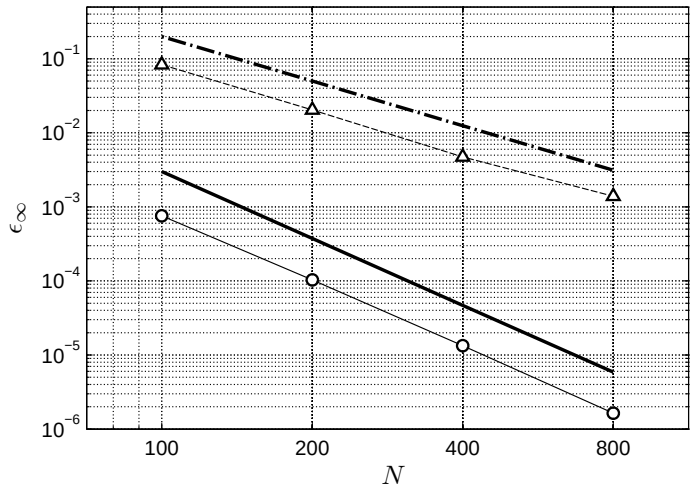
The following error norms are defined over the set of particles

$$\begin{aligned} \epsilon_2 &\triangleq \|\epsilon\|_2 \triangleq \frac{hD}{\omega_g} \left(\sum_p (\widetilde{(\nabla^2\omega)}_p - (\nabla^2\omega)_p)^2 \right)^{\frac{1}{2}} \\ \epsilon_\infty &\triangleq \|\epsilon\|_\infty \triangleq \frac{D^2}{\omega_g} \max_p \left| \widetilde{(\nabla^2\omega)}_p - (\nabla^2\omega)_p \right| , \end{aligned}$$

with $\widetilde{(\nabla^2\omega)}_p$ and $(\nabla^2\omega)_p$ respectively the interpolated field and the analytical field $\nabla^2\omega$ evaluted at $\mathbf{x}_p$. The error norms are reported in Fig. 6.9 and exhibit a third order convergence rate for **case 1**. This is expected, as the interpolation kernel M_4' enjoys the same convergence properties. The rate of convergence for the L_2-error norm is 3.01 (between $N = 400$ and $N = 800$) and 3.02 for the L_∞-norm. **Case 2** exhibits a second order convergence (2.07 for the L_2-error norm and 1.76 for the L_∞-norm), with a higher error level.



(a)



(b)

Figure 6.9: *Grid convergence study for the M2P interpolation procedure: (a) L_2-error norm and (b) L_∞-error norm. The error norm is represented by a thin solid line with "$\circ$"-signs for* ***case 1*** *and a thin dashed line with "Δ"-signs for* ***case 2****; the thick solid line shows a third order slope and the thick dash-dotted line shows a second order slope.*

6.2 Particle-to-mesh interpolation (P2M)

A fundamental difference with the M2P operation is that the P2M scheme intrinsically enlarges the vorticity support as it explicitly affects two grid nodes on each side of the particles, whereas the M2P simply collects data onto the existing particles. Following the philosophy of the M2P approach, we might as well think of the P2M procedure as an interpolation. We again choose to carry the point information ω on a particle, as opposed to the previously defined intensity α. Hence we relax the exact conservation of moments and again focus on the local accuracy of the remeshed field, which should eventually provide the necessary smoothness to operate the spatial differentiation required for the VPM method. Moreover, the targeted convergence of the method in space should likewise imply the convergence of the computed moments for a vanishing grid spacing.

The aim is to keep the method as close as possible to the initial approach, i.e. by keeping the loop over the particles and thus avoiding to identify the set of particles near a grid point susceptible to contribute to the interpolation. We consider here two approaches: one is based on the extension procedure enabled by level set-based methods and interface-tracking algorithms [113, 2, 99] (see Section 6.2.1) and the other one suggests "inverting" the M2P approach from Section 6.1.2 (see Section 6.2.2). The first approach will be used for the remainder of this work in the framework of VPM methods, as the second methodology, while promising and more general, still needs some stabilization in order to converge for all cases.

6.2.1 Wall data extension approach

Similarly to the M2P operation, we wish to straightforwardly apply the classical redistribution scheme Eq. (6.3), after the pre-computation of a ghost field. In the present case, we need a set of ghost *particles*, akin to the previous grid ghosts. The difficulty resides here in the fact that, in addition to the value of these ghosts, we also need to provide their position, that may in general differ from the associated grid nodes. This choice is all the more so important, that the required smoothness of the regridded field is closely linked to the distribution of the particles in space, as was already mentioned in the introduction of this chapter. The information about the positions may be extracted from the surrounding flow, whereas the value of the ghosts is related to the prescribed

wall data, again similarly to the M2P approach. By comparison, the "image" particles used in [83] may be considered as ghost particles whose position is determined by symmetry of the original particle's position with respect to the wall.

Taking into account the flow particles for the ghost construction is challenging, due to their (possibly) arbitrary positions. In the present approach we therefore restrict the information for the ghost computation to the wall data only. Anticipating the type of information that is provided at the wall by a VPM method, namely the vorticity flux (i.e. $(\partial\omega/\partial n)_\alpha$ with the normal $\mathbf{n}$ pointing outwards of the body), we opt for a linear extension of the vorticity along the normal to the body boundary. This can be related to what was done for M2P where a "quadratic" extension was chosen. Now assume that we also know the vorticity value ω_α at the wall point $\mathbf{x}_\alpha$. Given the signed distance function $\phi(\mathbf{x})$ (i.e., the level set; it is negative in the body), the expression for the extension reads

$$\omega^{\text{ext}}(\mathbf{x}) \triangleq \omega_\alpha + \phi(\mathbf{x}) \left(\frac{\partial\omega}{\partial n}\right)_\alpha , \tag{6.8}$$

with $\mathbf{x}_\alpha = \mathbf{x} - \phi(\mathbf{x})\ \mathbf{n}$ the projection of $\mathbf{x}$ onto the boundary and $\mathbf{n}$ the normal vector emanating from $\mathbf{x}_\alpha$. Basically, if we further know the position $\mathbf{x}_p^*$ of the ghost particles inside the body, we may use the combined set of real particles ω_q and ghost particles $\omega_p^* \triangleq \omega^{\text{ext}}(\mathbf{x}_p^*)$ to provide the support for a P2M interpolation onto the grid.

This type of extension was also used in [75]. There, the wall point $\mathbf{x}_\alpha$ is computed geometrically using the orthogonal projection in spherical coordinates. The present computation of the extended field is based on the work of Peng et al. [99], which is more appropriate to be combined with the immersed interface framework, as no projection is required. Let us first consider an extension field $q(\mathbf{x})$ that is constant along the normal, resulting thus in $q(\mathbf{x}) = q_\alpha$ with $\mathbf{x}_\alpha = \mathbf{x} - \phi(\mathbf{x})\ \mathbf{n}$ and q_α the associated wall data (in fact, Peng et al. [99] do not need a linear extension but, based on their approach, we will show how to adapt the method to provide it). The field q is the steady state solution (in a bounded domain) of the following Hamilton-Jacobi equation :

$$\frac{\partial q}{\partial \tau} + S(\phi)\ \nabla\phi \cdot \nabla q = 0 , \tag{6.9}$$

with τ a pseudo time variable and $S(\phi)$ the signature function defined as

$$S(\phi) = \begin{cases} -1 & \text{if } \phi < 0 \,, \\ 0 & \text{if } \phi = 0 \,, \\ 1 & \text{if } \phi > 0 \,. \end{cases}$$

The characteristics of this hyperbolic equations are straight lines normal to the body boundary, as $\nabla\phi = \mathbf{n} \triangleq (n^x, n^y)$ can also be defined away from the boundary (ϕ being a distance function we also have that $|\nabla\phi| = 1$). The "advection velocity" is thus simply $S(\phi)\,\mathbf{n}$ and it points away from the interface. This equation only requires a boundary condition on the interface, which is precisely the prescribed data q_α.

Eq. (6.9) is solved on the grid in a narrow band domain $\mathcal{B}$ defined by $-d_\mathrm{b} \leq \phi(\mathbf{x}) \leq d_\mathrm{f}$ enclosing the interface (the width of this band is a few h and will be determined later on, see Fig. 6.11). To that end, we furthermore need to define a regularized signature function on the grid:

$$s_{ij} = \frac{\phi_{ij}}{\sqrt{\phi_{ij}^2 + h^2}} \,.$$

This modification solely affects the norm of the effective advection velocity and therefore slightly increases the time for the wall information to propagate. We do not seek to perform a high order time discretization here, as the equation is meant to be solved iteratively until a steady state is reached. The numerical scheme suggested by Peng et al. [99] (first order upwind scheme with an Euler time integration) is adapted so as to account for the information provided at the control points $\mathbf{x}_\alpha$.

The Hamilton-Jacobi Eq. (6.9) is discretized as

$$\begin{aligned} q_{ij}^{n+1} = q_{ij}^n - \Delta\tau \Bigg[& \left(s_{ij} n_{ij}^x\right)^+ \left(\frac{\partial q}{\partial x}\right)_{ij}^{\oplus} + \left(s_{ij} n_{ij}^x\right)^- \left(\frac{\partial q}{\partial x}\right)_{ij}^{\ominus} \\ & + \left(s_{ij} n_{ij}^y\right)^+ \left(\frac{\partial q}{\partial y}\right)_{ij}^{\oplus} + \left(s_{ij} n_{ij}^y\right)^- \left(\frac{\partial q}{\partial y}\right)_{ij}^{\ominus} \Bigg] \,, \end{aligned}$$

where $\Delta\tau$ is the pseudo time step, $(x)^+ \triangleq \max(x, 0)$ and $(x)^- \triangleq \min(x, 0)$. Typically $\Delta\tau = 0.5h$, as the advection velocity is smaller or equal to 1.

The spatial derivatives of q at $\mathbf{x}_{ij} = (x_i, y_j)$ are computed as follows:

$$\left(\frac{\partial q}{\partial x}\right)_{ij}^{\oplus} = \begin{cases} \dfrac{q_{i,j} - q_\alpha}{x_i - x_\alpha} & \text{if } \mathbf{x}_{ij} \text{ is irregular ,} \\ \dfrac{q_{i,j} - q_{i-1,j}}{\Delta x} & \text{otherwise ,} \end{cases}$$

$$\left(\frac{\partial q}{\partial x}\right)_{ij}^{\ominus} = \begin{cases} \dfrac{q_\alpha - q_{i,j}}{x_\alpha - x_i} & \text{if } \mathbf{x}_{ij} \text{ is irregular ,} \\ \dfrac{q_{i+1,j} - q_{i,j}}{\Delta x} & \text{otherwise ,} \end{cases} \tag{6.10}$$

$$\left(\frac{\partial q}{\partial y}\right)_{ij}^{\oplus} = \begin{cases} \dfrac{q_{i,j} - q_\alpha}{y_j - y_\alpha} & \text{if } \mathbf{x}_{ij} \text{ is irregular ,} \\ \dfrac{q_{i,j} - q_{i,j-1}}{\Delta y} & \text{otherwise ,} \end{cases}$$

$$\left(\frac{\partial q}{\partial y}\right)_{ij}^{\ominus} = \begin{cases} \dfrac{q_\alpha - q_{i,j}}{y_\alpha - y_j} & \text{if } \mathbf{x}_{ij} \text{ is irregular ,} \\ \dfrac{q_{i+1,j} - q_{i,j}}{\Delta y} & \text{otherwise .} \end{cases}$$

Note that the grid point $\mathbf{x}_{ij}$ is here called "irregular" when the distance to the nearest control point *along the direction corresponding to the derivative of* q is smaller than the grid space h.

Since, the evaluation of the derivatives in Eq. (6.10) is ill-posed when $|x_i - x_\alpha|$ or $|y_j - y_\alpha|$ is small, we further define the switch parameter ε that delimits a near-wall region $|\phi|/h \leq \varepsilon$ from the rest of the computational domain $|\phi|/h > \varepsilon$. Inside this near-wall region, all nodes are irregular if $\varepsilon < 1$ (contrary to above, the terminology for an "irregular" node here corresponds to the one given in Chapter 4, i.e. $\mathbf{x}_{ij}$ is irregular when the distance to the nearest control point *in any grid direction* is smaller than the grid size h). In this area, the Hamilton-Jacobi Eq. (6.9) is not solved in order to prevent the ill-posed configurations in Eq. (6.10). Instead, we simply enforce the boundary condition as $q_{ij} = q_\alpha$, where $\mathbf{x}_\alpha$ is the nearest control point. The justification for the approximation $q_{ij} = q_\alpha$ stems from the fact that the switch parameter is chosen so as to fulfill $\varepsilon << 1$, in practice.

As an initial condition we set $q_{ij} = 0$, except for irregular grid nodes, where we again impose $q_{ij} = q_\alpha$ corresponding to the boundary value of the nearest control point $\mathbf{x}_\alpha$ it is surrounded by. The extension field $\omega^{\text{ext}}(\mathbf{x})$ is constructed by applying the above methodology to $(\partial\omega/\partial n)_\alpha$ and ω_α individually, obtaining thus $\omega_\alpha^{\text{ext}}(\mathbf{x})$ and $(\partial\omega/\partial n)_\alpha^{\text{ext}}(\mathbf{x})$.

The combination (see Eq. (6.8)) is then performed as

$$\omega^{\text{ext}}(\mathbf{x}) = \omega_\alpha^{\text{ext}}(\mathbf{x}) + \phi(\mathbf{x}) \left(\frac{\partial \omega}{\partial n} \right)_\alpha^{\text{ext}} (\mathbf{x}) \ .$$

Hence we do not need to find $\mathbf{x}_\alpha$ by computing a projection onto the boundary, contrary to [75].

It should be observed that the switch parameter ε has to be chosen carefully. It indeed controls the $\mathcal{O}(\varepsilon h)$ error that is made locally on q when assigning $q_{ij} = q_\alpha$ in the near-wall region $|\phi|/h \leq \varepsilon$. If the tangential gradient of q at the wall is significant, the more the grid line differs from the normal line to the body boundary emanating from the control point, the bigger the error will be, unless ε is small enough. Yet, the value of ε must not be too small, as $|\mathbf{x}_{ij} - \mathbf{x}_\alpha| \geq |\phi(\mathbf{x}_{ij})| > \varepsilon h$ may become very small while performing the numerical differentiation in Eq. (6.10). In any case, the nodes in the near-wall region (and thus affected by the error) are only a subset of all irregular nodes (their number scales with $\varepsilon L/h$, with L the perimeter of the body) and the global convergence of the L_2-error norm should not be significantly affected, as opposed to the L_∞ norm whose convergence might deteriorate when h is small. From a practical point of view, the value $\varepsilon = 10^{-2}$ seems to provide satisfying results in terms of convergence, even if $\varepsilon \sim h/L$ is required for formal convergence.

The extension approach is validated using the same numerical setup as for the M2P interpolation. The wall data to be extended is provided by evaluating the Gaussian function $\omega(x, y) = g(x, y, 0)$ (and its normal derivative) from Eq. (5.8) on the cylinder boundary from Section 6.1.3. The resulting field $\omega^{\text{ext}}(\mathbf{x})$ obtained after 20 pseudo time steps is shown in Fig. 6.10 for $N = 100$. The extension is also performed on the flow side. The reason for this shall become clear hereafter, when discussing the positions $\mathbf{x}_p^*$ of the ghost particles, that also have to provided.

The choice for the ghost particle positions $\mathbf{x}_p^*$ is nearly arbitrary, yet it must satisfy some conditions. First, notice that placing the ghost particles at the grid nodes amounts to simply ignore their existence, as the redistribution of a grid node is the identity (by application of Eq. (6.2) with $\xi = 0$) and thus does not affect any of the grid nodes in the flow domain. Secondly, the set of particles should be such that every grid point in the flow domain sees 16

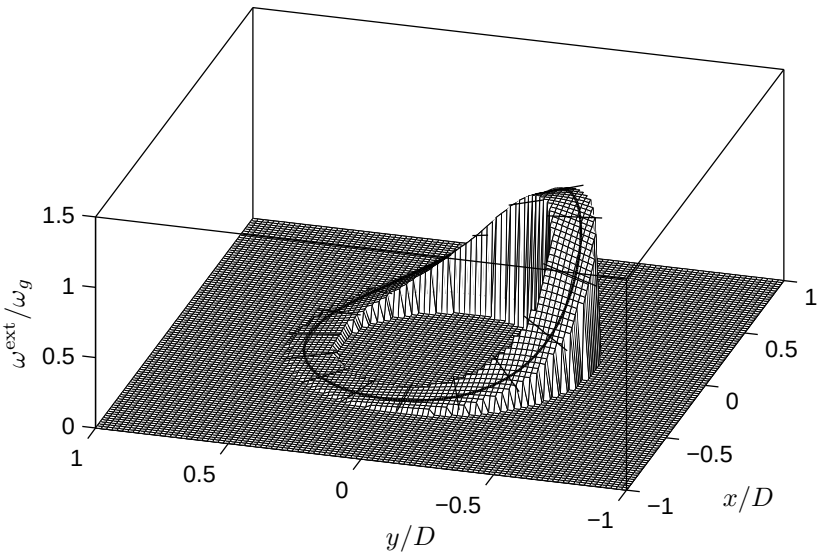


Figure 6.10: *P2M wall data extension ω^{ext} obtained after 20 pseudo time steps for $N = 100$. The cylinder boundary is represented by a thick solid line; the exact extension is also shown at some points (thin solid lines).*

particles (real or ghost) in its M_4' interpolation range, in 2-D (see Fig. 6.11). This condition actually corresponds to the absence of clustering and depletion of particles (the partition of unity should be preserved). On the one hand, this is ensured precisely by the use of regular redistribution, and on the other hand, the smoothness of the flow velocity preserves the well-ordered spatial configuration of the particles for short times.

This observation suggests using an extension of the velocity field inside the body in order to advect the ghost particles in the same way as are the flow particles (see Fig. 6.11). Referring to the previous Section 6.1, such an extension is provided by applying the M2P ghost computation to the velocity field (this is anyway required for the advection of the flow particles in a VPM method).

In summary, the P2M algorithm consists of the following steps (see also Fig. 6.11):

1. **Ghost particle advection**: during the advection of the flow particles, use the M2P grid extension of the velocity field $\mathbf{u}^{\mathrm{ext}}$ to also move the ghost particles inside the body. Hence the ghost particle position is determined by $d\mathbf{x}_p^*/dt = \mathbf{u}^{\mathrm{ext}}(\mathbf{x}_p^*)$ (the ghost particles are represented by open circles in Fig. 6.11). Note that $\mathbf{u}^{\mathrm{ext}}$ needs to be interpolated onto the ghost particles, too (M2P).

2. **Wall data extension**: solve Eq. (6.9) for $\omega_\alpha^{\text{ext}}$ and $(\partial\omega/\partial n)_\alpha^{\text{ext}}$ to obtain the extended vorticity field ω_{ij}^{ext} in the narrow band domain $\mathcal{B}$ (delimited by dotted lines in Fig. 6.11; the light grey colored area shows all affected grid nodes).

3. **M2P interpolation**: $\omega_p^* = \omega^{\text{ext}}(\mathbf{x}_p^*)$ is obtained by interpolating ω_{ij}^{ext} at $\mathbf{x}_p^*$ using the 2-D M_4' interpolation scheme. The required width of the narrow band domain $\mathcal{B}$ is then $-4\Delta l \leq \phi(\mathbf{x}) \leq 2\Delta l$ with $(\Delta l)^2 \triangleq (\Delta x)^2 + (\Delta y)^2$.

4. **P2M interpolation**: The particles $\mathbf{x}_p^*$ provide the necessary support for the P2M interpolation that maps ω_q and ω_p^* onto the grid in order to finally obtain $\tilde{\omega}_{ij}$ (real particles are represented as dotted circles in Fig. 6.11; the red bullet shows a particular grid node along with its interpolation support consisting of red colored real and ghost particles). The P2M interpolation is only performed on grid nodes of the flow domain.

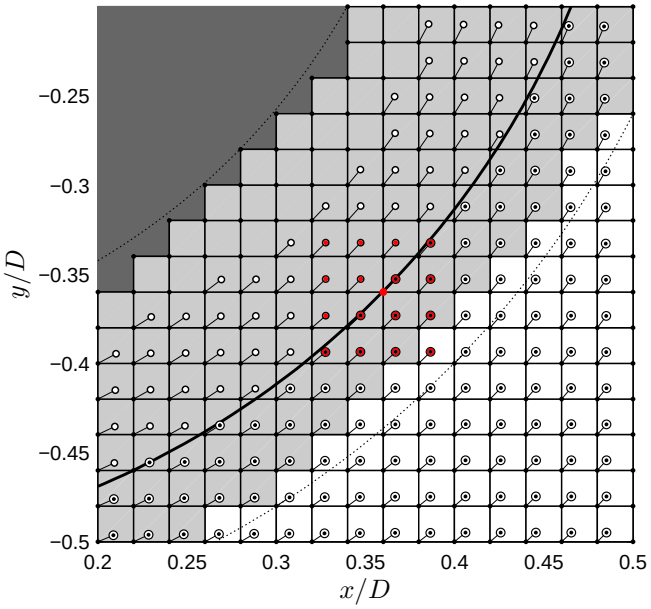


Figure 6.11: *Sketch of the P2M interpolation scheme in the case of a cylinder and at a grid resolution $N = 100$: grid nodes (small bullets); real particles (dotted circle); P2M ghost particles (open circles); body boundary (thick solid line); wall extension domain $\mathcal{B}$ (delimited by dotted lines) with affected grid nodes (light grey colored area); a particular grid node (red bullet) with its interpolation support (red colored real and ghost particles).*

The global P2M approach is validated by considering the same advection field as Section 6.1.2, namely $\mathbf{u}_{\text{adv}} = (\Gamma_{\text{adv}}/2\pi r)\,\hat{\mathbf{e}}_\theta$, along with the same CFL condition on Δt_{adv}. This velocity field is then extended using the tools of Section 6.1.2. The flow particles are initialized with $\omega_q = \omega(x_q, y_q)$ and the following error norms are defined

$$\begin{aligned} \epsilon_2 &\triangleq \|\epsilon\|_2 \triangleq \frac{h}{\omega_g D}\left(\sum_{i,j}(\tilde{\omega}_{ij} - \omega(x_i, y_j))^2\right)^{\frac{1}{2}} \\ \epsilon_\infty &\triangleq \|\epsilon\|_\infty \triangleq \frac{1}{\omega_g}\max_{i,j}|\tilde{\omega}_{ij} - \omega(x_i, y_j)| \ . \end{aligned}$$

The grid convergence study results are shown in Fig. 6.12 (with 50 pseudo time steps). Four cases are considered here:

- **Case 1** : $\varepsilon = 0.5$ with analytically provided $(\partial\omega/\partial n)_\alpha$ and ω_α.
- **Case 2** : $\varepsilon = 10^{-2}$ with analytically provided $(\partial\omega/\partial n)_\alpha$ and ω_α.
- **Case 3** : $\varepsilon = 10^{-2}$ with analytically provided $(\partial\omega/\partial n)_\alpha$, but ω_α is computed using the compatible extrapolation from Eq. (5.7), as would be performed by a VPM method.
- **Case 4** : placing the ghost particles exactly at the grid nodes (hence they do not affect any grid node in the flow domain).

The rate of convergence for the L_2-error norm is 2.04 (between $N = 400$ and $N = 800$) and 2.01 for the L_∞-norm for **case 2** and **case 3**. These cases nearly coincide, which is due to the fact that the compatible extrapolation scheme to compute ω_α is $\mathcal{O}(h^4)$ and thus very accurate. Setting $\varepsilon = 0.5$ (**case 1**) significantly deteriorates the rate of convergence for the L_∞-norm (1.26), whereas the L_2-error norm (1.75) is less affected, as previously claimed. For **case 4**, the rate of convergence for the L_2-error norm is 0.49 and the L_∞-norm does not converge at all (rate 0.00). This shows how a change in the ghost particle position tremendously affects the convergence and the error level.

We do not recover the third order convergence of the M_4' interpolation kernel (even for **case 2** and **case 3**), since we use a linear extension along the normal, as opposed to the quadratic extension from the M2P approach. Yet, we may hardly increase the order of the approach, unless higher order derivatives are also extended along the normal. The expression of these derivatives would require computing cross derivatives along both grid directions at the control

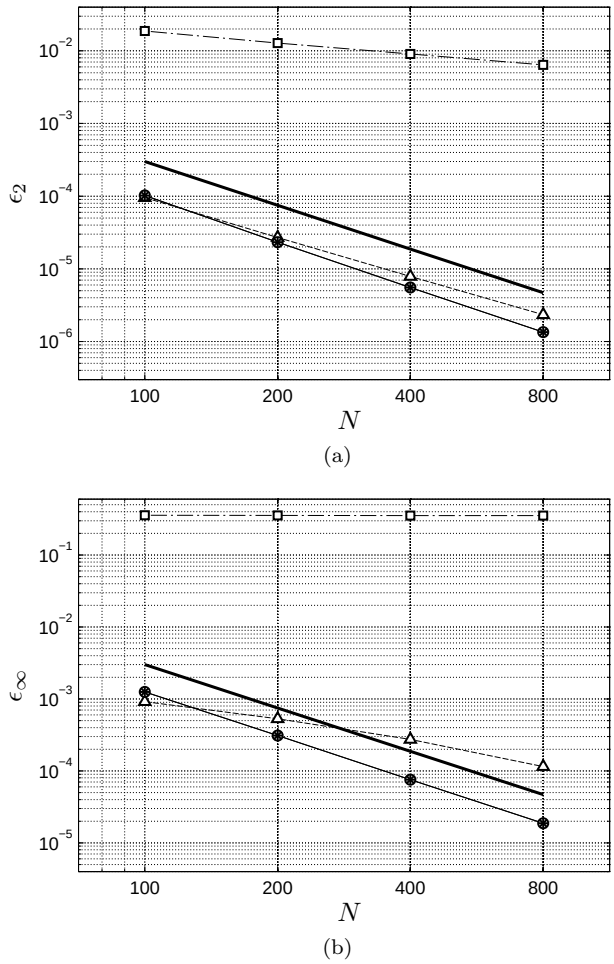


Figure 6.12: *Grid convergence study for the P2M interpolation procedure: (a) L_2-error norm and (b) L_∞-error norm. The error norms are shown for* ***case 1*** *(thin dashed line with "Δ"-signs),* ***case 2*** *(thin solid line with "$\circ$"-signs),* ***case 3*** *(thin solid line with "$*$"-signs) and* ***case 4*** *(thin dash-dotted line with "$\square$"-signs); the thick solid line shows a second order slope.*

points. The advection term in the Hamilton-Jacobi Eq. (6.9) could also be computed using an upwind space discretization of higher order, e.g. a WENO scheme [115].

It should also be noted that this approach does not guarantee the incompressibility of the velocity field inside the body. Enforcing this explicitly would require solving a Poisson equation inside the body with a no-through flow condition at the wall. This is actually exactly the same framework as for the outer

flow computation, and the tangential velocity would thus not necessarily fit the prescribed wall velocity. Again, one may consider diffusing the resulting slip velocity inside the body similarly to the outer flow computation and hence obtain a smooth and incompressible extension of the velocity. All these steps could be coupled to the outer flow computation.

Nevertheless, the above suggestion of improvement is maybe superfluous, as the present approach still provides a ghost velocity field that is smooth in practice and "follows the trends" of the outer flow field. This feature is of great importance when considering that the ghost particle offsets with respect to the grid should behave in a continuous manner for neighboring particles, so as to provide a proper interpolation support for the grid points and hence avoid a local convergence breakdown, as was observed for **case 4** in Fig. 6.12.

6.2.2 Alternative approach

First, it should be observed that - even in an unbounded domain - P2M is not the inverse of M2P, in the sense that interpolating a grid field to particles and consequently remapping the information onto the grid does not lead to the initial field (e.g. consider a discrete Dirac function defined on the grid and a set of uniformly spaced particles with a non-zero offset with respect to the grid).

Nevertheless, one could alternatively try to construct the P2M procedure by pursuing this inversion idea, while allowing the addition of an unknown artificial correction to the particles. Briefly, these corrections should be such that the application of the original P2M scheme using the corrected particles provides a remeshed grid field giving rise to the uncorrected set of particles through the application of the M2P approach from Section 6.1.2. The resulting grid field would then be the intended solution that satisfies the prescribed conditions at the wall, by construction of the grids ghost that are implicitly required by the M2P operation. The procedure thus does not strictly invert the M2P, but rather provides a self-consistent tool for the remeshing operation. Iterations are however required as the problem is global.

The algorithm can be summarized as follows in 1-D (see also Fig. 6.13):

- **Initialization** ($k = 0$): we have the set of real particles ω_p, a set of particle corrections $\Delta\omega_p^k = 0$ (in the flow) and a set of ghost particles $\Delta\hat{\omega}_p^k = 0$ (in the body).

- **Iteration**:

  1. Compute the grid field ω_j^k using the (uncorrected) P2M approach in the flow domain, based on the particles $(\omega_p + \Delta\omega_p^k)$ and $\Delta\hat{\omega}_p^k$.
  2. Compute the M2P grid ghosts for ω_j^k (using the associated wall data), according to Section 6.1.2.
  3. Apply the M2P scheme to ω_j^k and its ghosts, the result being ω_p^k (in the flow) and $\hat{\omega}_p^k$ (in the body).
  4. Compute the new corrections as $\Delta\omega_p^{k+1} = \Delta\omega_p^k + (\omega_p - \omega_p^k)$ and $\Delta\hat{\omega}_p^{k+1} = \hat{\omega}_p^k$.
  5. $k \leftarrow k+1$ and go back to step 1, unless the convergence for the error $|\omega_p^k - \omega_p|$ has been met.

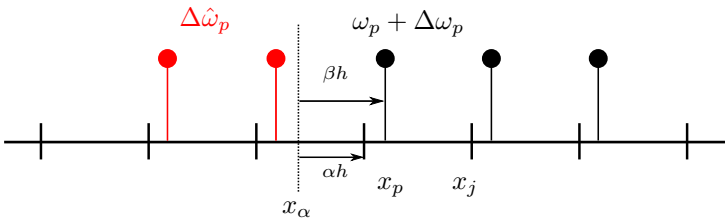
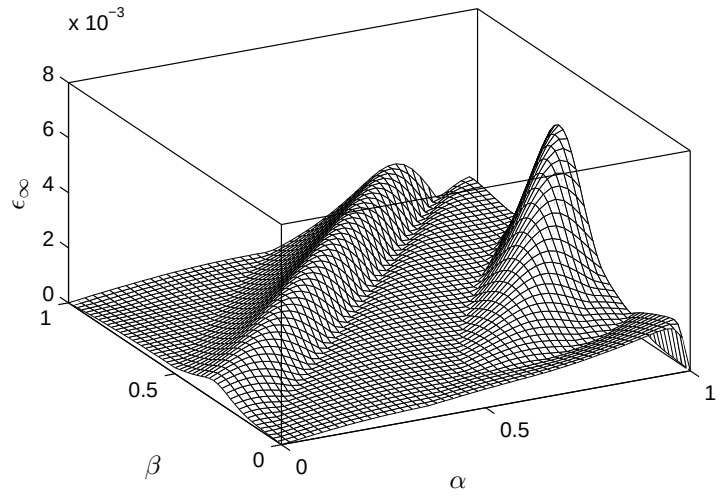


Figure 6.13: *Sketch of the alternative P2M approach.*

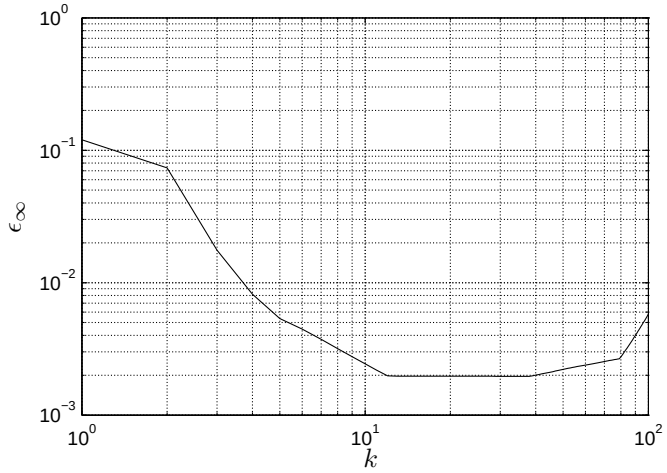
The algorithm actually solely iterates over a few grid points near the wall, the remainder of the grid nodes is pre-computed using the uncorrected P2M procedure and frozen thereafter. Moreover, we overwrite the first grid node value with an interpolation based on the wall data and the value at some other nodes, so as to stabilize the iteration as much as possible.

Some preliminary tests show that this approach converges, except for some pathological configurations of the grid nodes' and particles' positions with respect to the grid (α and β). As shown in Fig. 6.14, the error diverges after tens of iterations.

A thorough analysis of the algorithm would maybe enlighten the reasons for this behavior. The advantage of the algorithm is that it is more prone to be extended to higher order methods, as it is built in a complementary fashion upon the M2P approach. Its stabilization would provide an interesting tool for the reciprocal interpolation between the grid and the particles.



(a)



(b)

Figure 6.14: *Preliminary tests for the alternative P2M approach: (a) L_∞-error norm at $k = 100$ for different α (offset of the grid with respect to the interface) and β (offset of the particles with respect to the interface); (b) Maximum L_∞-error norm observed for all (α, β) as a function of the iteration index k.*

Chapter 7

Immersed interface vortex particle-mesh solver

The aim of this chapter is to gather the tools that have been developed in Chapter 4, 5 and 6, and to combine them into an immersed interface-enabled vortex particle-mesh solver.

First, the remaining required numerical ingredients are detailed in Section 7.1, along with the description of the time stepping algorithm. Secondly, some results are given for the well-documented test case of the impulsively started flow past a circular cylinder (Section 7.2.1). Next, the ability of the solver to simulate the flow past an airfoil (Section 7.2.2) and the vortex shedding induced by a cylinder (Section 7.2.3) are also demonstrated. As a perspective of future development, Section 7.3 presents a possible way to account for an outflow condition. Finally, some conclusions are drawn in Section 7.4.

7.1 Time stepping algorithm

Similarly to what has been done in Chapter 3, we describe here the algorithm used in order to solve the 2-D Navier-Stokes equations in vorticity-velocity formulation, using the previously developed immersed interface tools. The numerical framework and the associated notations follow those of Chapter 3. We focus here on the treatment of the unbounded domain and of the inner solid body boundaries, as opposed to Chapter 3 where no body was present inside the domain and the outer boundaries of the computational domain were solid walls.

Let us consider the computational domain defined by $\Omega_{comp} \triangleq [x_L, x_R] \times [y_B, y_T]$ with an associated cell-centered $M \times N$ grid of mesh size h (see Fig. 3.2). For the sake of simplicity, the presence of a single and non-moving body is assumed here, its boundary being described by $\partial\Omega_b$ (hence $\mathbf{u} = \mathbf{u}_b = 0$ on $\partial\Omega_b$). Moreover, we consider a constant free stream velocity field $\mathbf{U}_\infty$. The set of particles $\mathbf{x}_p$ carrying a vorticity ω_p is initialized at the grid node positions $\mathbf{x}_{ij}$, thus particles also exist inside the body (yet, their vorticity is zero), following the lines of the "immersed" interface/boundary approach. As a reminder, the Lagrangian evolution of the particles is prescribed by the following ordinary differential equations

$$\begin{aligned} \frac{d\mathbf{x}_p}{dt} &= \mathbf{u}_p \\ \frac{d\omega_p}{dt} &= \nu \left(\nabla^2\omega\right)_p \ . \end{aligned}$$

The time integration is again performed using a mid-point rule Runge-Kutta 2 scheme and the right-hand sides are computed on the grid, using (corrected) finite differences. It is based on the DRK2-SUB time-stepping algorithm from Appendix E. Yet, as the studies from Chapter 3 showed that all discussed time integration algorithms (for the no-slip enforcing procedure) are first order in time and provide a similar error level, any other choice would have been equivalent. The required modifications are minor, since the computational operations are identical and solely their chronology changes, basically.

The subscript g for a field f (with $f = 0$ inside the body) indicates a "ghost augmented" field f_g, i.e. f is complemented with a set of ghost values inside the body in the vicinity of the boundary (the ghosts can be nodes or particles depending on the nature of the discretization of f, that is f_{ij} or f_p). Furthermore, some wall data (required for the particle-mesh interpolation, according to Chapter 6 and Appendix B) is defined over the control points $\mathbf{x}_\alpha$ at the time t^n

$$\begin{aligned} \mathcal{U}_\alpha^n &\triangleq \left\{ \left(\mathbf{u}_\alpha \ , \ \frac{\partial \mathbf{u}}{\partial \xi}\bigg|_\alpha \right) \ \middle| \ \forall \mathbf{x}_\alpha \in \partial\Omega_b \ , \ t = t^n \right\} \\ \mathcal{L}_\alpha^n &\triangleq \left\{ \left(\left(\nabla^2\omega\right)_\alpha \ , \frac{\partial}{\partial \xi}\left(\nabla^2\omega\right)_\alpha \right) \ \middle| \ \forall \mathbf{x}_\alpha \in \partial\Omega_b \ , \ t = t^n \right\} \\ \mathcal{W}_\alpha^n &\triangleq \left\{ \left(\omega_\alpha \ , \ \frac{\partial \omega}{\partial n}\bigg|_\alpha \right) \ \middle| \ \forall \mathbf{x}_\alpha \in \partial\Omega_b \ , \ t = t^n \right\} \ , \end{aligned}$$

with ξ the current direction defined in Chapter 5 and n the normal direction.

The velocity values at the wall $\mathbf{u}_\alpha$ should be equal to zero at convergence; however the residual slip is not fully canceled at a finite resolution and, as a consequence, $\mathbf{u}_\alpha$ is computed and taken into account during the velocity M2P interpolation, for consistency reasons.

The following scheme summarizes the adopted integration strategy and some of the computational steps requiring a specific attention are described thereafter:

Initialization :

1. Compute the signed distance function to the body (level set).
2. Compute the control points.
3. Precompute the immersed interface stencils : Poisson equation, vortex sheet evaluation, diffusion and ghost weights for the M2P interpolation.
4. Construct the tree for the multipole method that is used in the unbounded iteration of the Poisson solver.
5. Assemble the matrix for the Poisson solver.

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*

$$\omega_{ij}^n \xrightarrow{\text{Poisson}} \mathbf{u}_{ij}^n,\, \mathcal{U}_\alpha^n$$
$$\mathbf{u}_{ij}^n,\, \mathcal{U}_\alpha^n \xrightarrow{\text{ghosts for M2P}} (\mathbf{u}_g)_{ij}^n$$
$$(\mathbf{u}_g)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} (\mathbf{u}_g)_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,(\mathbf{u}_g)_p^n$$

- *Diffusion :*

$$\omega_{ij}^n \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^n \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_\alpha^n = 0$$
$$\omega_{ij}^n\,, \left(\nabla^2\omega\right)_{ij}^n\,, \left.\frac{\partial\omega}{\partial n}\right|_\alpha^n = 0 \xrightarrow{\text{wall data}} \mathcal{L}_\alpha^n\,, \mathcal{W}_\alpha^n$$
$$\left(\nabla^2\omega\right)_{ij}^n,\, \mathcal{L}_\alpha^n \xrightarrow{\text{ghosts for M2P}} \left(\nabla^2\omega_g\right)_{ij}^n$$
$$\left(\nabla^2\omega_g\right)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2},*} = \omega_p^n + \frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2},*},\, \mathbf{x}_p^{n+\frac{1}{2}},\, \mathcal{W}_\alpha^n \xrightarrow{\text{ghosts for P2M}} (\omega_g)_p^{n+\frac{1}{2},*}$$
$$(\omega_g)_p^{n+\frac{1}{2},*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2},*}$$

- *Near-wall diff. :*

$$\omega_{ij}^{n+\frac{1}{2},*} \xrightarrow{\text{Poisson}} \mathbf{u}_{ij}^{n+\frac{1}{2},*},\, \Delta\gamma_\alpha^{n+\frac{1}{2},*}$$
$$\bar{q}_\alpha^n = \frac{\Delta\gamma_\alpha^{n+\frac{1}{2},*}}{\Delta t/2} \xrightarrow{\text{SI diff.}} (\omega_w)_{ij}^{n+\frac{1}{2}}$$
$$\omega_{ij}^{n+\frac{1}{2}} = \omega_{ij}^{n+\frac{1}{2},*} + (\omega_w)_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*

$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{Poisson}} \mathbf{u}_{ij}^{n+\frac{1}{2}},\, \mathcal{U}_\alpha^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}},\, \mathcal{U}_\alpha^{n+\frac{1}{2}} \xrightarrow{\text{ghosts for M2P}} (\mathbf{u}_g)_{ij}^{n+\frac{1}{2}}$$
$$(\mathbf{u}_g)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} (\mathbf{u}_g)_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\, (\mathbf{u}_g)_p^{n+\frac{1}{2}}$$

- *Diffusion :*

$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu \left.\frac{\partial\omega}{\partial n}\right|_\alpha^{n+\frac{1}{2}} = 0$$
$$\omega_{ij}^{n+\frac{1}{2}}, \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}}, \left.\frac{\partial\omega}{\partial n}\right|_\alpha^{n+\frac{1}{2}} = 0 \xrightarrow{\text{wall data}} \mathcal{L}_\alpha^{n+\frac{1}{2}}, \mathcal{W}_\alpha^{n+\frac{1}{2}}$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}},\, \mathcal{L}_\alpha^{n+\frac{1}{2}} \xrightarrow{\text{ghosts for M2P}} \left(\nabla^2\omega_g\right)_{ij}^{n+\frac{1}{2}}$$
$$\left(\nabla^2\omega_g\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} = \omega_p^n + \Delta t\, \nu \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*},\, \mathbf{x}_p^{n+1},\, \mathcal{W}_\alpha^{n+\frac{1}{2}} \xrightarrow{\text{ghosts for P2M}} (\omega_g)_p^{n+1,*}$$
$$(\omega_g)_p^{n+1,*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1,*}$$

- *Near-wall diff. :*

$$\omega_{ij}^{n+1,*} \xrightarrow{\text{Poisson}} \mathbf{u}_{ij}^{n+1,*},\, \Delta\gamma_\alpha^{n+1,*}$$
$$\bar{q}_\alpha^{n+\frac{1}{2}} = \frac{\Delta\gamma_\alpha^{n+1,*}}{\Delta t} \xrightarrow{\text{SI diff.}} (\omega_w)_{ij}^{n+1}$$
$$\omega_{ij}^{n+1} = \omega_{ij}^{n+1,*} + (\omega_w)_{ij}^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles.

In the present case, all computations were performed with $n^r = 1$ (redistribution occurs at every time step). While the core of the methodology has been detailed in Chapters 4, 5 and 6, some additional comments about the initialization are made in Section 7.1.1. Section 7.1.2 presents special features concerning the Poisson solver and Section 7.1.3 describes the near-wall diffusion of the vorticity flux. The term "SI diff." appearing in the near-wall diffusion step stands for "sub iteration diffusion" and the operation is detailed in the latter section. Finally, the interpolation is further discussed in Section 7.1.4.

7.1.1 Initialization

The grid level set ϕ_{ij} can be precomputed, since the body does not move in the present case. When no analytical expression is available, this operation is performed by computing the signed distance of the grid nodes with respect to

a set of line segments approximating the body geometry. It allows to easily perform several operations, such as the identification of the control points or the computation of the normal vectors, which is a well-known feature in the literature about interface tracking methods.

Control points and normal vectors The control points are simply computed as the roots of ϕ_{ij} along the grid lines, which furthermore provides a way to limit the number of control points per grid line segment (between two adjacent nodes) to at most one item (accounting for more control points would not make any sense since the body geometry is then clearly under resolved at the grid level).

For bodies of complex geometry (where the level set and the normal vectors are not known analytically), the normal vector $\mathbf{n}_\alpha = (\nabla\phi/|\nabla\phi|)_\alpha$ is computed at the control points based on ϕ_{ij} and using a corrected stencil. The level set is continuous and differentiable across smooth boundaries and the nodes inside the body can therefore be used for the stencil discretizing the first derivatives of ϕ ($\partial\phi/\partial x$ and $\partial\phi/\partial y$). In case some sharp edges are present (e.g. an airfoil, see Section 7.2.2), the level set is no longer differentiable. Yet, these edges are naturally regularized by the level set computation (edges are geometrically under resolved, no matter the grid resolution). The stencil computation is based on the same approach as that for the wall velocity $\mathbf{u}_\alpha$ evaluation that is detailed in Appendix B.2, except that all 1-D stencils (in the current and transverse directions with respect to the associated control point) are chosen to be as centered as possible around the control point, independently of their intersection with the body boundary. The wall information required by the immersed interface schemes is $\phi_\alpha = 0$, by definition.

Control point length scale for surface integration A length scale $b_{\alpha,k}$ associated to every control point $\mathbf{x}_{\alpha,k}$ can also be evaluated, based on the level set and an immersed interface finite difference scheme (the index k covers all control points in the x and y directions). The key idea is to take benefit of the link that exists between the jump of the streamfunction normal derivative (and hence the singular vortex sheet strength) and the "bulk vortex sheet" that is a grid quantity and results from the immersed interface correction terms. In a continuous context, the vortex singular sheet can be written by means of a convolution as

$$\omega_{\Delta\gamma}(\mathbf{x}) \triangleq \oint_{\partial\Omega_b} \Delta\gamma(\mathbf{x}')\ \delta(\mathbf{x}-\mathbf{x}')\ d\mathbf{x}' \ .$$

Once the problem has been discretized, the latter becomes a bulk quantity and it can be seen as a mollification of the singular vortex sheet along the normal to the boundary $\partial\Omega_b$. Considering the discretized Poisson equation $\nabla^2\Psi = -\omega$, it is here defined as

$$\left(\nabla^2\Psi\right)^h_{ij} = -\omega_{ij} - \sum_{k\in\mathcal{N}_{ij}} \left(-\frac{J_{\alpha,k}}{h^2}\right) \triangleq -\omega_{ij} - \left(\omega_{\Delta\gamma}\right)_{ij} \ ,$$

with $\left(\nabla^2\Psi\right)^h_{ij}$ the classical 5-point stencil for the Laplacian, $J_{\alpha,k}$ the correction terms (i.e. $J^+_{\alpha,k}$ or $J^-_{\alpha,k}$) and $\mathcal{N}_{ij}$ the subset of control points affecting the stencil of the grid node $\mathbf{x}_{ij}$, according to Chapter 4. And conversely, a given control point $\mathbf{x}_{\alpha,k}$ induces corrections for the two neighboring nodes and hence it contributes to the total bulk vortex sheet circulation by a quantity $\Gamma_{\alpha,k} \triangleq -h^2\ (J^-_{\alpha,k} + J^+_{\alpha,k})/h^2$.

One may further assign a virtual panel of length $b_{\alpha,k}$ to the control point $\mathbf{x}_{\alpha,k}$. Its circulation is then also accordingly $b_{\alpha,k}\Delta\gamma_{\alpha,k}$ (considering a panel of uniform intensity), with $\Delta\gamma_{\alpha,k} = -(\partial\Psi/\partial n)_{\alpha,k}$ the vortex sheet value at $\mathbf{x}_{\alpha,k}$. Both these circulations should match and an equivalent "numerical" panel length can thus be defined as

$$b_{\alpha,k} \triangleq \frac{(J^-_{\alpha,k} + J^+_{\alpha,k})}{\left(\dfrac{\partial\Psi}{\partial n}\right)_{\alpha,k}} \ , \tag{7.1}$$

where $(\partial\Psi/\partial n)_{\alpha,k}$ can be evaluated using the tools from the next Section 7.1.2. The latter definition is however solution dependent and is ill-posed when the vortex sheet is equal to zero. Moreover, it requires to solve the Poisson equation in order to obtain the corrections.

A better approach, that does not require any Poisson solution and is well-posed, consists in using the same definition Eq (7.1), however applied to the level set ϕ_{ij}. Indeed, this field naturally enjoys the necessary properties in order to compute $b_{\alpha,k}$, as $(\partial\phi/\partial n)_{\alpha,k} = 1$, by definition. Let us moreover consider the following auxiliary function

$$\zeta(\mathbf{x}) \triangleq \begin{cases} \phi(\mathbf{x}) & \text{if } \mathbf{x} \in \Omega_f \ , \\ 0 & \text{if } \mathbf{x} \in \Omega_b \ . \end{cases}$$

This function is continuous and presents a unitary normal derivative jump. Computing the immersed interface corrections $(J^\zeta)_{\alpha,k}$ associated to $\nabla^2\zeta$ (using the fact that $\zeta_{\alpha,k} = 0$) directly yields the desired length scale

$$b_{\alpha,k} \triangleq (J^\zeta)^-_{\alpha,k} + (J^\zeta)^+_{\alpha,k} \ .$$

The link between the vortex sheet and the correction terms remains valid here, as ζ is the solution of an underlying Poisson equation (the source term of this Poisson equation is simply $\nabla^2\zeta$).

A grid convergence study is performed in order to verify if this definition indeed corresponds to a natural length scale. The test case consists in computing the perimeter of a circle of diameter D and centered at $\mathbf{x}_c = (0, 0)$ in a domain $[-D, +D] \times [-D, +D]$ discretized using a $N \times N$ grid. The results, presented in Fig. 7.1, show that the error $\epsilon_b \triangleq |\pi - \sum_k b_{\alpha,k}/D|$ converges with a second order slope (actually 1.97).

This length scale may for example be used as an integration measure for contour integrals computed on the body wall (e.g. evaluation of the body forces using the wall vorticity for the friction contribution and the vorticity flux for the pressure contribution, cfr. [73]). The present formulation stands in contrast with classical immersed boundaries, where the definition of consistent integration measures can be problematic [44].

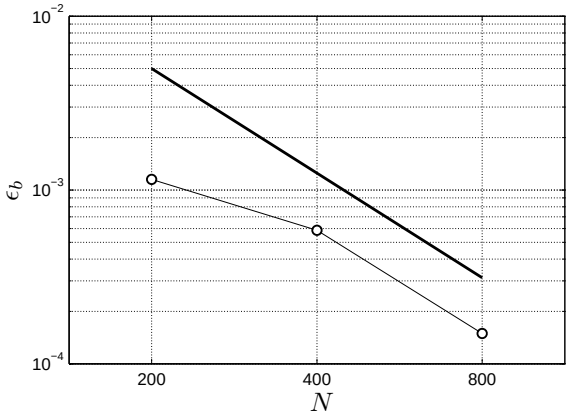


Figure 7.1: *Grid convergence study for the evaluation of the perimeter of a circle using the length scales $b_{\alpha,k}$: present error (thin solid line with "○"-signs) and second order slope (thick solid line).*

7.1.2 Poisson solver

The streamfunction Ψ is computed by applying the methodology developed in Chapter 4. The circulation of the "bulk vortex sheet" $(\omega_{\Delta\gamma})_{ij}$ (associated to the incremental vortex sheet $\Delta\gamma$) has to be prescribed and, also following [72], Kelvin's theorem must be enforced in order to complete the problem statement (see Appendix D). For a non-rotating body, this corresponds to

$$\Delta\Gamma \triangleq \oint_{\partial\Omega_b} \Delta\gamma(\mathbf{x}')\ d\mathbf{x}' = 0\ ,$$

and $\int_{\mathbb{R}^2} \omega(\mathbf{x}')\ d\mathbf{x}' = 0$ holds anyway (as all the vorticity is generated at solid walls). The sequence of immersed interface operations being not strictly conservative (at convergence, it should be the case, though), a correction is required and, if we further assume that the vorticity support is entirely included inside the computational domain, we can impose that the total circulation is equal to zero

$$\sum_{i,j} \Big(\omega_{ij} + (\omega_{\Delta\gamma})_{ij}\Big)\ h^2 = 0\ . \tag{7.2}$$

Before solving the Poisson equation, notice that the vorticity is processed on the outer boundaries of the computational domain $\partial\Omega_{comp}$ using a smooth cut-off function, such that $\omega = 0$ on $\partial\Omega_{comp}$. This treatment clearly induces a loss of circulation when the vorticity reaches the outer boundaries, at the outflow plane. Nevertheless, it does not seem to influence the solution too much when the outflow plane is far from the body, according to [109] where the flow past a cylinder was studied using a penalized VPM method (the forces and the Strouhal number are in good agreement with reference studies). Note that the vorticity support is then no longer entirely included inside the computational domain. One could therefore think of a way to track the circulation that has crossed the outflow plane, so as to still enforce zero total circulation in the framework of the above correction Eq. (7.2).

We further mention that the previous solution and outer boundary condition (recall that this is an unknown too, in the present methodology) are reused as a first guess during the Miller iteration (see Section 4.3.3), so as to improve the convergence and hence to minimize the number of calls to the linear solver per Poisson solution. In practice, this number is equal to 3 or 4, when the flow is fully established.

Once the streamfunction is computed, the velocity is obtained using the corrected schemes from Appendix B.1. The vortex sheet evaluation, that is required for the vorticity flux, is based on the same stencils as those used for $\mathbf{u}_\alpha$, it is given by

$$\Delta\gamma_\alpha = -n_\xi \left(\frac{\partial\Psi}{\partial\xi}\right)_\alpha - n_\eta \left(\frac{\partial\Psi}{\partial\eta}\right)_\alpha ,$$

where η is the transverse direction. The evaluation of the vortex sheet is illustrated in Figure 7.2 for an impulsively started cylinder at $t = 0^+$ (using $h/D = 5 \cdot 10^{-3}$ with h the mesh size and D the cylinder diameter; see Section 7.2.1 for a more detailed case description) and it is compared to the analytical vortex sheet of a potential flow past a cylinder that represents the initial condition of such an impulsively started flow. The maximum observed error on $\Delta\gamma/U_\infty$ is $7.254 \cdot 10^{-5}$.

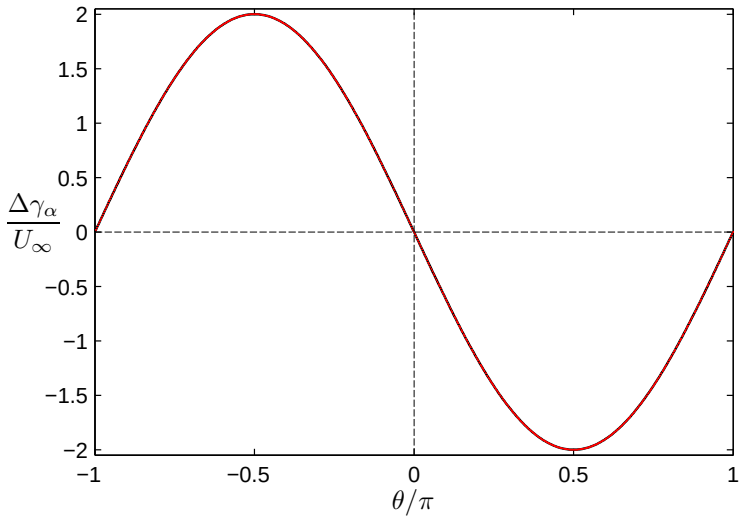


Figure 7.2: *Wall vortex sheet at $t = 0^+$ for the impulsively started cylinder flow, as a function of the angle θ ($\theta = 0$ downstream of the cylinder) : present (red $*$) and analytical (solid black line).*

7.1.3 Near-wall diffusion

The near-wall diffusion step is performed using a sub-iteration procedure, in order to better capture the near-wall vorticity field. Considering the predictor step (the corrector is treated similarly), we start from $(\omega_w)_{ij}^{n+\frac{1}{2},0} = 0$ and we

compute the following Euler scheme using N_{sub} sub steps

$$(\omega_w)_{ij}^{n+\frac{1}{2},k+1} = (\omega_w)_{ij}^{n+\frac{1}{2},k} + \frac{\Delta t_{\text{sub}}}{2}\,\nu\left(\nabla^2\omega_w\right)_{ij}^{n+\frac{1}{2},k} \quad ,$$

with $\Delta t_{\text{sub}} \triangleq \Delta t/N_{\text{sub}}$. The Laplacian is computed using corrected finite differences with a compatible extrapolation scheme in order to prescribe the flux $\bar{q}_\alpha^n$ at every control point (see Chapter 5). The same compatible extrapolation scheme is also used to compute the zero flux diffusion step. The value $N_{\text{sub}} = 5$ is adopted here for the applications presented in Section 7.2.

Nothing particular is done in order to ensure the strict conservation of circulation during this step, since the approach intrinsically computes field values ω pointwise, as opposed to the intensity $\int \omega\, d\mathbf{x}$ used in classical vortex methods. Yet, the conservation of circulation is obtained at convergence, thanks to the accuracy of the resulting vorticity field.

Nevertheless, the combination of this near-wall diffusion solver and the immersed interface Poisson solver is not fully consistent. The reason for this resides in the fact that the Poisson solver does not really "feel" the presence of the solid wall, in the sense that it only sees the corrections in the form of the grid bulk vortex sheet $(\omega_{\Delta\gamma})_{ij}$, in addition to the "real" flow vorticity ω_{ij}.

The intrinsic integration support associated to a given node vorticity ω_{ij} is the area h^2 and its circulation is hence $\omega_{ij}h^2$, whether it crosses the boundary or not. As an illustration of this assertion, consider the following small numerical test. Let us consider a vorticity field consisting of a discrete singularity of circulation Γ_0 at an arbitrary grid node $\mathbf{x}_{mn}$ (thus $\omega_{ij} = \Gamma_0\delta_{im}\delta_{jn}/h^2$). Now, compute the associate velocity field $\mathbf{u}_{ij}$ (by solving the Poisson equation with the immersed interface corrections at the wall) and compute the resulting circulation $\oint_{\mathcal{C}} \mathbf{u}\cdot d\mathbf{x}$ where $\mathcal{C}$ is a rectangle aligned with the grid, enclosing the body and the node $\mathbf{x}_{mn}$ (e.g. using a trapezoidal rule for the integration). Regardless of the position of $\mathbf{x}_{mn}$ with respect to the interface, this measured circulation is then Γ_0 up to the solver tolerance.

As a consequence, the Poisson solver treats the vorticity as a cell-averaged quantity, whereas the near-wall diffusion uses it as a pointwise quantity (remember that the streamfunction solution has a pointwise meaning, by construction). What makes the link between both perspectives is precisely the bulk vortex sheet. The bulk vortex sheet basically has two contributions, one coming from the actual vortex sheet $\Delta\gamma$ and the other one complementing and correcting the flow vorticity intensity $\omega_{ij}h^2$, due to the intersection of the ir-

regular cells with the boundary. Yet, computing this decomposition explicitly is hardly achievable. This is another reason to impose the total circulation, Eq. (7.2) (flow vorticity and bulk vortex sheet), as opposed to imposing the bulk vortex sheet circulation alone.

7.1.4 Particle-mesh interpolation

The M_4' interpolation kernel is used here for the mapping operations. Computing the ghosts relies on the techniques developed and presented in Chapter 6. The wall data sets $\mathcal{U}_\alpha$ and $\mathcal{L}_\alpha$ required for the M2P ghost computation of $\mathbf{u}$ and $\nabla^2\omega$ are obtained using the schemes from Appendix B.2. Figure 7.3 illustrates the resulting ghost velocity field $(\mathbf{u}_g)_{ij}^n$ for an impulsively started cylinder at $Re = 550$ (see Section 7.2.1 for the case description).

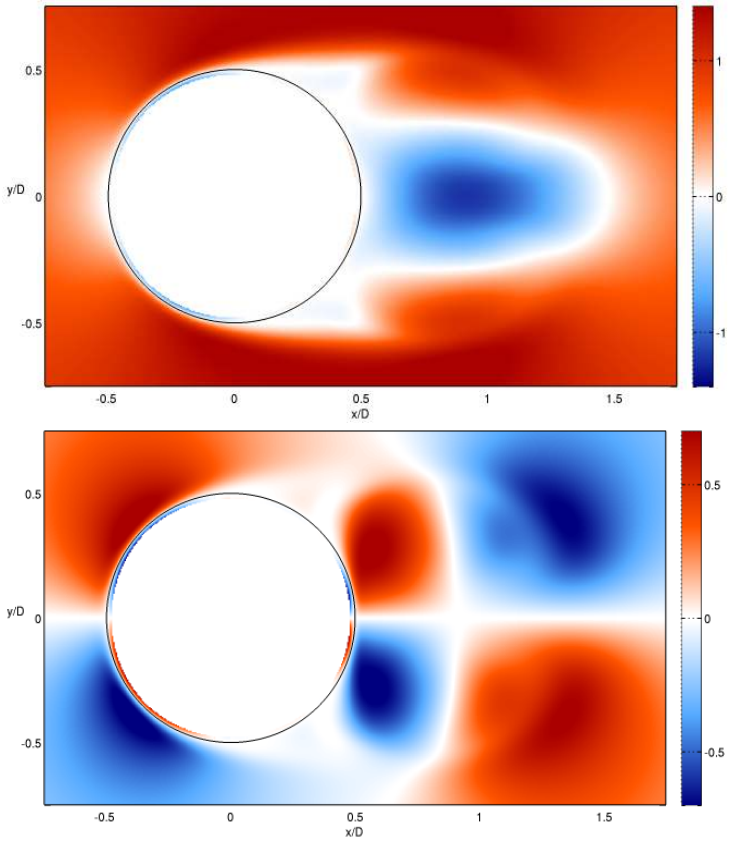


Figure 7.3: *Velocity ghost field $(\mathbf{u}_g)_{ij}^n/U_\infty$ for an impulsively started cylinder at $T = 3$ and for $Re = 550$: $(u_g)_{ij}^n/U_\infty$ (top) and $(v_g)_{ij}^n/U_\infty$ (bottom).*

7.2 Results

Section 7.2.1 describes the test case about an impulsively started cylinder, Section 7.2.2 provides simulation results for an impulsively started airfoil and Section 7.2.3 shows the ability of the solver to capture unsteady phenomena at the wall that are not limited to the starting phase, but that carry on in the flow regime (vortex shedding induced by the flow past a cylinder).

7.2.1 Impulsively started cylinder

The flow past an impulsively started cylinder offers a convenient framework for the validation of the present immersed interface vortex particle-mesh solver. It has been studied extensively both theoretically [30, 29, 5] and computationally in Koumoutsakos and Leonard [73] and Ploumhans and Winckelmans [102] (referred to as KL and PW respectively). The simulation of this type of flow is particularly challenging, as the initial acceleration - and hence also the initial force - is infinite. The following expression of the drag coefficient [5], valid for short times, shows the associated singularity behaving like $t^{-1/2}$, which makes the simulation under resolved in time, regardless of the chosen time step Δt:

$$C_D \triangleq \frac{\mathbf{F}\cdot\hat{\mathbf{v}}}{\frac{1}{2}\rho U_\infty^2 D} = 4\left(\frac{\pi}{Re\,T}\right)^{\frac{1}{2}} + 2\pi\left(9-\frac{15}{\pi^{\frac{1}{2}}}\right)\frac{1}{Re}\,, \tag{7.3}$$

where $\mathbf{F}$ is the force per unit length acting on the cylinder, $\mathbf{U}_\infty \triangleq U_\infty\ \hat{\mathbf{v}}$ the upstream velocity field, D the diameter of the cylinder, ρ the fluid density, $T \triangleq U_\infty t/D$ the dimensionless time, $Re \triangleq U_\infty D/\nu$ the Reynolds number with ν the kinematic fluid viscosity. The linear impulse $\mathbf{I}$ of the fluid in the flow direction resulting from Eq. (7.3) then reads

$$\begin{aligned}\frac{\mathbf{I}\cdot\hat{\mathbf{v}}}{U_\infty D^2} &\triangleq \frac{1}{U_\infty D^2}\ \hat{\mathbf{v}}\cdot\int_{\Omega_f}\mathbf{x}\times(\omega\hat{\mathbf{e}}_z)\,d\mathbf{x} \\ &= -4\left(\frac{\pi T}{Re}\right)^{\frac{1}{2}} - \pi\left(9-\frac{15}{\pi^{\frac{1}{2}}}\right)\frac{T}{Re} - \frac{\pi}{2}\,,\end{aligned} \tag{7.4}$$

as $\mathbf{F} = -\rho\, d\mathbf{I}/dt$.

Vortex methods relying on Lighthill's model for the no-slip enforcement, as discussed in Chapter 3, are particularly well-suited for the simulation of impulsively started flows, thanks to the proper computation of the vortex sheet, that is subsequently diffused to determine the vorticity/circulation that must

enter the flow at every time step, as the vortex sheet determines the wall vorticity flux. The study of the present test case using purely Lagrangian vortex methods [73, 102] indeed show that the resulting global diagnostics (the forces and the linear impulse), as well as the locally computed data (wall vorticity, wall flux, etc.) agree well with the theoretical predictions and reference computations.

According to the remarks that have been made in Section 7.1.3, the computation of the linear impulse of the flow for the present methodology must include the vorticity associated to the bulk vortex sheet, as follows

$$\mathbf{I}^n = \sum_{i,j} \mathbf{x}_{ij} \times \hat{\mathbf{e}}_z \left(\omega_{ij}^n + (\omega_{\Delta\gamma})_{ij}^n \right) h^2 \ ,$$

similarly to the computation of the total circulation. Indeed, on the one hand, $(\omega_{\Delta\gamma})_{ij}$ contains a contribution of the flow vorticity, as previously explained, and, on the other hand, the vortex sheet is a component of the flow (in the sense of Lightill's model). The integration support for each grid node is again h^2, as it was already the case for the added mass evaluation from Section 4.4.2.

The case $Re = 550$ is studied here in a computational domain $[-D; 6D] \times [-1.5D; 1.5D]$, using a 1400×600 grid (the mesh size is thus $h/D = 5 \cdot 10^{-3}$). The cylinder is centered at the origin $\mathbf{x}_c = (0, 0)$. The time step is set to $\Delta T \triangleq U_\infty \Delta t/D = 2.5 \cdot 10^{-3}$, which leads to a Fourier number $r \triangleq \nu \Delta t/h^2 = 0.182$. The ghost computation for the velocity M2P interpolation (see Chapter 6) is switched on after 50 time steps, i.e. before that, the ghost velocity is set to zero. This is required, as the abrupt acceleration of the body may lead to the generation of some low amplitude noise in the near-wall vorticity field, that can be amplified by the interpolation procedure.

Fig. 7.4 shows the x-component of the linear impulse (i.e. in the flow direction), as a function of T. A comparison is made between the present approach, the results from Ploumhans and Winckelmans [102] (a purely Lagrangian vortex method) and the analytical solution Eq. (7.4). The computations reported from [102] correspond to the "G+V" approach (here referred to as PW1) and to the "KL"-like approach (here PW2) defined therein. Both approaches compute the diffusion term by means of the PSE scheme [41] with a zero-flux condition at the wall and the particle redistribution near the wall is carried out using one-sided schemes. A vortex panel method is used for the no-through flow velocity computation and explicit integral formulas for the near-wall diffusion (see

Appendix E.8 for a detailed review of these formulas). Basically, the "G+V" approach uses "random vibrations" at each redistribution and the "KL"-like approach uses a "body fitted" grid (based on polar coordinates) that follows the cylinder boundary for the redistribution (the redistribution technique is close to what is done in Koumoutsakos and Leonard [73], hence the name "KL"-like). The different computational setups are summarized in Table 7.1.

	h/D	ΔT	n^r	denom. in [102]
Present	$5 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$	1	-
PW1C-550	$6.03 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	5	"G+V"
PW1F-550	$3.015 \cdot 10^{-3}$	$2.5 \cdot 10^{-3}$	5	"G+V"
PW2C-550	$6.03 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	5	"KL"-like

Table 7.1: *Comparison of the numerical parameters used for the present approach and for the reference results from Ploumhans and Winckelmans [102] at $Re = 550$ (redistribution is performed every n^r time steps).*

The Fourier number for the 3 computations from [102] is $r = 0.5$, which is stable, since the time integration of the vorticity equation is there performed using a first order Euler scheme.

Results are given for short times in Fig. 7.4(a). The immersed interface VPM solver seems to agree well with all results, particularly with the coarser cases PW1C-550 and PW2C-550. This makes sense, as the spatial resolution is nearly identical in both cases (the present time resolution is higher, though). Note that the mesh size for the present approach has been chosen as the closest "round" value close to that from PW1C-550 and PW2C-550 for the sake of simplicity.

As expected, the initial phase gives rise to larger oscillations, due to the singular nature of the flow at $t = 0^+$. Yet, the present approach seems to capture this phase as well as the case from [102] with the finer resolution (PW1F-550). Another point that deserves attention is the very accurate capture of the initial impulse value (at $t = 0^+$), which corresponds to the impulse of a potential flow. This further justifies the above evaluation of the impulse, as it only consists of the bulk vortex sheet contribution at that time (there is no other vorticity present in the flow).

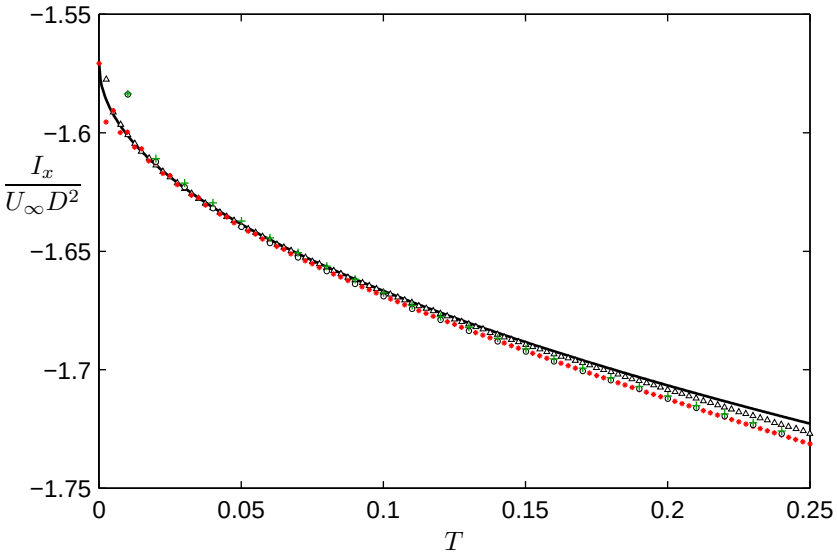
Fig. 7.4(b) provides the results for long times, by comparing the present approach to the PW2C-550 computation (analytical expressions are lacking

at such times). The PW2C-550 is considered as a reference here, since the redistribution is performed using a "body fitted" grid, as mentioned above. This type of wall treatment is more accurate but it can hardly be applied to more complex geometries. In the early phase, both curves are indistinguishable and a small difference appears later, which could be explained by the first order time integration used for the vorticity in Ploumhans and Winckelmans [102].

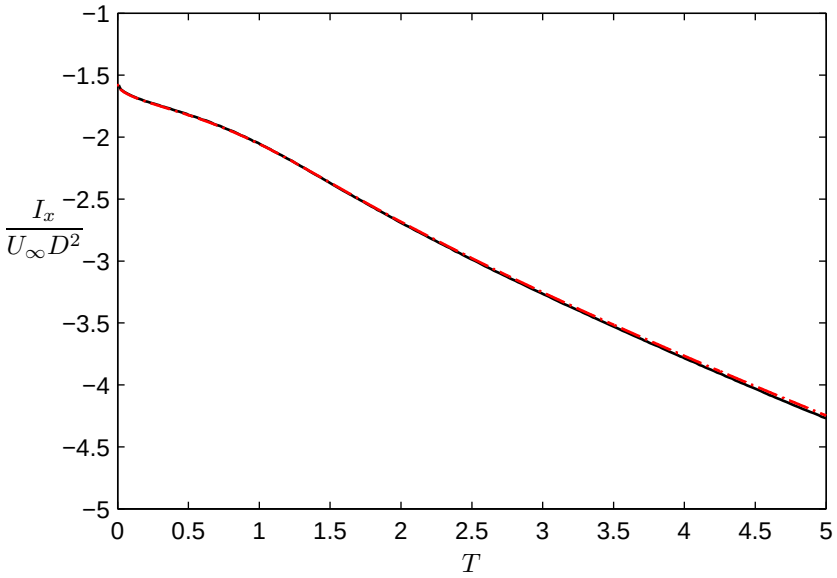
The same comparison is also made for the drag coefficient in Fig. 7.5. The analytical solution corresponds to Eq. (7.3) whereas, for the numerical studies, the computation of the force is based on the time derivative of the impulse, which is here evaluated using centered second order finite differences. One must keep in mind that the drag coefficients from Ploumhans and Winckelmans [102] have been obtained by first filtering the impulse signal using a five-point moving average prior to the numerical differentiation. On the contrary, the present results do not rely on any filtering. Again, results seem to agree quite well, except for the very first time steps, where larger oscillations are present. The long time evolution is captured quite well.

In Fig. 7.6, some iso contours of the vorticity field at various times are compared to the results from PW1C-550. Results are very similar, except for some small differences in the near wake (at $T = 3$, there is no local maximum at $\mathbf{x}/D \sim (1.25; \pm 0.2)$ for the present approach; and at $T = 3$ and $T = 5$, the vorticity maxima at the recirculation centers are slightly different). Those discrepancies could be explained by the higher redistribution frequency applied in the present approach (the redistribution period is $\Delta T^r = 2.5 \cdot 10^{-3}$ for the present computation, $\Delta T^r = 5 \cdot 10^{-2}$ for PW1C-550 and PW2C-550, and $\Delta T^r = 1.25 \cdot 10^{-2}$ for PW1F-550). Fig. 7.7 further shows a close-up view of the vorticity in the downstream part of the flow.

The immersed interface approach also allows to compute the vorticity at the wall, based on the vorticity flux at every control point, according to the tools that have been developed in Chapter 5. Fig. 7.8 shows a comparison at $T = 1$ with the results from Koumoutsakos and Leonard [73], referred to as the "KL" approach (it has to be distinguished from the "KL"-like approach PW2). The vorticity maximum present some high-frequency noise, compared to the KL methodology. Yet, both curves are in good agreement, keeping in mind that the KL computation uses a body-fitted grid for the redistribution and that an accurate capture of field values at the wall remains challenging for vortex particle-mesh methods.

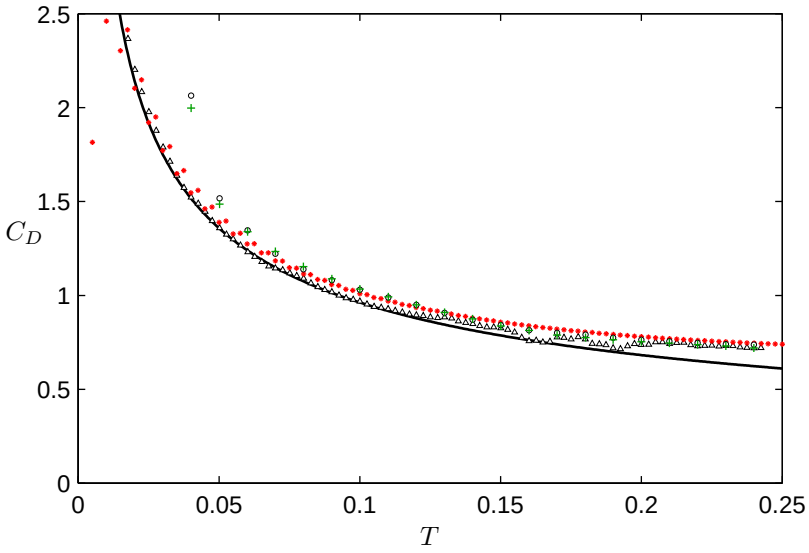


(a)

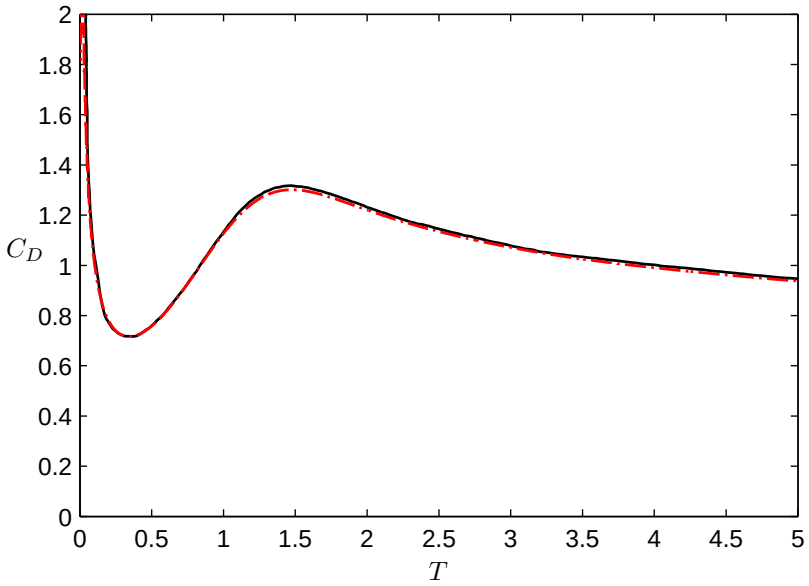


(b)

Figure 7.4: *Linear impulse for the impulsively started cylinder flow at $Re = 550$; (a) short time comparison : present (red $*$), PW1C-550 ($\circ$), PW1F-550 ($\triangle$), PW2C-550 (green $+$) and analytical Eq. (7.4) (solid line); (b) long time comparison : present (red dash-dotted line) and PW2C-550 (solid line).*



(a)



(b)

Figure 7.5: *Drag coefficient for the impulsively started cylinder flow at $Re = 550$; (a) short time comparison : present (red $*$), PW1C-550 ($\circ$), PW1F-550 ($\triangle$), PW2C-550 (green $+$) and analytical Eq. (7.3) (solid line); (b) long time comparison : present (red dash-dotted line) and PW2C-550 (solid line).*

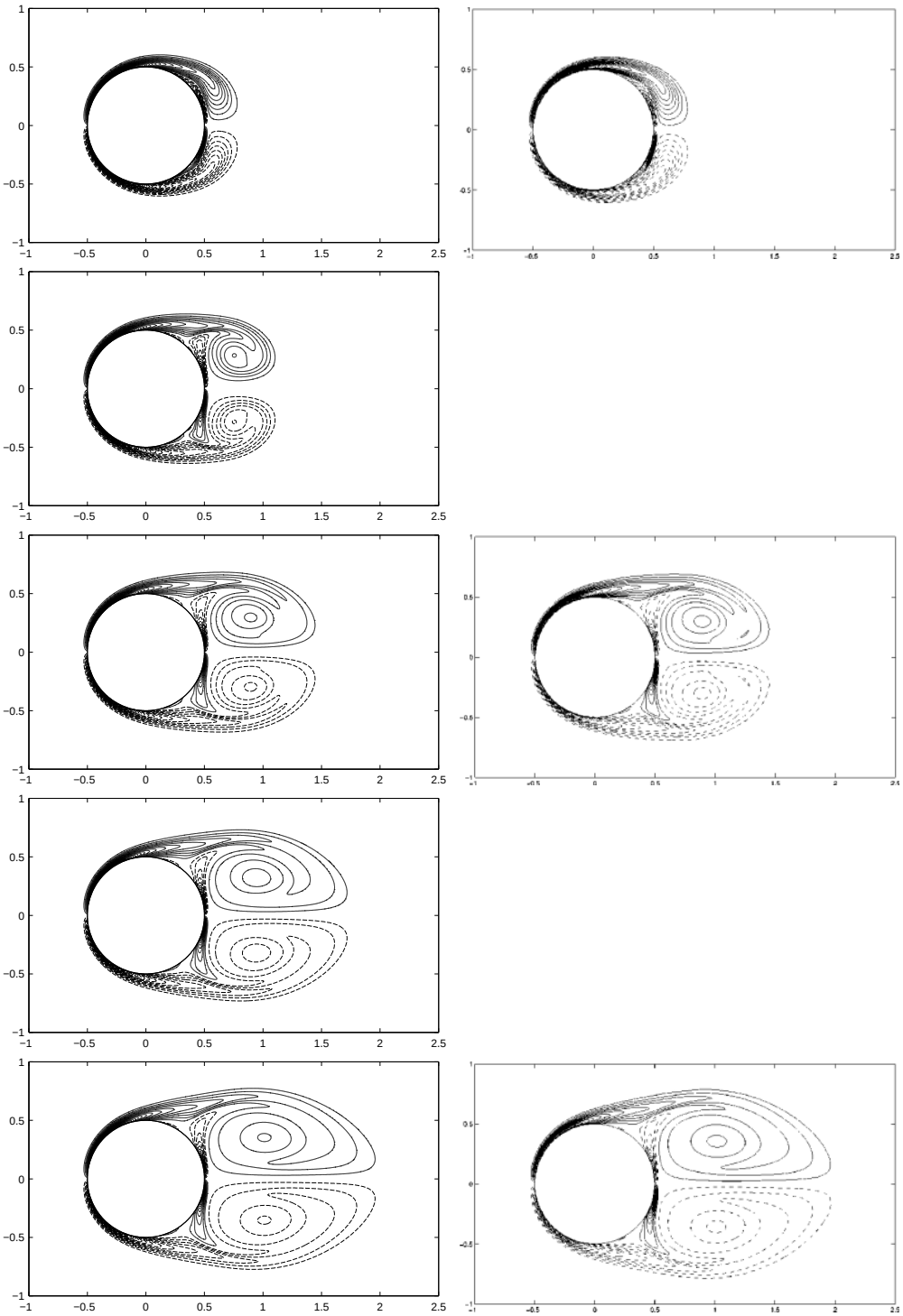


Figure 7.6: *Iso contours of vorticity $\omega D/U_\infty$ for $T = 1, 2, 3, 4$ and 5 (top to bottom) for the impulsively started cylinder flow at $Re = 550$: present (left) and PW1C-550 (right; figures are reproduced from Ploumhans and Winckelmans [102]). Levels are by steps of 2 ($\omega = 0$ is skipped).*

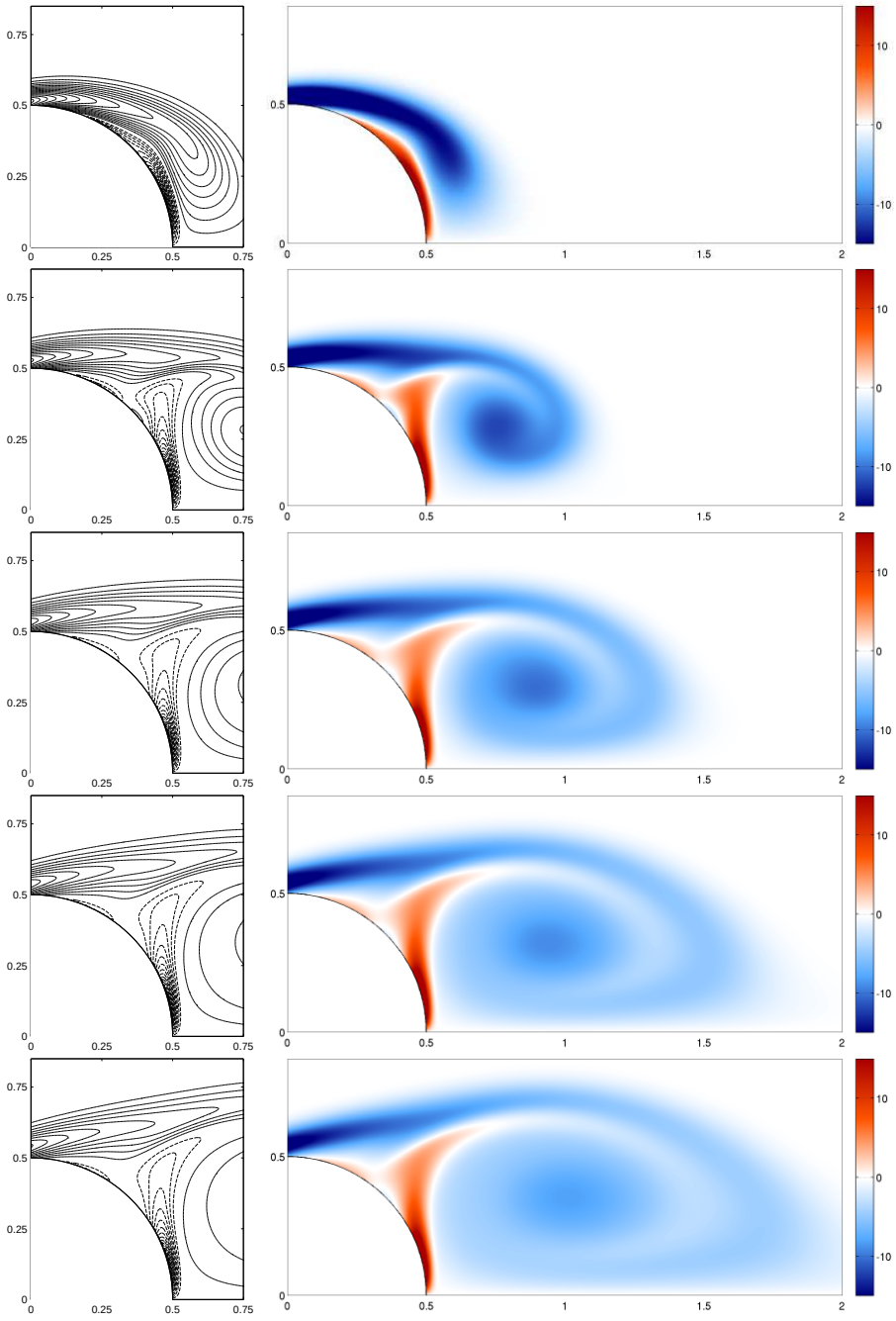


Figure 7.7: *Zoom on the vorticity field $\omega D/U_\infty$ at $T = 1, 2, 3, 4$ and 5 (top to bottom) for an impulsively started cylinder flow ($Re = 550$).*

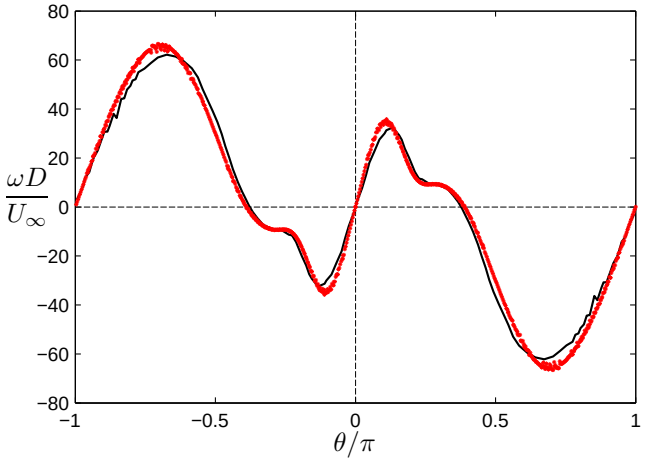


Figure 7.8: *Wall vorticity $\omega D/U_\infty$ for the impulsively started cylinder flow at $Re = 550$ and at $T = 1$: present (red $*$) and KL (solid line).*

Some results are also shown for the case $Re = 3000$. The computational domain is here $[-D; 2D] \times [-D; D]$ with a 1875×1250 grid (thus $h/D = 1.6 \cdot 10^{-3}$). The time step $\Delta T = 1 \cdot 10^{-3}$ leads to a Fourier number $r = 0.130$ and the ghost computation for the velocity interpolation procedure is switched on after 50 time steps, similarly to the case at $Re = 550$.

Fig. 7.9 shows the linear impulse in the x direction and Fig. 7.10 provides the drag coefficient. Again, we compare the impulse and the drag coefficient with the results from Ploumhans and Winckelmans [102], for the same approaches PW1 and PW2 (see Table 7.2 for the computational setups). For this Reynolds number, the differences between the various curves become more important, especially for the drag coefficient.

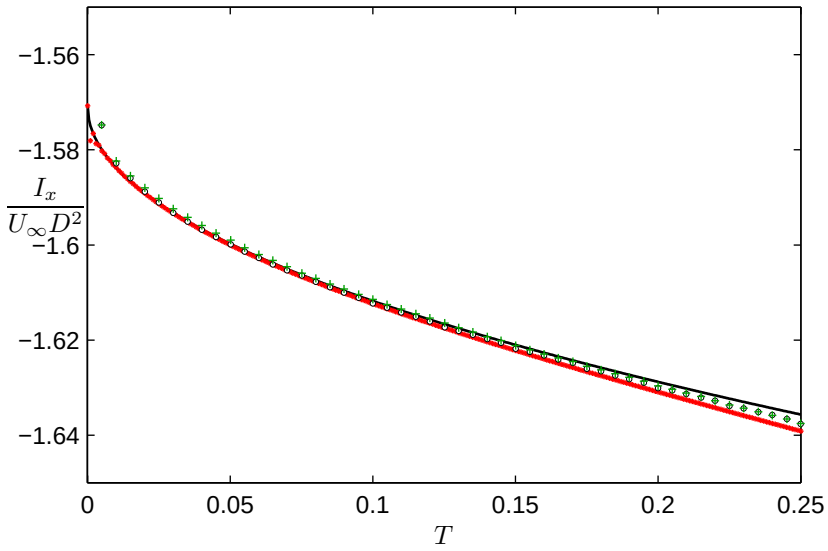
	h/D	ΔT	n^r	denom. in [102]
Present	$1.6 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	1	-
PW1-3000	$1.87 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	5	"G+V"
PW2-3000	$1.87 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	5	"KL"-like

Table 7.2: *Comparison of the numerical parameters used for the present approach and for the reference results from Ploumhans and Winckelmans [102] at $Re = 3000$ (redistribution is performed every n^r time steps).*

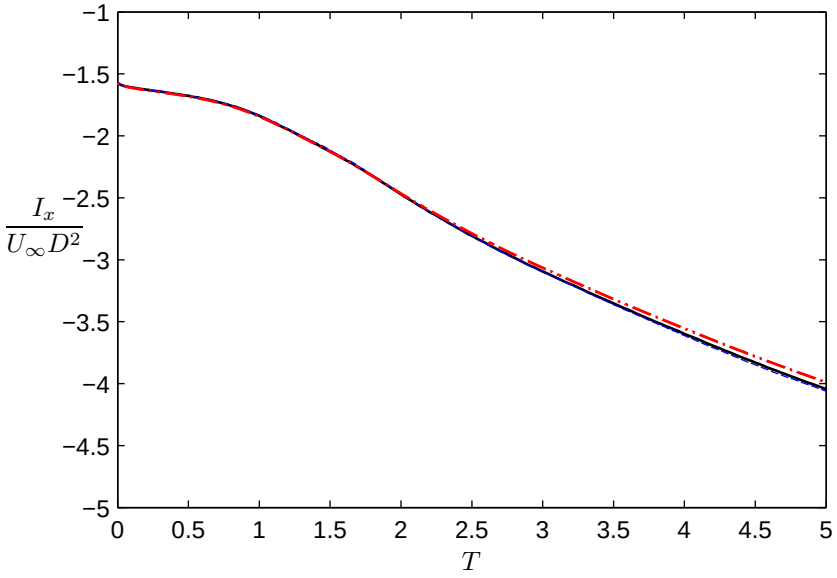
The present approach captures quite well the starting phase for the impulse, which is due to the five time smaller time step, compared to the results from Ploumhans and Winckelmans [102]. The short time evolution of I_x presented in Fig. 7.9(a) seems to indicate that the immersed-interface VPM solver and the PW1 approach (and also PW2) begin to differ at $T \simeq 0.15$, whereas the long time evolution in Fig. 7.9(b) shows that the results are still in good agreement for longer times.

Capturing the drag coefficient correctly is more challenging, as can be seen in Fig. 7.10(b). The PW1 and the PW2 approaches appear to disagree, with PW1 being noisier. The present methodology differs from PW2 at about the same moment as PW1. The peak around $T \simeq 2$ is under predicted by the present approach, however the kinks observed at $T \simeq 0.2, 1.35$ and 2.2 for both PW1 and PW1 do not occur here.

Concerning the iso contours of vorticity in Fig. 7.11, the immersed interface solver and PW1-3000 seem to agree very well until $T = 2$ and, apart from fine scale details, no real differences are observed. From $T = 3$ on, some discrepancies occur in the recirculation zones, which may also be linked to the present higher redistribution frequency which results in an increased diffusion. Fig. 7.12 again provides a close-up view of the vorticity in the downstream part of the flow.

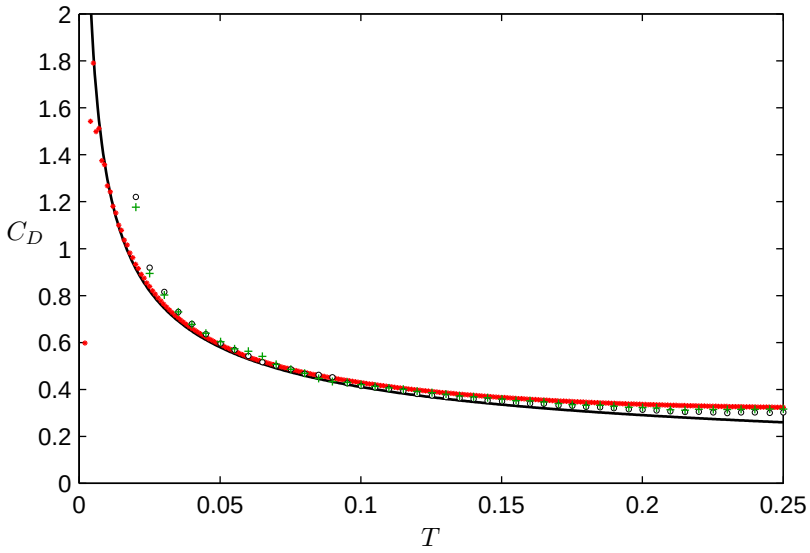


(a)

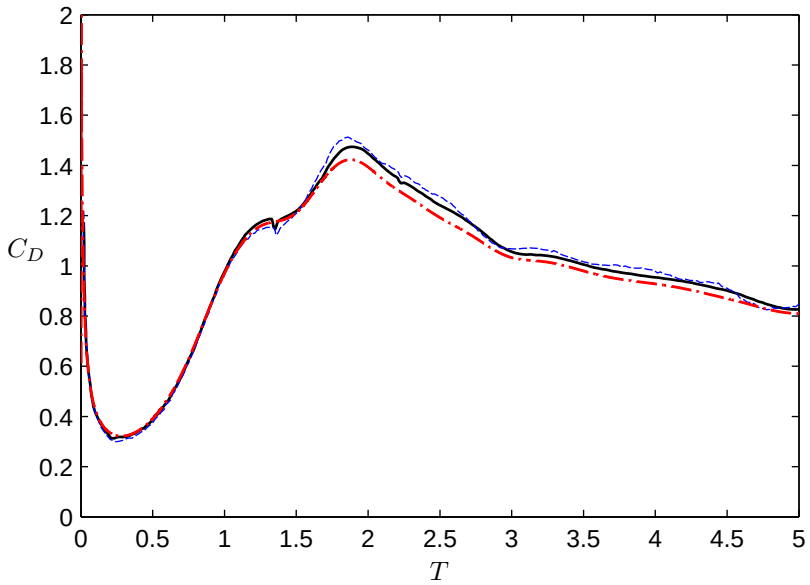


(b)

Figure 7.9: *Linear impulse for the impulsively started cylinder flow at $Re = 3000$; (a) short time comparison : present (red $*$), PW1-3000 ($\circ$), PW2-3000 (green $+$) and analytical Eq. (7.4) (solid line); (b) long time comparison : present (red dash-dotted line), PW1-3000 (blue dashed line) and PW2-3000 (solid line).*



(a)



(b)

Figure 7.10: *Drag coefficient for the impulsively started cylinder flow at $Re = 3000$; (a) short time comparison : present (red $*$), PW1-3000 ($\circ$), PW2-3000 (green $+$) and analytical Eq. (7.3) (solid line); (b) long time comparison : present (red dash-dotted line), PW1-3000 (blue dashed line) and PW2-3000 (solid line).*

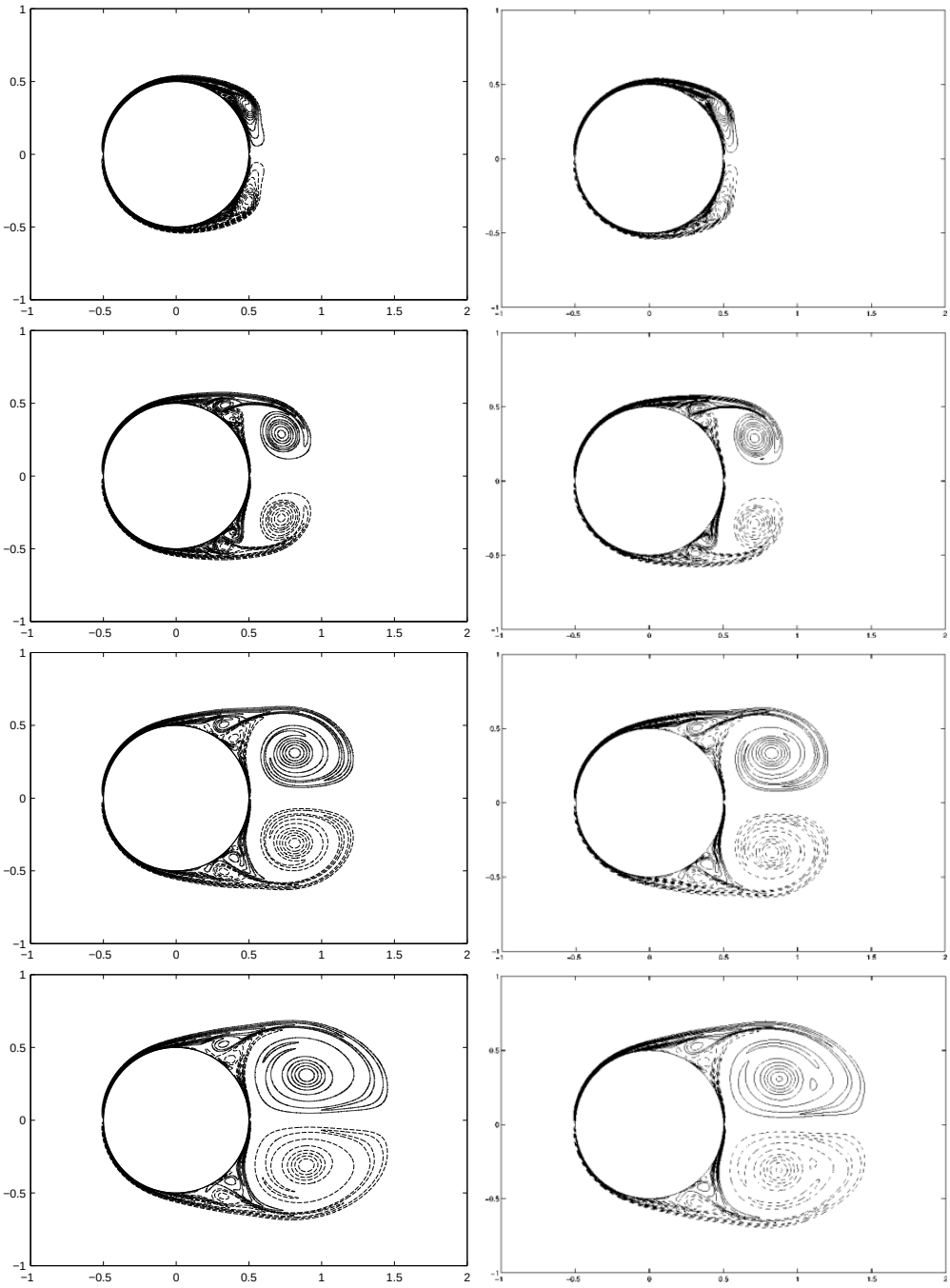


Figure 7.11: *Iso contours of vorticity $\omega D/U_\infty$ for $T = 1, 2, 3$ and 4 (top to bottom) for an impulsively started cylinder flow at $Re = 3000$: present (left) and PW1-3000 (right; figures are reproduced from Ploumhans and Winckelmans [102]). Levels are by steps of 4 ($\omega = 0$ is skipped).*

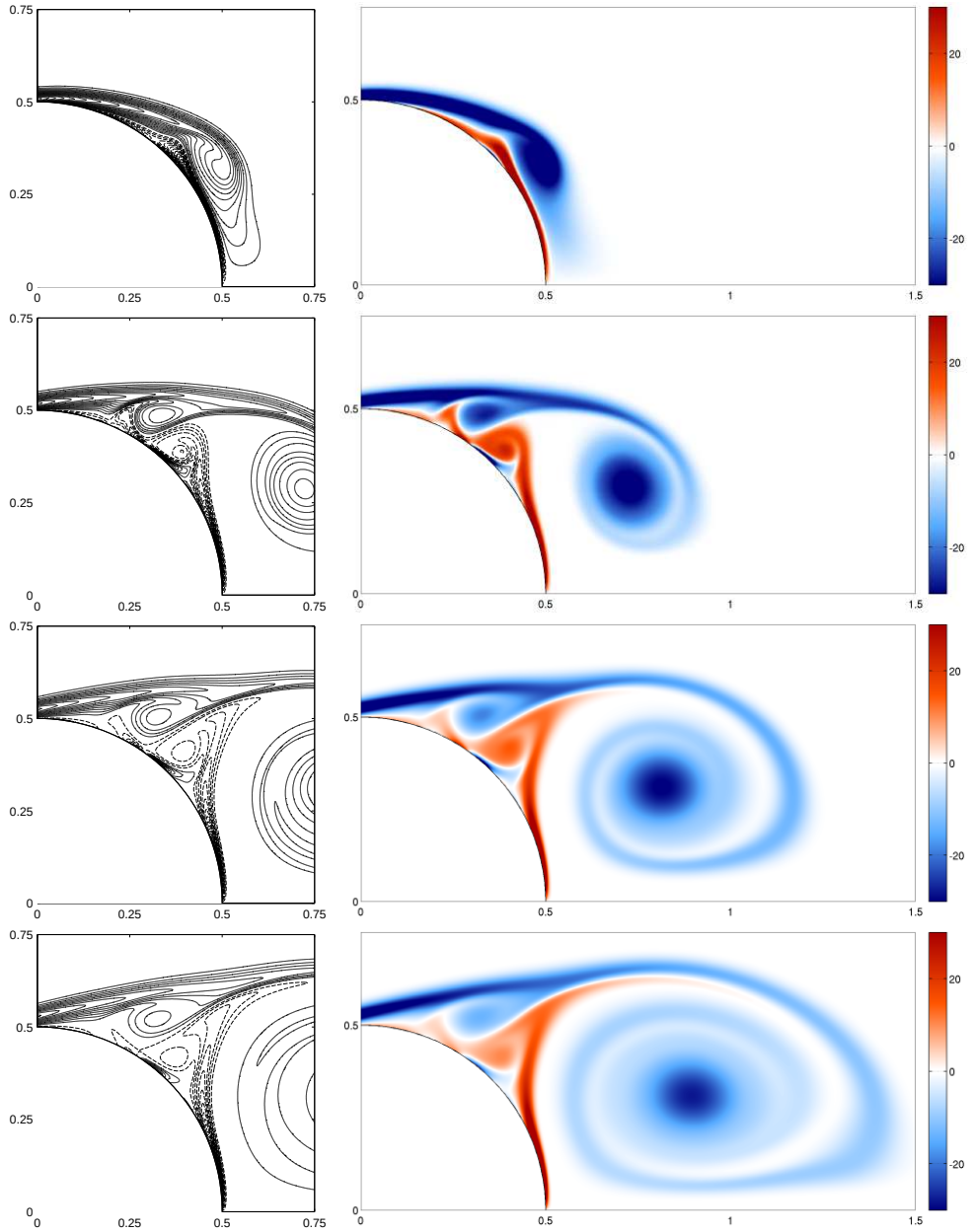


Figure 7.12: *Zoom on the vorticity field $\omega D/U_\infty$ at $T = 1, 2, 3$ and 4 (top to bottom) for an impulsively started cylinder flow ($Re = 3000$).*

7.2.2 Flow past an impulsively started NACA0021 airfoil

In this section, we study the ability of the immersed interface VPM solver to simulate the impulsively started flow past a NACA0021 airfoil in a high angle of attack (and stalled) configuration. The Reynolds number based on the chord c is taken as $Re \triangleq U_\infty c/\nu = 500$ and the angle of attack is given by $\alpha = 20°$. The computational domain is $[-0.25\ c;\ 2.75\ c] \times [-c; c]$ with a 1200×400 grid ($h/c = 2.5 \cdot 10^{-3}$) and the time step is $\Delta T \triangleq U_\infty \Delta t/c = 5 \cdot 10^{-4}$ ($r = 0.160$).

Fig. 7.13 shows the vorticity field at $T \triangleq U_\infty t/c = 0.1, 0.5, 1$ and 1.5. One may recognize the initial creation of the starting vortex associated to the lift generation. Once it has gained its full intensity, the starting vortex is advected downstream towards the outflow plane. Capturing this phenomenon properly is challenging, since, in addition to the drag singularity occurring at $T = 0^+$ (such as for the impulsively started cylinder), the initial velocity field is moreover singular at the sharp trailing edge. The latter corresponds to the potential flow field obtained with zero circulation around the airfoil, which does not satisfy the Kutta-Joukowsy condition. Fig. 7.14 shows that the vortex sheet (thus the tangential velocity) at $T = 0^+$ is indeed singular at the trailing edge. Hence, a separation point appears initially on the suction side near the trailing edge (at $T = 0^+$ this point is even a stagnation point). The near-wall diffusion process instantaneously regularizes the velocity field at $T = 0^+$ and the separation point moves further downstream until it reaches the trailing edge.

Fig. 7.15 further details this process by showing first the streamlines at $T = 0^+$, along with the stagnation point and the associated "bulk vortex sheet" grid field ω_γ (the support for this field is the set of irregular points according to Chapter 4). Secondly, the vorticity field is visualized together with the streamlines at several times. At early times, the problem is under resolved in this area and the displacement of the separation point is hardly captured. The higher vorticity (in absolute value) on the suction side in the vicinity of the trailing edge at $T = 0.1$ reflects the unsteady regime of the flow around the trailing edge and is partly due to the starting vortex, that is still close to this region at this time. Note that, for an attached steady flow past an airfoil, the vorticity integrals over the pressure and suction sides should be equal.

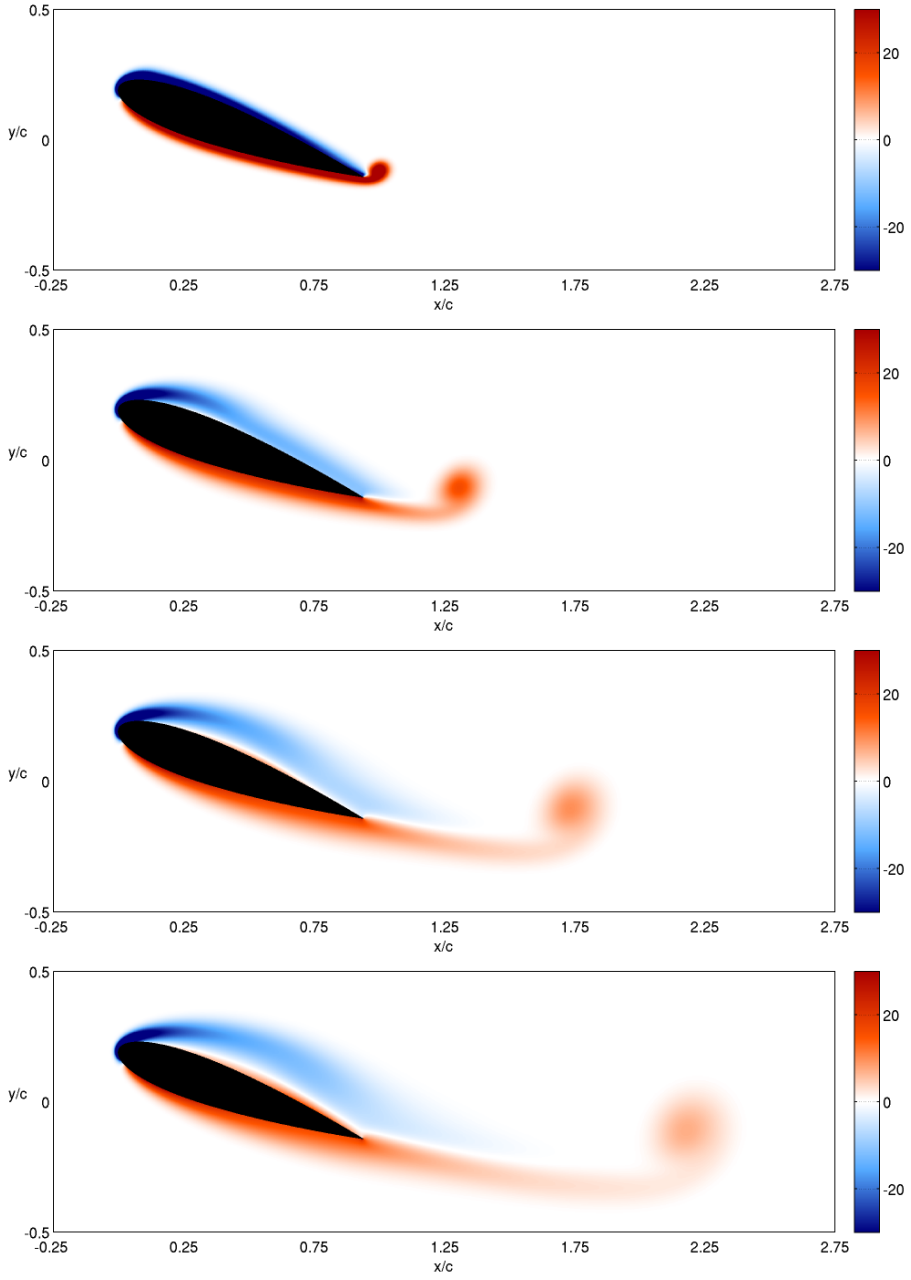


Figure 7.13: *Vorticity field $\omega c/U_\infty$ for an impulsively started NACA0021 airfoil at an angle of attack $\alpha = 20°$ and at $Re = 500$ (top to bottom : $T = 0.1, 0.5, 1$ and 1.5).*

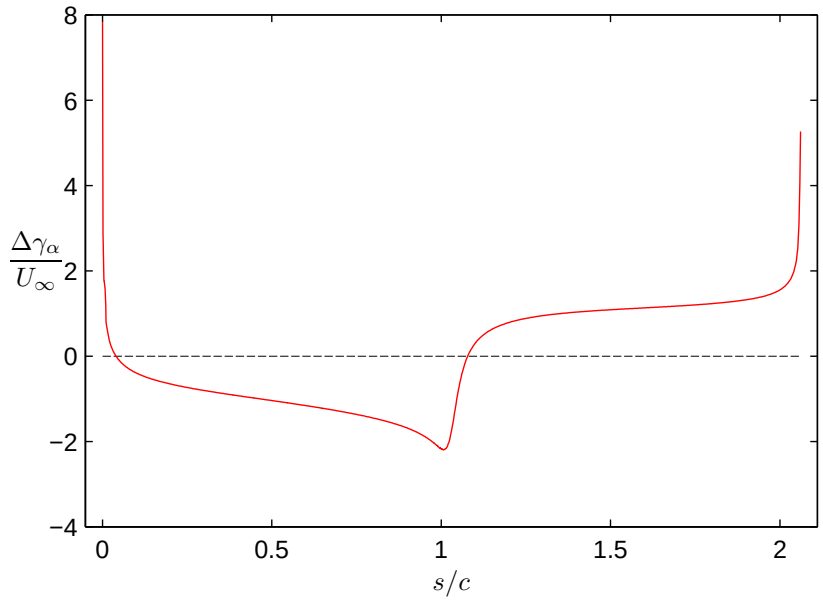


Figure 7.14: *Wall vortex sheet at $T = 0^+$ for an impulsively started NACA0021 airfoil at an angle of attack $\alpha = 20°$ ($Re = 500$) (as a function of the curvilinear coordinate s; $s = 0$ at the trailing edge and s is positive for a counterclockwise rotation).*

A separation zone appears on the suction side, which can be seen in Fig. 7.13 from $T = 0.5$ on. A close-up view with the streamlines is provided in Fig. 7.16, showing the growing of the separation bubble from $T = 0.8$ to $T = 1.2$. Moreover Fig. 7.17 shows some diagnostics, such as the linear impulse I_x, I_y and the force coefficients defined by

$$C_D \triangleq \frac{F_x}{\frac{1}{2}\rho U_\infty^2 c}\,, \quad C_L \triangleq \frac{F_y}{\frac{1}{2}\rho U_\infty^2 c}\,.$$

Both coefficients are computed based on the time derivative of the impulse, as was the case in Section 7.2.1. The evolution of the diagnostics after $T = 1.5$ is not shown here, as some of the vorticity leaves the domain.

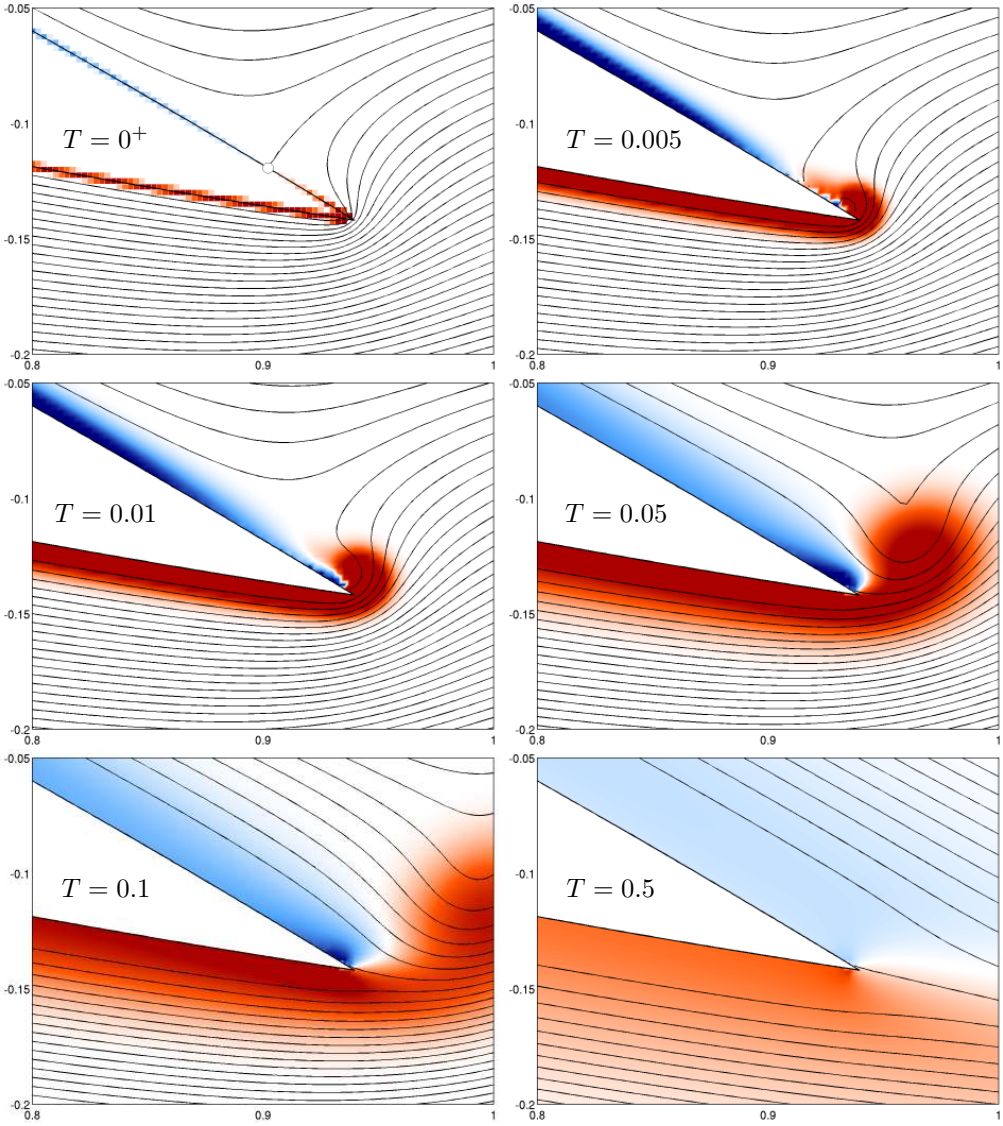


Figure 7.15: *Zoom of the streamlines in the trailing edge region of an impulsively started NACA0021 airfoil at an angle of attack $\alpha = 20°$ ($Re = 500$), along with the bulk vortex sheet ω_γ at $T = 0^+$ (with $\circ$ the stagnation point; top left figure) and the vorticity field $\omega c/U_\infty$ at various times (the color legend is different for the vorticity and for the bulk vortex sheet).*

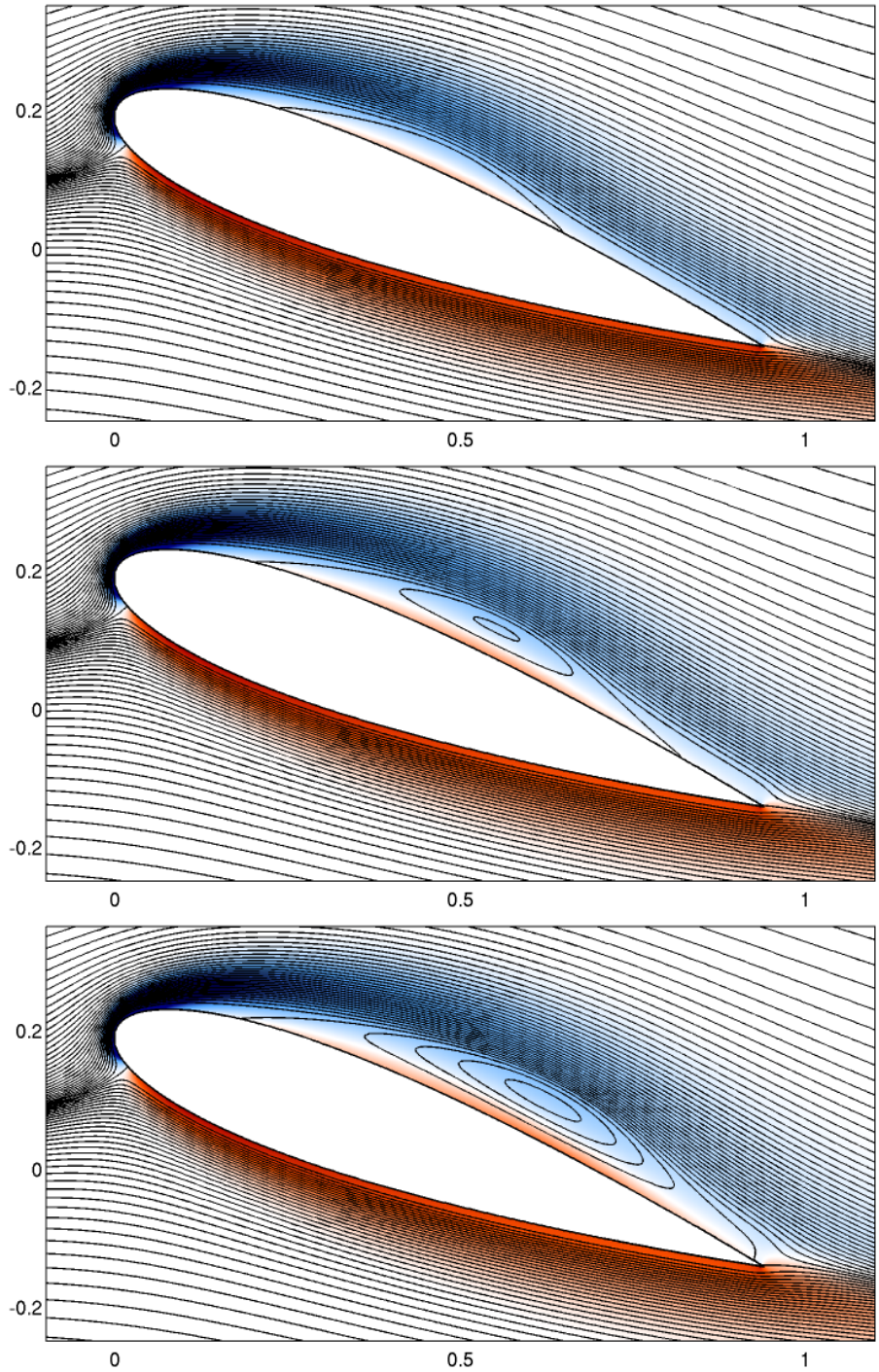
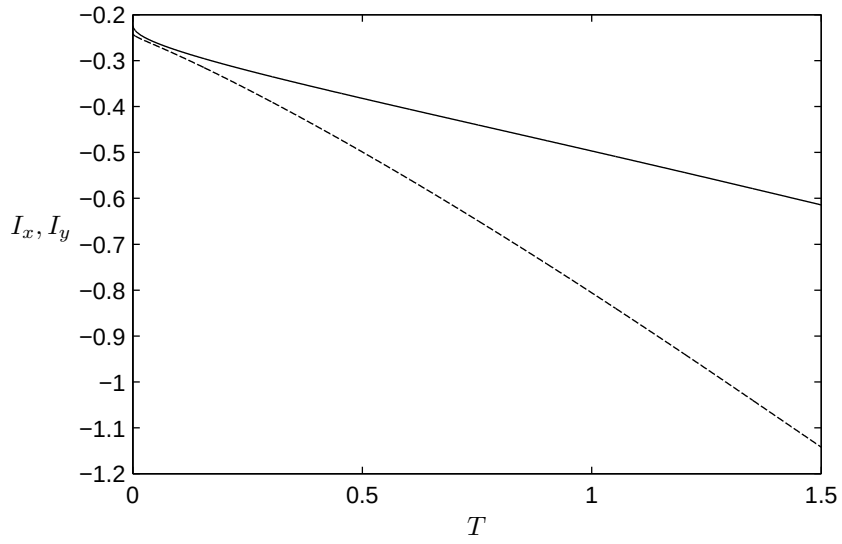
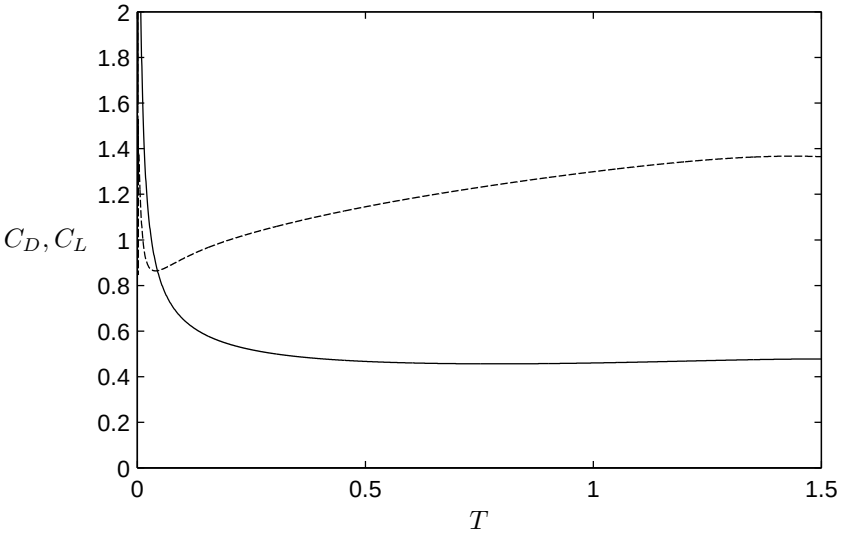


Figure 7.16: *Vorticity field $\omega c/U_\infty$ and streamlines for an impulsively started NACA0021 airfoil at an angle of attack $\alpha = 20°$ ($Re = 500$) (top to bottom : $T = 0.8, 1$ and 1.2; the iso-values of Ψ for the streamlines are not equidistant and the steps follow a geometric progression around $\Psi = \overline{\Psi}$, the streamfunction constant corresponding to the body, see Chapter 4).*



(a)



(b)

Figure 7.17: *Diagnostics for an impulsively started NACA0021 airfoil at an angle of attack $\alpha = 20°$ and at $Re = 500$: (a) linear impulse I_x (solid line) and I_y (dashed line); (b) force coefficients C_D (solid line) and C_L (dashed line);*

7.2.3 Flow past a cylinder at $Re = 100$

In this section, the flow past a cylinder at $Re = 100$ is studied. A 800×300 grid is used here for the simulation and the computational domain is given by $[-3D;\ 13D] \times [-3D; 3D]$ ($h/D = 2 \cdot 10^{-2}$ and $\Delta T = 5 \cdot 10^{-3}$, hence $r = 0.125$).

As the present simulation is performed for very long times, the vorticity support is much bigger than the domain. The correction Eq. (7.2) on the bulk vortex sheet is therefore no longer valid, since the circulation inside the computational domain is not necessarily equal to zero. In order to circumvent this issue, the cylinder is here placed symmetrically with respect to the grid (up-down). As a consequence, we can simply enforce $\sum_{ij} (\omega_{\Delta\gamma})_{ij} h^2 = 0$.

This is not allowed for an asymmetric configuration, as it leads to a flow circulation $\sum_{ij} \omega_{ij} h^2$ that may become significantly different from zero, in the long term (even if the vorticity is entirely included in the domain). Indeed, the numerical errors may be biased on the top or the bottom side of the body, depending on the body position with respect to the grid, and the associated spurious generation of circulation has been observed to be self-sustaining (i.e. it is unstable). The correction Eq. (7.2) avoids this behavior and is applicable, as long as the entire vorticity is inside the computational domain.

The symmetry adopted in the present case ensures that any error made on the one side of the cylinder is also made on the other side and the drift of the flow circulation is prevented. In an asymmetric configuration where the vorticity is not entirely included in the domain, one could think of tracking the circulation that leaves the domain, as mentioned in Section 7.1.2. Hence, Eq. (7.2) would remain valid on condition that the tracked circulation is also accounted for. We leave this as a subject for future investigation.

Let us now come back to the topic of this section. Starting from any initial condition, the wake naturally becomes unstable after a certain time and the flow enters the well-known regime of "vortex shedding". The time evolution of the flow is then dominated by a single frequency f that is further characterized by the non-dimensional "Strouhal number"

$$St \triangleq \frac{U_\infty f}{D} ,$$

where D is the cylinder diameter.

The vorticity field is shown in Fig. 7.18 at various times during the transition phase towards the established vortex shedding regime. The successive stages correspond to a sampling of approximately twice the shedding period. The

actual oscillation frequency in the transition phase does not match the Strouhal number yet, since one may observe that the wake is slightly shifted between two consecutive stages.

The force computation based on the flow linear impulse is also no longer valid, for the same reason as above (the vorticity is not entirely included in the computational domain). The approach from Noca et al. [96, 97] is therefore adopted, as was already the case in Chapter 2. It results from a force balance over a control volume V_{CV} including the body $\partial\Omega_b$ (and with outer boundary S_{CV}) and it only requires computing surface integrals over S_{CV} and $\partial\Omega_b$ (i.e. contour integrals in 2-D). Furthermore, it does not involve the pressure, which makes the approach well-suited for vortex methods. The force $\mathbf{F}$ applied on the body reads

$$\begin{aligned}
\frac{\mathbf{F}}{\rho} \;=\; & -\frac{1}{\mathcal{N}-1}\frac{d}{dt}\int_{S_{\mathrm{CV}}+\partial\Omega_b}\big((\mathbf{u}\cdot\mathbf{x})\mathbf{n}-\mathbf{u}(\mathbf{x}\cdot\mathbf{n})+(\mathcal{N}-1)\mathbf{x}(\mathbf{u}\cdot\mathbf{n})\big)\,d\mathbf{x} \\
& +\int_{S_{\mathrm{CV}}}\left(\frac{1}{2}(\mathbf{u}\cdot\mathbf{u})\mathbf{n}-(\mathbf{n}\cdot\mathbf{u})\mathbf{u}\right)\,d\mathbf{x} \\
& -\frac{1}{\mathcal{N}-1}\int_{S_{\mathrm{CV}}}(\mathbf{n}\cdot\mathbf{u})(\mathbf{x}\times\boldsymbol{\omega})\,d\mathbf{x}+\frac{1}{\mathcal{N}-1}\int_{S_{\mathrm{CV}}}(\mathbf{n}\cdot\boldsymbol{\omega})(\mathbf{x}\times\mathbf{u})\,d\mathbf{x} \\
& +\frac{1}{\mathcal{N}-1}\int_{S_{\mathrm{CV}}}\mathbf{x}\times(\mathbf{n}\times\nabla\cdot\mathbf{T})\,d\mathbf{x}+\int_{S_{\mathrm{CV}}}\mathbf{n}\cdot\mathbf{T}\,d\mathbf{x}, \\
& +\frac{1}{\mathcal{N}-1}\frac{d}{dt}\int_{\partial\Omega_b}\mathbf{x}\times(\mathbf{n}\times\mathbf{u})\,d\mathbf{x}-\int_{\partial\Omega_b}\mathbf{n}\cdot\mathbf{u}\,d\mathbf{x}\,,
\end{aligned}$$

where $\mathbf{T}=\nu\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)$ and $\mathcal{N}=2$ is the number of dimensions. All contour integrals over $\partial\Omega_b$ vanish, since $\mathbf{u}=0$ at the wall and, in 2-D, we have $\mathbf{n}\cdot\boldsymbol{\omega}=0$ which further cancels the fourth integral. The present control volume is represented in Fig. 7.18.

Fig. 7.19 shows the resulting lift and drag coefficients (C_L and C_D, the associated definitions have been provided in Section 7.2.1) as a function of the time T and Table 7.3 provides the time averaged drag, the amplitude of the lift coefficient ($\Delta C_L \triangleq 0.5(\max(C_L)-\min(C_L))$) and the Strouhal number, along with some reference results from the literature. Despite the fact that the computational domain is quite short (13 diameters downstream of the cylinder) and that much of the circulation is lost, the three diagnostics are in very good agreement with those of the literature.

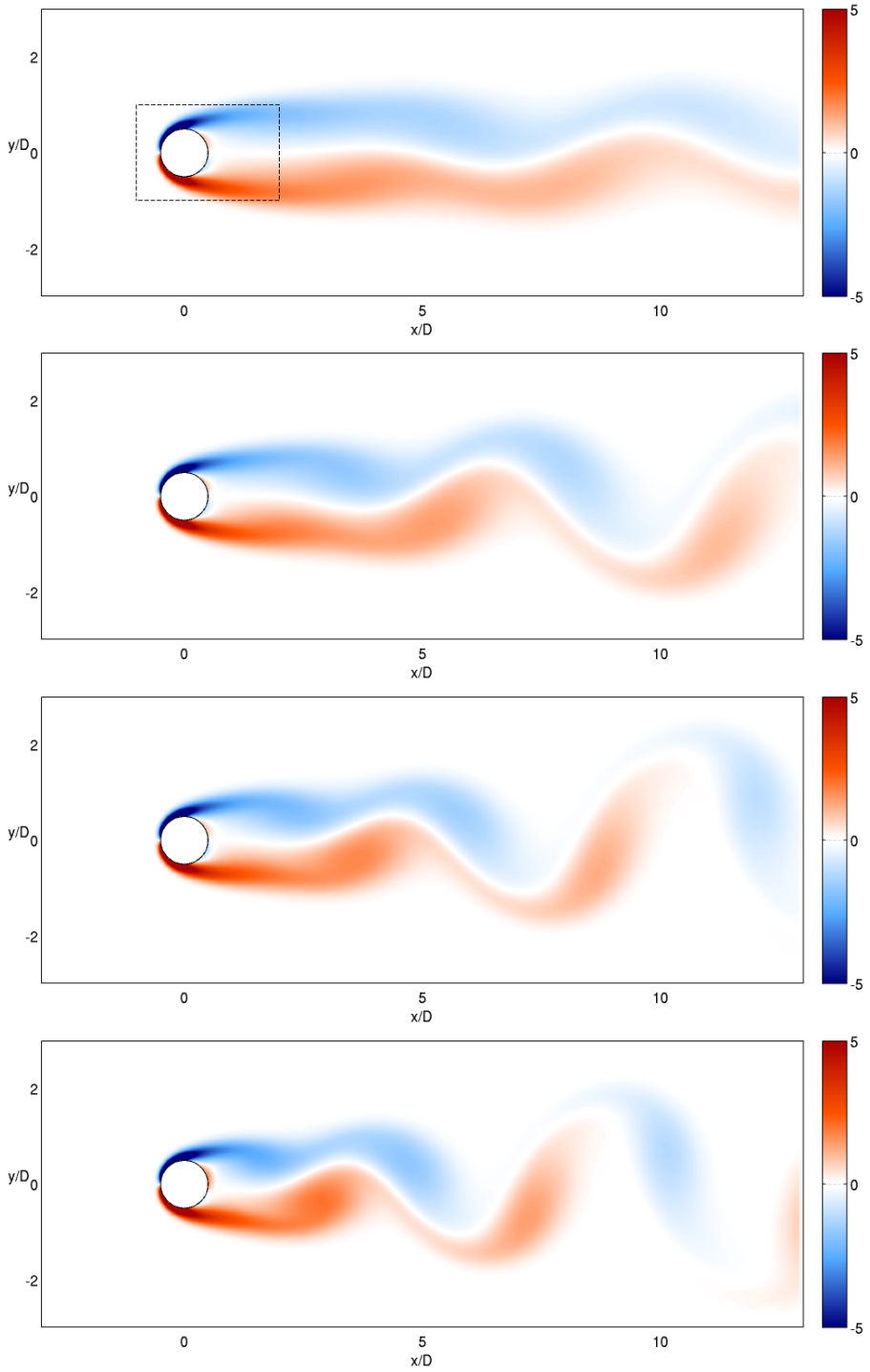


Figure 7.18: *Vorticity field $\omega D/U_\infty$ for the flow past a cylinder at $Re = 100$ (top to bottom : $T = 140, 152, 164$ and 176). In the top figure, the control volume V_{CV} for the force computation [96, 97] is also represented (dashed box).*

	$\overline{C}_D$	ΔC_L	St
Present	1.34	0.34	0.166
Rossinelli et al. [109]	1.35	0.33	0.166
Kevlahan et al. [67]	1.35	0.27	0.168
Shiels et al. [114]	1.33	0.30	0.167

Table 7.3: *Time averaged drag ($\overline{C}_D$), lift amplitude (ΔC_L) and Strouhal number for the flow past a cylinder at $Re = 100$; comparison between the present approach and reference results from the literature.*

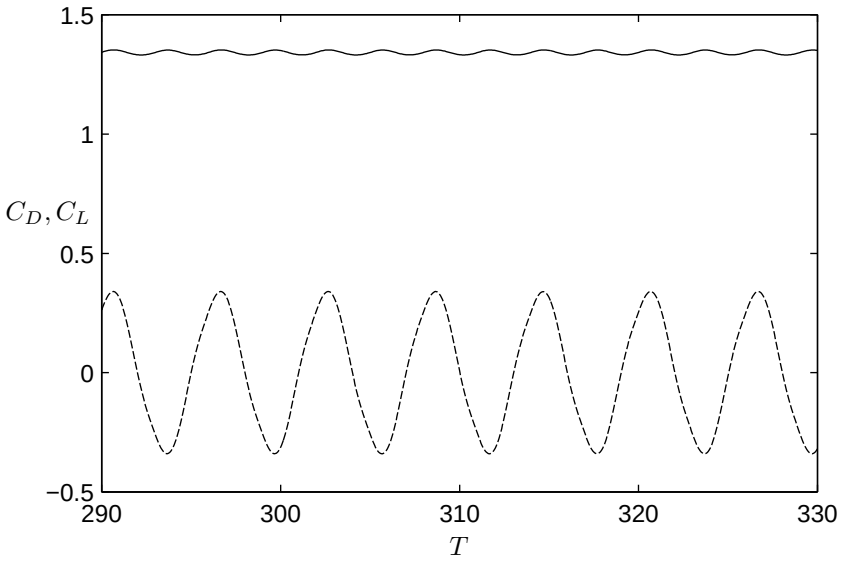


Figure 7.19: *Lift and drag coefficients (C_L, dashed; C_D, solid) for the flow past a cylinder at $Re = 100$.*

7.3 Accounting for an outflow condition

As a perspective of future development, one could use an "outflow condition" instead of simply neglecting the vorticity that leaves the domain. It is obtained by accounting for an image vorticity field that corresponds to an even symmetry across the outflow plane of the actual vorticity field in Ω_{comp}, hence $\partial\omega/\partial x = 0$ at $x = x_R$ in the present case (leading also to $\partial u/\partial x = 0$ and $v = 0$). This coincides with the "through flow plane" used in [28] for space-developing simulations (yet, one would then still need to track the "lost circulation" for the correction Eq. (7.2), especially for bodies with a non-negligible lift component).

Considering the developments made in Chapter 4, handling such a through flow plane is compatible with the present Poisson solver. First, note that the symmetry on ω further entails that $\partial\Psi/\partial x = 0$ at the outflow plane $x = x_R$. Hence, at the outflow plane, the Dirichlet boundary condition from Chapter 4 must be replaced by the Neumann condition $\partial\Psi/\partial n = 0$ for all sub steps of the algorithm. As a consequence, there is no longer an artificial vortex sheet γ_{comp} at $x = x_R$ after the first step of the algorithm computing the solution using $\Psi = 0$ on the outer boundary of the domain (except for the outflow plane, where $\partial\Psi/\partial n = 0$ must now be imposed). Yet, $\gamma_{comp} \neq 0$ on all 3 other edges of the outer boundary. Finally, one must then take into account the "images" of γ_{comp} and $\Delta\gamma$ across $x = x_R$ (in Chapter 4, the latter was noted γ) while one performs the iterative computation of the Dirichlet boundary condition using the fast summation technique. The obtained streamfunction consequently satisfies the aforementioned outflow condition. Fig. 7.20 sums up this procedure and further illustrates the spatial configuration of the source images. Note that adapting the computation of the diffusion and the particle-mesh interpolation is straightforward (cfr. the tools from Section 3.2 so as to account for $\partial\omega/\partial n = 0$ at $x = x_R$).

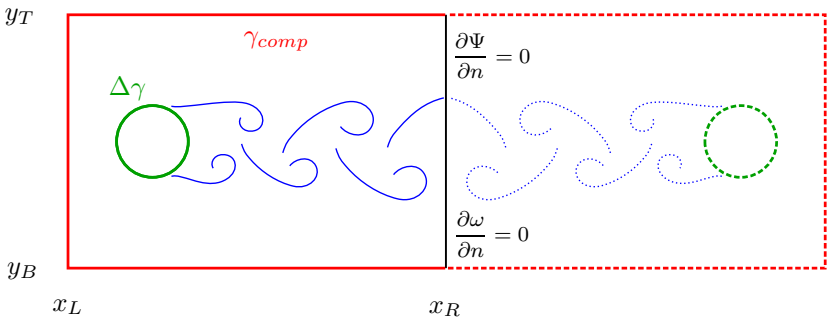


Figure 7.20: *Illustration of the configuration that is required when accounting for an outflow condition: both ω and Ψ are even across $x = x_R$ and the vortex sheets γ_{comp} and $\Delta\gamma$ have a symmetric counterpart across $x = x_R$.*

7.4 Conclusions

The present methodology allows to accurately perform the simulation of some challenging flows, such as the flow past an impulsively started solid body. The results agree quite well with those of purely Lagrangian and panel-based VP

(vortex particle) approaches [73, 102, 103]. In addition, no time-filtering is required in the present case for the drag prediction, contrary to what is done in [102], and no oscillations are observed, except for the very first time steps after the impulsive acceleration. Considering the issues encountered in Chapter 2, for the development of a hybrid Eulerian-Lagrangian solver, this is quite an interesting feature.

Comparing the immersed interface solver with other existing VPM methods shows that the sharp treatment of the wall stands in contrast with penalized approaches [109, 50], where the near-wall flow is smeared due to the force mollification. From another point of view, the present approach also provides a more consistent alternative to panel-based VPM methods [83], as all different differential operations near the wall are performed on the grid using similar stencils. Indeed, the operations contributing to the inconsistency of the panel-based VPM solver have all been replaced by their immersed interface counterpart:

- No-through flow Poisson solver (Chapter 4).
- Near-wall diffusion and no-slip enforcement (Chapter 5).
- Particle-mesh interpolation near the wall (Chapter 6).

The application of the tools that are presented here (or a subset of them) could therefore maybe contribute to the successful development of a fully operational hybrid Eulerian-Lagrangian solver, which remains a prerequisite for the simulation of high Reynolds number wall-bounded flows using vortex methods.

Yet, this assertion has not been verified and some issues remain. The present method appears to be less robust for under resolved cases, which is a typical feature of purely grid-based methods. As a matter of fact, the bulk vortex sheet can become very noisy in the starting phase. When the grid resolution is not sufficient (or when the time step is too small), the near-wall diffusion is not able to provide enough correction to the solution and the latter blows up.

Considering the hybrid solver, this is quite problematic, since the aim is to couple the near-wall solver with an under resolved VPM method. A possible remedy for this, among others, could consist in favoring the conservation of the circulation during the different computational steps. However, this probably would come at the cost of slightly reducing the near-wall accuracy.

It would also be interesting to consider the effect of reducing the redistribution frequency. It is a priori not clear if this would improve or worsen the robustness with respect to under resolved flow configurations.

Furthermore, the computational time of the present approach is quite high and there is much space for improvement, especially for the Poisson solver. Fig. 7.21 shows a brief summary of the wall-clock time spent by the solver inside the different operations of the VPM approach. One observes that the Poisson solution clearly requires the highest computational time ("Advection"). This is all the more important that the call to the linear solver ("Advection: Hypre") actually uses 32 cores here, whereas all the other operations are performed sequentially. The "Biot-Savart (ext-ext)" item corresponds to the convolution of the vortex sheet γ_{comp} in order to compute the outer boundary condition and "Biot-Savart (int-ext)" represents the convolution of the inner sheet γ (see Chapter 4). The former takes more time because a direct summation is performed here, whereas the latter uses fast summation techniques. One also observes that the near-wall diffusion operation is also quite time consuming, as the number of sub-iterations is here $N_{sub} = 20$.

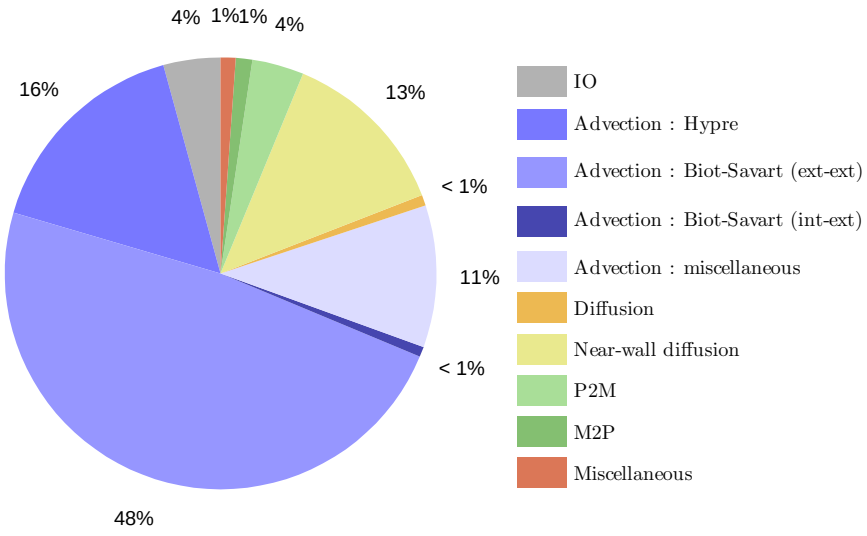


Figure 7.21: *Wall-clock time percentage for the different operations performed by the present immersed interface VPM method (for the test case from Section 7.2.3); all operations are sequential, apart from the linear solver inside the Poisson solver (Hypre) that uses 32 cores.*

Chapter 8

Numerical dispersion and dissipation errors of a 1-D redistributed Lagrangian method

As explained in Chapter 6, redistribution is required in vortex methods in order to remedy the flow induced distortion of the particles' positions. Redistributing (or equivalently *remeshing*) the particles consists in periodically replacing the set of distorted particles by a new set whose positions correspond to the underlying grid nodes. While it helps keeping a good representation of the vorticity field, this operation introduces some additional numerical errors to the solution.

In [39], an equivalence between remeshed particle methods and finite difference methods is highlighted in the framework of transport equations and similarities with the Lax-Wendroff scheme are observed, when remeshing occurs at every time step. This gives a hint about the nature of the numerical errors that are introduced. The link between remeshed particle methods and finite differences is furthermore exploited in [36] so as to devise TVD redistribution schemes for linear and non-linear transport equations, and the approach is extended to develop non-oscillatory schemes for large time steps [84], with the aim to reduce the produced numerical errors.

The present work mainly considers the performance of some classical redistribution schemes, in particular of the M_4' scheme derived by [93]. It has already been shown in [132] that the redistribution acts as a hyper-viscosity operator. In the present one-dimensional study, we show that, next to the numerical dissipation error, a dispersion error is also introduced. Both types of errors are quantified in the framework of a linear advection equation and the study is performed in Fourier space, similarly to what was originally done in [112] and [93], where the derivation of high order interpolation schemes was precisely based on the properties of their transforms, or also to [18], where isotropic and compact schemes were developed for alternative lattices.

8.1 Derivation of the numerical dissipation and dispersion errors

Consider the following one-dimensional linear advection equation

$$\frac{\partial u}{\partial t} + c\,\frac{\partial u}{\partial x} = 0\ , \tag{8.1}$$

with c a uniform advection velocity. The solution of this equation, in infinite or periodic domain, is

$$u(x,t) = U_0(x-ct) \triangleq u(x-ct,0)\ , \tag{8.2}$$

where $U_0(x) \triangleq u(x,0)$ is the initial condition. For the remainder of this study, let us furthermore consider a periodic domain $x \in \Omega \triangleq [0,L]$, i.e. $U_0(x-kL) = U_0(x)$ for any integer k and therefore $u(x-kL,t) = u(x,t)$.

Using N particles of position $x_p(t)$ $(p = 0,...,N-1)$ in Ω and carrying the function value $u_p(t) \triangleq u(x_p(t),t)$, we can write the Lagrangian discretization of this equation as

$$\begin{aligned} \frac{dx_p}{dt} &= c\ , \quad x_p(0) = ph \\ \frac{du_p}{dt} &= 0\ , \quad u_p(0) = u(ph,0)\ , \end{aligned} \tag{8.3}$$

with $h = L/N$ the uniform mesh spacing. For the simple hyperbolic equation Eq. (8.1), this discretization is actually equivalent to the classical *method of characteristics* where the solution is integrated along the characteristics (which are straight lines in the present case).

The exact solution of the set of ordinary differential equations Eqs. (8.3) can be computed as

$$\begin{aligned} x_p^{n+1} &= x_p^n + c\Delta t \\ u_p^{n+1} &= u_p^n \ , \end{aligned}$$

where $x_p^n \triangleq x_p(t^n)$, $u_p^n \triangleq u_p(t^n)$, Δt is the time step and n the time step index ($t^{n+1} \triangleq t^n + \Delta t$). The above time integration amounts to perform an explicit Euler integration (any other time integration is equivalent here and provides the exact solution, by definition). Eq. (8.3) is intrinsically stable, as its right-hand side is independent of the solution. The time step is therefore arbitrary and we get the exact solution at any time t, up to round off error.

While being trivial, Eq. (8.1) allows to shed some light on the fundamental spectral behavior of the redistribution operation, and these properties can be translated to more complex two- and three-dimensional problems, where the advection is far from being linear and the velocity far from being uniform, unlike the present case.

Formally, the redistribution consists in replacing, after a few time steps, say every $(n^r)^{\text{th}}$ time step, the distorted set of particles x_q^n carrying the value u_q^n (and that have moved according to the local velocity field) by a new set of particles whose positions $x_p^{n,r} \triangleq ph$ coincide with the underlying grid nodes. Without loss of generality, we set $n^r = 1$ and the redistribution period $\Delta t^r \triangleq n^r \Delta t$ is simply Δt, since the time integration is anyway exact here. A measure of the redistribution frequency is given by the following CFL-like number

$$\beta \triangleq \frac{c\,\Delta t}{h} \ ,$$

that can be interpreted as the number of cells the solution has traveled before being redistributed. The sequence of operations we wish to study therefore consists in first advecting the particles from $x_q^n = qh$ to $x_q^{n+1} = (\beta + q)h$ (no numerical error is introduced during this step) and, secondly, in redistributing the freshly moved particles onto the grid as $x_p^{n+1,r} = ph$. Following Eq. (6.1), the function value carried by the new particle at $x_p^{n+1,r}$ is computed as

$$u_p^{n+1,r} = \sum_{q=0}^{N-1} u_q^{n+1}\, v_{per}\left(x_p^{n+1,r} - x_q^{n+1}\right) \ , \tag{8.4}$$

with $v_{\text{per}}(x) \triangleq \sum_{m=-\infty}^{+\infty} v(x - mL)$ the periodized redistribution kernel based on the interpolation scheme $v(x) \triangleq w(x/h)$.

Consider for example the classical third order scheme $w(\xi) = M_4'(\xi)$ derived by Monaghan [93] (see Chapter 6 and Appendix C for more details about the redistribution and the associated kernels).

At this point, let us introduce some new notation that is better suited for the following study performed in Fourier space. For the remainder of this Chapter, a *discrete* function will be noted as $f[\cdot]$ (with an integer as the argument) and $f(\cdot)$ stands for a *continuous* function (with a real value as the argument). Moreover, let us here consider one time step, going from $n = 0$ to $n = 1$ (in Section 8.2.2 we will also consider several successive steps). Hence, the values $u^0[p] \triangleq u_p^0$ and $u^{1,r}[p] \triangleq u_p^{1,r}$ are formally discrete functions of the particle index p on the grid $x_p \triangleq ph$. Notice that u_p^1 resides on a grid that is shifted with respect to the original grid x_p and we do not introduce the new notation here in order to avoid any ambiguity concerning the underlying grid definition. As a consequence, Eq. (8.4) is equivalently described by a discrete convolution on the grid x_p

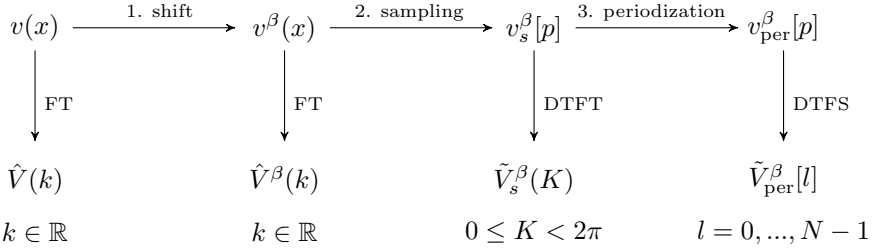
$$u^{1,r}[p] = \sum_{q=0}^{N-1} u^0[q]\ v_{\text{per}}^{\beta}[p-q]\ , \tag{8.5}$$

with $v_{\text{per}}^{\beta}[p] \triangleq v_{\text{per}}\left((p-\beta)h\right)$ the discretized and periodized redistribution scheme that is furthermore shifted in space by an amount βh.

The DTFS (*Discrete-Time Fourier Series* for discrete and periodic functions) of this convolution is

$$\tilde{U}^{1,r}[l] = N\tilde{V}_{\text{per}}^{\beta}[l]\ \tilde{U}^0[l] = \tilde{\rho}^r[l]\ \tilde{U}^0[l] \qquad l = 0, ..., N-1\ ,$$

with $\tilde{\rho}^r[l] \triangleq N\tilde{V}_{\text{per}}^{\beta}[l]$ the amplification factor due to the redistribution and $\tilde{F}[l] \triangleq \text{DTFS}\{f[p]\}$ the corresponding Fourier transform defined in Appendix C. In order to obtain $\tilde{V}_{\text{per}}^{\beta}[l]$ as a function of $\hat{V}(k) \triangleq \text{FT}\{v(x)\}$ (*Fourier Transform* for non periodic continuous functions with $k = 2\pi/\lambda$ the wavenumber and λ the wavelength; see Appendix C), we have to perform the following sequence of operations in Fourier space (see also Fig. 8.1):



with $v^\beta(x) \triangleq v(x - \beta h)$, $v_s^\beta[p] \triangleq v((p-\beta)h)$ and DTFT the *Discrete-Time Fourier Transform* for non periodic discrete functions (see Appendix C).

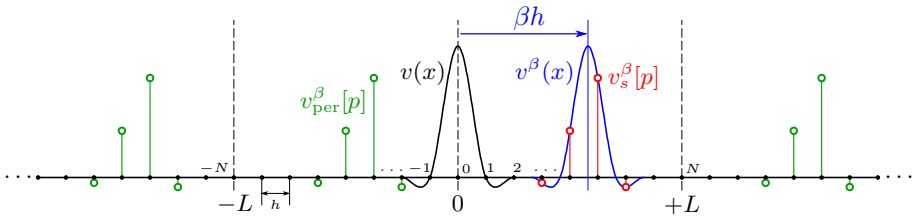


Figure 8.1: *Sketch of the sequence of operations (here in the physical space) required in order to obtain the discrete function $v_{per}^\beta[p]$.*

The details of these operations are

1. **Shift** : The first operation consists in a shift βh of the redistribution kernel $v(x)$. The operation in Fourier space is

$$\hat{V}^\beta(k) = \exp(-jk\beta h)\, \hat{V}(k) \qquad k \in \mathbb{R}\ .$$

2. **Sampling** : Sampling $v^\beta(x)$ at rate h yields the following distribution

$$v_s^\beta(x) \triangleq \sum_{m=-\infty}^{+\infty} v^\beta(mh)\, \delta(x - mh)\ .$$

   Its FT is the periodization of the spectrum $\hat{V}^\beta(k)$ (with a "spectral period" $2\pi/h$)

$$\hat{V}_s^\beta(k) = \frac{1}{h} \sum_{m=-\infty}^{+\infty} \hat{V}^\beta\left(k - \frac{2\pi m}{h}\right) \qquad k \in \mathbb{R}\ .$$

   Making the link between the FT $\hat{V}_s^\beta(k)$ of the sampled function and the DTFT $\tilde{V}_s^\beta(K)$ of the associated discrete function $v_s^\beta[p]$ and using the sampling relation $k = K/h$, we obtain

$$\tilde{V}_s^\beta(K) = \hat{V}_s^\beta\left(\frac{K}{h}\right) = \frac{1}{h} \sum_{m=-\infty}^{+\infty} \hat{V}^\beta\left(\frac{K - 2\pi m}{h}\right)\ .$$

   One can see here that aliasing will occur because the redistribution kernels we are working with have an infinitely broad spectrum $\hat{V}(k)$.

3. **Periodization** : The periodization (with a discrete period N) of the shifted and discrete redistribution scheme can also be rewritten as $v_{\rm per}^{\beta}[p] = \sum_{m=-\infty}^{+\infty} v_s^{\beta}[p-mN]$. As a consequence, the spectrum $\tilde{V}_s^{\beta}(K)$ is sampled at a rate $\Delta K = 2\pi/N$ or equivalently $K = l\Delta K = 2\pi l/N$. The link between the periodized DTFT $\tilde{V}_s^{\beta}(K)$ and the associated DTFS $\tilde{V}_{\rm per}^{\beta}[l]$ finally yields

$$\begin{aligned}\tilde{V}_{\rm per}^{\beta}[l] &= \frac{1}{N}\tilde{V}_s^{\beta}\left(\frac{2\pi l}{N}\right) = \frac{1}{N}\hat{V}_s^{\beta}\left(\frac{2\pi l}{L}\right) \\ &= \frac{1}{L}\sum_{m=-\infty}^{+\infty}\hat{V}^{\beta}\left(\frac{2\pi}{h}\left(\frac{l}{N}-m\right)\right) .\end{aligned}$$

The amplification factor is then

$$\tilde{\rho}^r[l] = \exp\left(-\frac{j2\pi l\beta}{N}\right)\sum_{m=-\infty}^{+\infty}\exp\left(j2\pi\beta m\right)\ \hat{W}\left(2\pi\left(\frac{l}{N}-m\right)\right), \quad (8.6)$$

because $\hat{V}(k) = h\hat{W}(kh)$, where $\hat{W}(k)$ is the *Fourier transform* of $w(\xi)$.

In order to quantify the error made by the redistribution procedure, let us consider the exact amplification factor. Using the DTFS coefficients $\tilde{U}_l(t)$ defined by $\tilde{U}_l(t^n) \triangleq \tilde{U}^n[l]$ for discrete time steps, the exact grid solution of Eq. (8.1) for a given set of initial modes $\tilde{U}_l(0)$ may be rewritten as a *Fourier Series* (FS; see Appendix C)

$$\begin{aligned}u(x,t) &= \sum_{l=-N/2+1}^{N/2}\tilde{U}_l(t)\ \exp(jk_l x) \\ &= \sum_{l=-N/2+1}^{N/2}\tilde{U}_l(0)\ \exp(jk_l(x-ct)) ,\end{aligned}$$

using the solution Eq. (8.2) and as the physical wavenumber is related to the mode l by $k_l = 2\pi l/L$, according to the above relations established between the different transforms. The exact amplification factor associated to the grid, and going from $t = 0$ to $t = \Delta t$, is therefore

$$\tilde{\rho}[l] = \exp(-jk_l c\Delta t) = \exp(-jk_l\beta h) = \exp\left(-\frac{j2\pi l\beta}{N}\right) ,$$

which is also, not surprisingly, equivalent to a shift β applied to a discrete

function. Moreover, one may recognize the definition of $\tilde{\rho}[l]$ in the term multiplying the infinite sum of Eq. (8.6). Hence, the spectrum $\hat{W}^{\star}(k)$ of an ideal redistribution scheme should satisfy, for all modes l:

$$\tilde{\gamma}^{\star}[l] \triangleq \sum_{m=-\infty}^{+\infty} \exp\left(j2\pi\beta m\right)\ \hat{W}^{\star}\left(2\pi\left(\frac{l}{N}-m\right)\right) = 1\ . \tag{8.7}$$

For a given redistribution scheme, any deviation from this equality results in numerical dissipation and/or dispersion errors. Following the classical analysis of those errors, the modulus of the amplification factor $|\tilde{\rho}^{r}[l]|$ quantifies the numerical dissipation error associated to the mode l, whereas the phase $\phi^{r}[l] \triangleq \arg(\tilde{\rho}^{r}[l])$ is related to the dispersion error.

Alternatively, one may also obtain $\tilde{\rho}^{r}[l]$ by performing directly the convolution of Eq. (8.5) (i.e. by redistributing $u^{1}[q]$ on the grid in order to get $u^{1,r}[q]$) and then computing $\tilde{U}^{1,r}[l]$ as DTFS$\{u^{1,r}[q]\}$. Considering the impulse response to $u^{0}_{\delta}[q] = \delta[q]$ (a single unitary particle at $q = 0$), we have $\tilde{U}^{0}_{\delta}[l] = 1/N$ and hence $\tilde{\rho}^{r}[l] = \tilde{U}^{1,r}_{\delta}[l]/\tilde{U}^{0}_{\delta}[l] = N\ \tilde{U}^{1,r}_{\delta}[l]$. This approach is strictly equivalent to the previous derivation but the former allows a more thorough analysis of the induced numerical errors, as it suggests writing the condition Eq. (8.7) that an ideal redistribution scheme must satisfy. Yet, the latter is still applicable when $\hat{W}(k)$ is not known.

As a side note, since $v^{\beta}_{\mathrm{per}}[p]$ is real, $\tilde{\rho}^{r}[l]$ is Hermitian symmetric in addition to being periodic and we therefore only consider the modes $l = 0, ..., N/2$ (by furthermore assuming that N is even). As expected, the smallest wavenumber is $k_0 = 0$ and the highest wavenumber on the grid is given by $k_{N/2} = \pi/h$ ("flip-flop mode").

The exact amplification factor has a unitary amplitude $|\tilde{\rho}[l]| = 1$ and its phase is given by

$$\phi[l] \triangleq \arg(\tilde{\rho}[l]) = -2\pi l\beta/N = -k_l h\ \beta\ . \tag{8.8}$$

Considering the principal branch defined by $-\pi \leq \mathrm{Arg}(\tilde{\rho}[l]) < \pi$, one may observe that branching occurs for the phase at the highest wavenumbers when $|\beta| > 1$. Hence, the computation of the amplification factor phase $\phi^{r}[l]$ has to account for the number $l\beta/N$ of wavelengths $\lambda_l \triangleq L/l$ the mode l has traveled

before being redistributed, and we have

$$\phi^r[l] = \mathrm{Arg}(\tilde{\gamma}^r[l]) - 2\pi l\beta/N = \mathrm{Arg}(\tilde{\gamma}^r[l]) - k_l h\ \beta\ ,$$

with $\tilde{\rho}^r[l] \triangleq \tilde{\rho}[l]\ \tilde{\gamma}^r[l]$. The maximum phase error is thus implicitly restricted to π for all modes. This restriction is however sensible, even for the mode $k_l h = \pi$ where a phase shift of π corresponds to one cell.

The associated dispersion error may be recast in its classical form as a modified wavenumber k^* by writing the redistributed solution as

$$\begin{aligned} u^r(x,\Delta t) &= \sum_{l=-N/2+1}^{N/2} \tilde{U}^{1,r}[l]\ \exp(jk_l x) \\ &= \sum_{l=-N/2+1}^{N/2} \tilde{U}^{0}[l]\ |\tilde{\rho}^r[l]| \exp(j\phi^r[l])\ \exp(jk_l x) \\ &\triangleq \sum_{l=-N/2+1}^{N/2} \tilde{U}^{0}[l]\ |\tilde{\rho}^r[l]|\ \exp(j(k_l x - k_l^* c\Delta t))\ , \end{aligned}$$

and we thus obtain the following expression for the dispersion error

$$k_l^* h = -\frac{\phi^r[l]}{\beta}\ . \tag{8.9}$$

Similarly, an effective diffusion coefficient ϵ^* can be defined by the following relations

$$\begin{aligned} |\tilde{\rho}^r[l]| &\triangleq \exp(-\epsilon_l^* k_l^2 \Delta t) \\ \frac{\epsilon_l^*}{ch} &= -\frac{\log|\tilde{\rho}^r[l]|}{\beta(k_l h)^2}\ . \end{aligned} \tag{8.10}$$

Hence, every redistributed mode $u_l^r(x,\Delta t) \triangleq \tilde{U}^{1,r}[l]\exp(jk_l x)$ can be seen as the solution of the following modified equation for a given β

$$\frac{\partial u}{\partial t} + c\left(\frac{k_l^* h}{k_l h}\right)\ \frac{\partial u}{\partial x} = \epsilon_l^* \frac{\partial^2 u}{\partial x^2}\ ,$$

which actually corresponds to a convection-diffusion equation.

For the remainder of this Chapter, the notation for the mode l will be dropped, unless stated otherwise and, any reference to the wavenumber k implicitly means k_l.

8.2 Results

The numerical dissipation error for the M_4' redistribution kernel is shown in Fig. 8.2, Fig. 8.3 and Fig. 8.4 for various values of β (the infinite sum in Eq. (8.6) is truncated to $m = -100, ..., 100$). Not surprisingly, $i + \beta$ induces the same error on $|\tilde{\rho}|$ as β for all $i \in \mathbb{Z}$, as the kernel is simply shifted by i cells with respect to its initial location β. Hence, the redistributed particle field preserves its "shape". Due to the kernel symmetry, $i - \beta$ also leads to the same error on $|\tilde{\rho}|$ as β. As can be seen in Fig. 8.2, no dissipation error is made when β is an integer, whereas the scheme is most dissipative when $\beta = i + 0.5$. The "flip-flop" mode, $kh = \pi$, completely vanishes for this value of β and the effective diffusion coefficient ϵ^* is thus infinite (see Fig 8.3). This is consistent with the observation that this mode can only appear in the form of a cosine, since $\Im\{\tilde{F}[N/2]\} = 0$ for any real function $f[p]$ ($\Im\{\tilde{F}[l]\}$ is odd and periodic of period N). The value $\beta = i + 0.5$ thus transforms the cosine into a sine with zero amplitude.

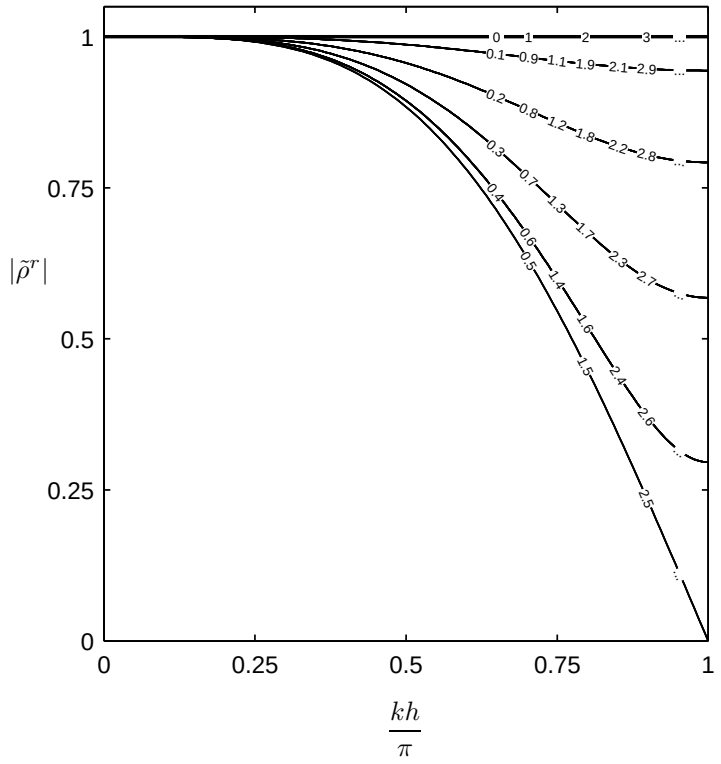


Figure 8.2: *Numerical dissipation error $|\tilde{\rho}|$ for the redistribution scheme M_4' for different values of $\beta \geq 0$ (the curves are labeled with the associated value of β).*

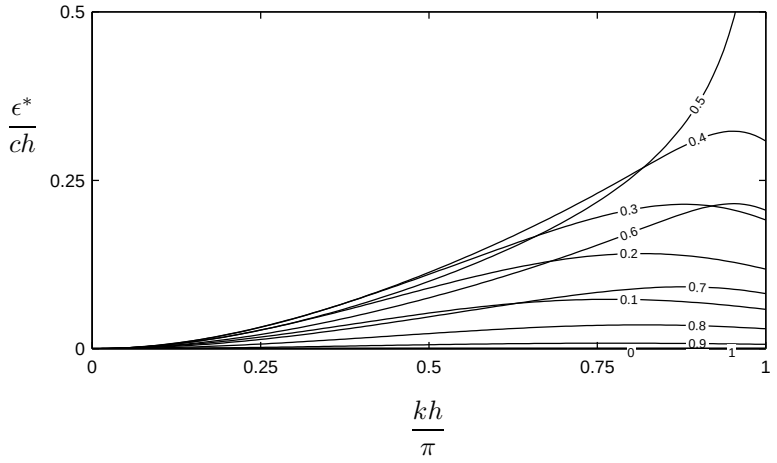


Figure 8.3: *Numerical dissipation error ϵ^* for the redistribution scheme M_4' for different values of β with $0 \leq \beta \leq 1$ (the curves are labeled with the associated value of β).*

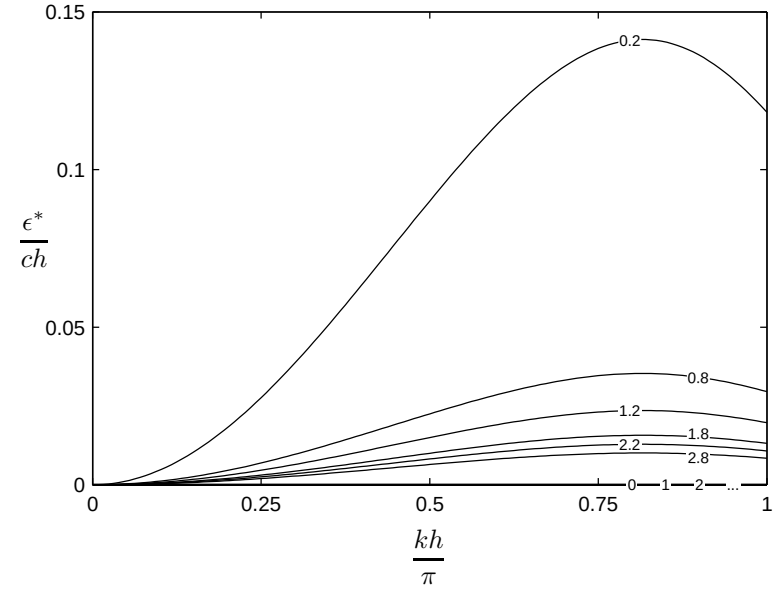


Figure 8.4: *Numerical dissipation error ϵ^* for the redistribution scheme M_4' when $\beta = 0.2 + i$ and $\beta = 0.8 + i$ with i an integer (thin solid lines); The curves are labeled with the associated value of β.*

The dispersion error is best understood when considering the associated phase error, as shown in Fig. 8.5 for $0 \leq \beta \leq 1$. The exact phase (dotted lines) is linear in k_l and also in β, according to Eq. (8.8). Another consequence of the previous observation about the "flip-flop" mode $kh = \pi$ is that it cannot

move arbitrarily with respect to the grid and its phase is therefore necessarily a multiple of π, as can be seen in Fig. 8.5. For $\beta \neq 0.5$, the phase is continuous as a function of the wavenumber k, and this leads to an all the more important phase error when $kh \to \pi$, as classically observed for any grid based method. One should observe that no dispersion error is made when β is a multiple of 0.5. The apparently inconsistent fact that no dispersion error is made for $kh = \pi$ when $\beta = 0.5$, is explained by the zero amplitude of the mode, according to the previously discussed dissipation error.

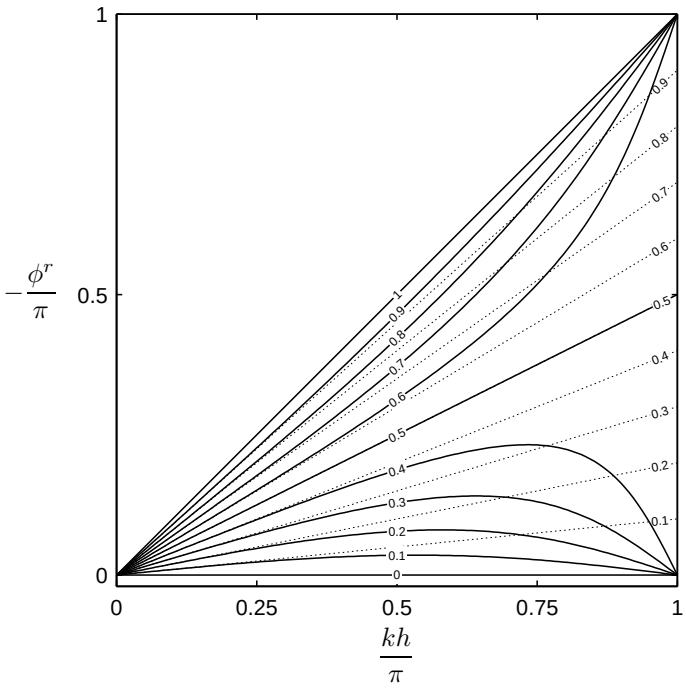


Figure 8.5: *Phase of the amplification factor $\tilde{\rho}^r$ for the redistribution scheme M_4' when $0 \leq \beta \leq 1$ (solid lines); phase of the exact amplification factor $\tilde{\rho}$ (dotted lines). The curves are labeled with the associated value of β.*

Based on these observations, we have the following expressions for the limits around $\beta = 0.5$

$$\lim_{\beta \to 0.5^-} \phi^r(k) = \begin{cases} \phi^r_{0.5}(k) & \text{if } kh \neq \pi \\ 0 & \text{if } kh = \pi \end{cases}$$

$$\lim_{\beta \to 0.5^+} \phi^r(k) = \begin{cases} \phi^r_{0.5}(k) & \text{if } kh \neq \pi \\ -\pi & \text{if } kh = \pi \end{cases},$$

where $\phi^r_{0.5}$ is the phase associated to $\beta = 0.5$.

The modified wavenumber k^* is shown in Fig. 8.6. Interestingly, all modes $k > 0$ move faster than the exact solution for $0.5 < \beta < 1$, whereas they are all slower for $0 < \beta < 0.5$. We can also compute the limits around $\beta = 0.5$ for the modified wave numbers as follows

$$\lim_{\beta \to 0.5^-} k^* h = \begin{cases} k^*_{0.5} h & \text{if } kh \neq \pi \\ 0 & \text{if } kh = \pi \end{cases}$$

$$\lim_{\beta \to 0.5^+} k^* h = \begin{cases} k^*_{0.5} h & \text{if } kh \neq \pi \\ 2\pi & \text{if } kh = \pi \end{cases} ,$$

as $\lim_{\beta \to 0.5^+}(k^* h) = -\lim_{\beta \to 0.5^+}(\phi^r/\beta) = -2 \lim_{\beta \to 0.5^+} \phi^r$ ($k^*_{0.5} h$ is the modified wave number associated to $\beta = 0.5$).

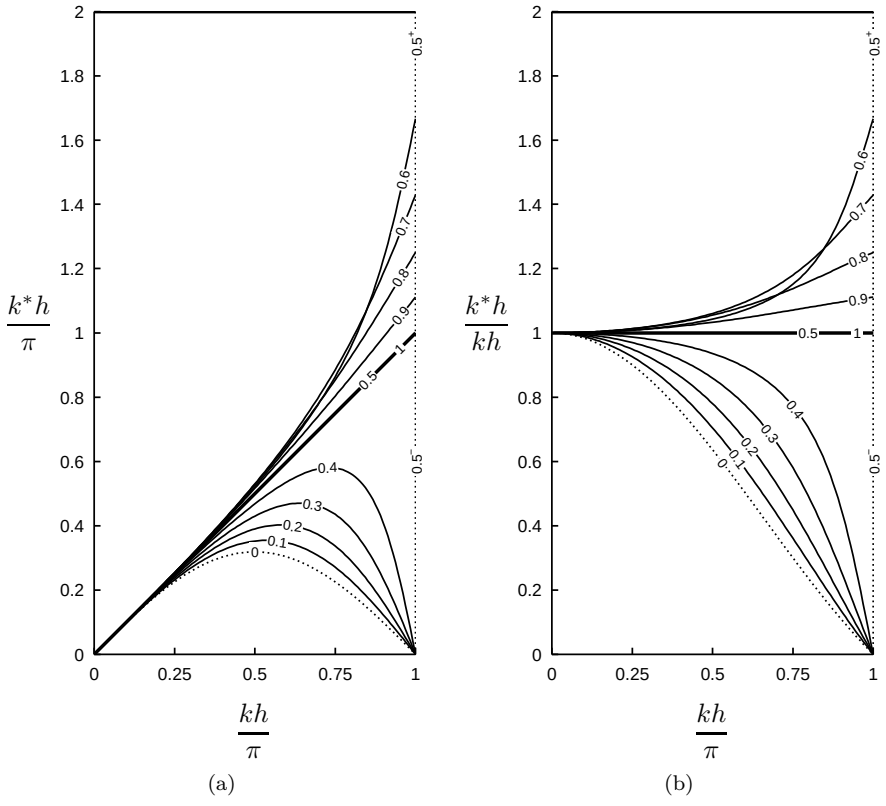


Figure 8.6: *Numerical dispersion error for the redistribution scheme M'_4 expressed as a modified wavenumber k^* when $0 \leq \beta \leq 1$ (thin solid lines for $\beta > 0$ and dotted lines for the limits $\beta \to 0$, $\beta \to 0.5^-$ and $\beta \to 0.5^+$); exact wavenumber k (thick solid line). The curves are labeled with the associated value of β.*

The numerical dissipation and dispersion properties of a redistributed Lagrangian method become more interesting when $\beta > 1$, as expected, since the redistribution frequency is then lowered accordingly (the particles travel further before being redistributed). However, this statement implicitly considers a *relative* measure of the underlying numerical errors. The effective diffusion coefficient ϵ^* and the modified wave number k^* fulfill this requirement, as they are both normalized by β (see Eq. (8.10) and Eq. (8.9)). As shown in Fig. 8.4 and Fig. 8.7, for an increasing integer i, the value $i + \beta$ leads to a decreasing dissipation error ϵ^*, as well as to a decreasing dispersion error k^*.

On the contrary, the amplitude $|\tilde{\rho}^r|$ gives an *absolute* measure of the dissipation error and the phase ϕ^r represents an *absolute* measure of the dispersion error. Hence, a shift of $i + \beta$ induces the same error $(\phi^r - \phi)$ as does a shift of β (the error never exceeds $\pi/2$, see Fig. 8.8), similarly to what was already observed for the dissipation error on $|\tilde{\rho}^r|$ (see Fig. 8.2). Moreover, due to the symmetry of the redistribution kernel, $(\phi^r - \phi)$ is odd with respect to β. In short, $|\tilde{\rho}^r|$ and $(\phi^r - \phi)$ are both periodic functions of β (with period 1) enjoying some symmetry properties (recall that $|\tilde{\rho}^r|$ is *even* with respect to β).

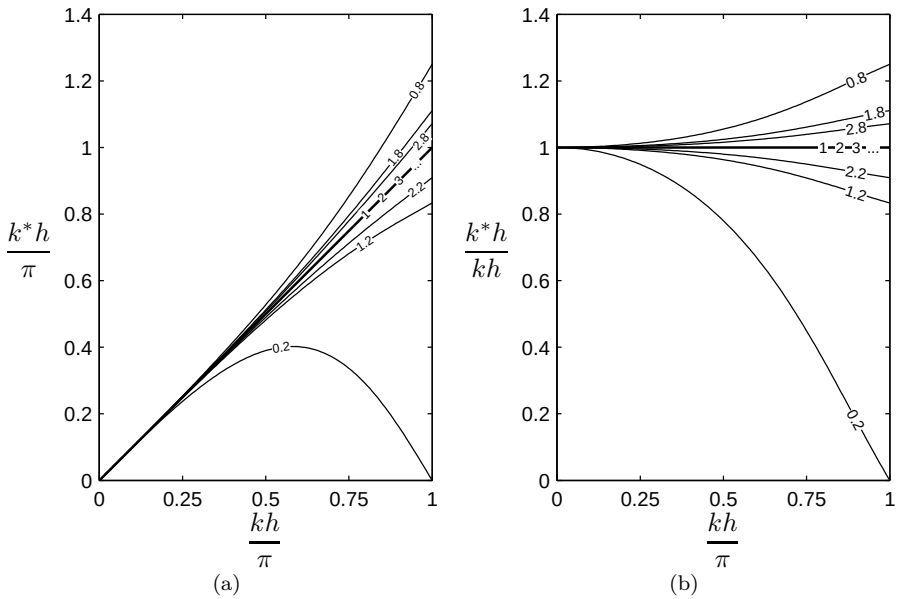


Figure 8.7: *Numerical dispersion error for the redistribution scheme M_4' expressed as a modified wavenumber k^* when $\beta = 0.2 + i$ and $\beta = 0.8 + i$ with i an integer (thin solid lines); exact wavenumber k (thick solid line). The curves are labeled with the associated value of β.*

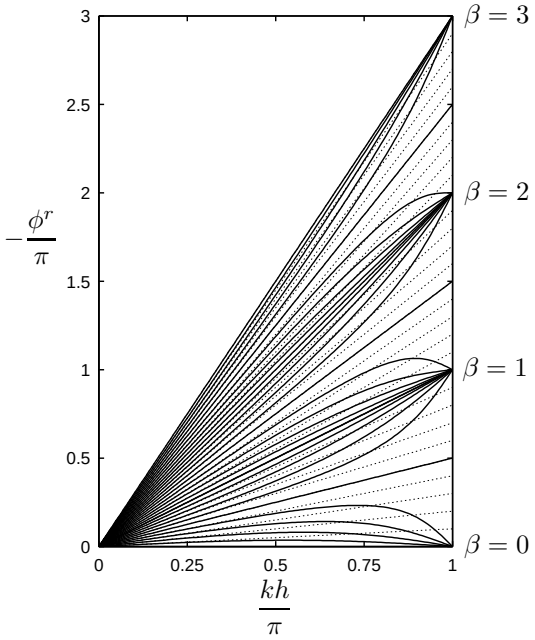


Figure 8.8: *Phase of the amplification factor $\tilde{\rho}^r$ for the redistribution scheme M_4' when $0 \leq \beta \leq 3$ (solid lines); phase of the exact amplification factor $\tilde{\rho}$ (dotted lines). The values of β to the right of the figure are given for $kh = \pi$.*

8.2.1 Averaged numerical errors

In a more general framework, where the advection velocity is not uniform, the average dispersion and dissipation errors may be computed by assuming a certain distribution $\sigma(\beta)$ in the flow. We define the following weighted averages

$$\begin{aligned} \overline{\phi^r - \phi} &\triangleq \frac{1}{\bar{\beta}} \int_0^{\beta_{\max}} \sigma \, (\phi^r - \phi) \, d\beta' \\ \overline{|\tilde{\rho}^r|} &\triangleq \frac{1}{\bar{\beta}} \int_0^{\beta_{\max}} \sigma \, |\tilde{\rho}|^r \, d\beta' \end{aligned}$$

with $\bar{\beta} \triangleq \int_0^{\beta_{\max}} \sigma \, d\beta'$.Strictly speaking, $\sigma(\beta)$ actually corresponds to a velocity distribution in *time*. Yet, we make the assumption here that the expressions developed throughout this section provide a good estimation of the average numerical errors observed for a given velocity distribution in space.

As a first approximation, we consider here a uniform distribution $\sigma(\beta) = 1$ in the interval $\beta \in [0, 1]$, as both functions are periodic with respect to β. Due to the oddness, $\overline{\phi^r - \phi} = 0$, which means that the dispersion error is zero

on average. In order to distinguish several redistribution schemes, we rather consider the following average, based on the modified wavenumber

$$\overline{k^*} \triangleq \frac{1}{\bar{\beta}} \int_0^{\beta_{\max}} \sigma k^* \, d\beta' \, ,$$

using again $\beta_{\max} = 1$[1].

Different redistribution schemes are compared in Fig. 8.9 and Fig. 8.10. The definitions of the redistribution schemes are given in Appendix C. Only continuous interpolating ($w(p) = \delta[p]$ with $p \in \mathbb{Z}$)) schemes are considered here (i.e. M_2, Λ_3, M_4' and M_6^{**}). Non-interpolating schemes ($w(0) \neq 1$, e.g. M_3 and M_4) lead to higher dissipation errors, in particular for an integer shift β, where interpolating schemes do not induce any error. All discontinuous schemes (e.g. Λ_2) are intrinsically asymmetric (for conservation reasons) and they lead to higher dispersion errors. The modified wavenumbers obtained when using centered finite differences are also shown in Fig. 8.10, when considering the dispersion properties (they have no dissipation error).

The M_2 kernel is the most dissipative scheme, whereas M_6^{**} is the less dissipative one. In terms of dispersion, M_4' and M_2 appear to be identical; they are a little worse than Λ_3 and M_6^{**}. Yet, all four schemes are very similar as to their dispersion errors. The comparison with the finite differences is particularly appealing. The average should be considered with some care, though. Indeed, as previously observed, the underlying function k^* is discontinuous with respect to β when $kh = \pi$ which induces high dispersion errors, and the average behavior is thus too optimistic in that case (equivalently stated, the standard deviation of k^* is high). The best scheme is M_6^{**}, but its interpolation range affects 6 points, as opposed to M_4' which affects 4 grid points. The latter still provides satisfactory dissipation and dispersion properties at a sensibly smaller computational cost.

[1] One may argue that the averaging interval should be taken larger here, since $\beta_{\max} >> 1$ in practice for a Lagrangian method ($\beta_{\max} = n^r \mathrm{CFL}_{\max}$, where $\mathrm{CFL}_{\max}$ and n^r may both be larger than 1). However, the choice $\beta_{\max} = 1$ allows a better comparison between the different schemes, while it is clear that the average dispersion error k^* tends to 0 when $\beta_{\max} \to \infty$, following previous observations.

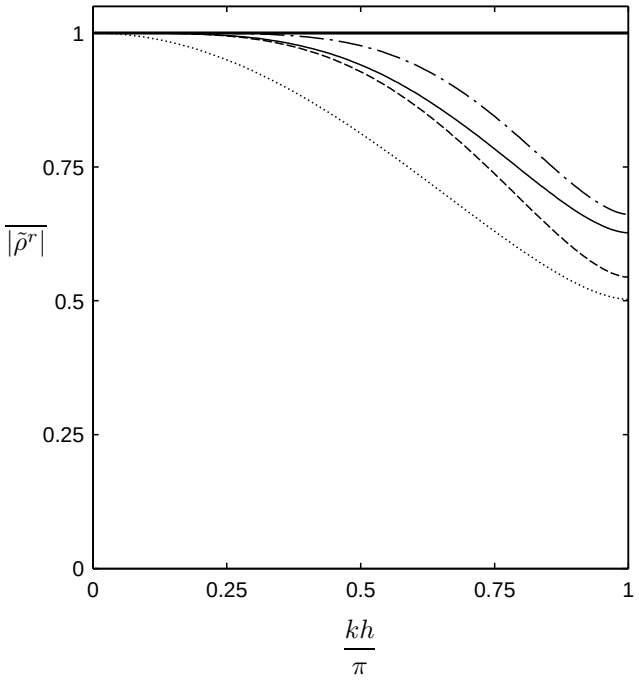


Figure 8.9: *Average numerical dissipation error $|\tilde{\rho}|$ for different redistribution schemes ($\beta_{max} = 1$): M_2 (dotted line), Λ_3 (dashed line), M_4' (thin solid line) and M_6^{**} (dash-dotted line).*

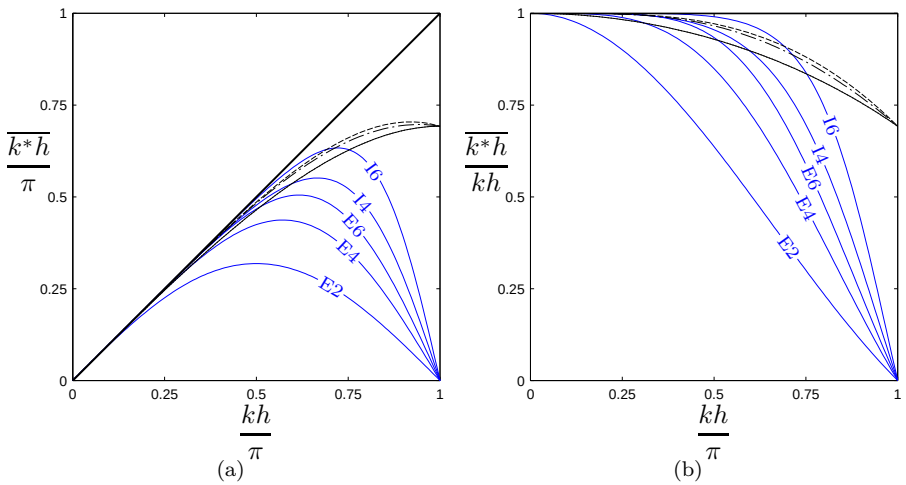


Figure 8.10: *Average numerical dispersion error k^* for different redistribution schemes ($\beta_{max} = 1$): M_2 (dotted line), Λ_3 (dashed line), M_4' (thin solid line) and M_6^{**} (dash-dotted line); exact wavenumber k (thick solid line); dispersion errors for centered finite differences (blue lines; "Ep" and "Ip" stand respectively for a p^{th} order explicit and implicit scheme).*

8.2.2 Recursive application of the redistribution

Instead of applying the redistribution scheme only once after the particles have traveled β cells, we also consider the case where the scheme is applied n times after each travel of β/n cells. This amounts to study the effect of increasing the redistribution frequency, for a given total displacement of β cells. In Fourier space, the recursively applied convolution becomes

$$\tilde{U}^{n,r}[l] = \left(\tilde{\rho}^r_{\beta/n}[l]\right)^n \; \tilde{U}^0[l] \; ,$$

where the new notation $\tilde{\rho}^r_{\beta/n}$ is adopted in order to point out that the amplification factor is now computed with a shift β/n in Eq. (8.6). After the n^{th} redistribution, an ideal scheme would perfectly recover the displacement of β cells (and without dissipation), for all modes. This is not the case here and, moreover, we have that $(\tilde{\rho}^r_{\beta/n})^n \neq \tilde{\rho}^r_{\beta}$, in general. The phase is now computed as

$$\phi^r_{n\cdot\beta/n} \triangleq \arg\left((\tilde{\rho}^r_{\beta/n})^n\right) = n \; \mathrm{Arg}(\tilde{\gamma}^r_{\beta/n}) - kh \; \beta \; ,$$

with $\tilde{\rho}^r_{\beta/n} \triangleq \tilde{\rho}_{\beta/n} \; \tilde{\gamma}^r_{\beta/n}$ and $\tilde{\rho}_{\beta/n}$ the exact amplification factor for a shift of β/n cells.

The errors resulting from a recursive application of the scheme are shown in Fig. 8.11 when $\beta = 1$, and in Fig. 8.12 when $\beta = 3.7$. The associated phase error is not shown here, as the information is redundant with k^*h in this case (it only differs from it by the factor $-\beta$). Considering the case $\beta = 1$, one may observe that no dissipation and no dispersion occur for $n = 1$, by definition, whereas $n = 2$ leads to a maximal dissipation error. This is in agreement with previous results, as the redistribution is applied twice with a shift of 0.5 cells. For $n > 2$, the recursive redistribution appears to converge to a non-dissipative, yet dispersive scheme. This is due to the fact that the intermediate shifts β/n tend to 0 when $n \to \infty$. Formally, we can write the following limit for the effective diffusion coefficient $\epsilon^*_{n\cdot\beta/n}$, associated to n applications of the redistribution scheme with a shift β/n:

$$\begin{aligned}\lim_{n\to\infty}\left(\frac{\epsilon^*_{n\cdot\beta/n}}{ch}\right) &\triangleq \lim_{n\to\infty}\left(\frac{\log|\tilde{\rho}^r_{\beta/n}|^n}{\beta(kh)^2}\right)\\ &= \lim_{n\to\infty}\left(\frac{\log|\tilde{\rho}^r_{\beta/n}|}{\frac{\beta}{n}(kh)^2}\right)\\ &= \lim_{\beta\to 0}\left(\frac{\epsilon^*_{1\cdot\beta}}{ch}\right)\ ,\end{aligned}$$

where $\epsilon^*_{1\cdot\beta}$ corresponds to a unique application of the scheme with a shift β. Referring to Fig. 8.3, we may state that $\lim_{\beta\to 0}\,(\epsilon^*_{1\cdot\beta}) = 0$ and as a consequence, we have that

$$\lim_{n\to\infty}|\tilde{\rho}^r_{\beta/n}|^n = 1\ ,$$

which is in agreement with Fig. 8.11 (a).

Similarly, the modified wavenumber $k^*_{n\cdot\beta/n}$, associated to n applications of the redistribution scheme with a shift β/n, also has a finite limit

$$\begin{aligned}\lim_{n\to\infty}(k^*_{n\cdot\beta/n}\ h) &\triangleq \lim_{n\to\infty}\left(kh - \frac{n}{\beta}\ \mathrm{Arg}(\tilde{\gamma}^r_{\beta/n})\right)\\ &= \lim_{\beta\to 0}\,(k^*_{1\cdot\beta}\ h)\ ,\end{aligned}$$

which is the modified wavenumber obtained when applying the redistribution only once for $\beta \to 0$. It may be computed using L'Hospital's rule as

$$\begin{aligned}\lim_{\beta\to 0}\,(k^*_{1\cdot\beta}\ h) &= -\lim_{\beta\to 0}\ \frac{\partial\phi^r}{\partial\beta}\\ &= \frac{\sum_{m=-\infty}^{+\infty}(kh-2\pi m)\ \hat{W}\,(kh-2\pi m)}{\sum_{m=-\infty}^{+\infty}\hat{W}\,(kh-2\pi m)}\\ &= \sum_{m=-\infty}^{+\infty}(kh-2\pi m)\ \hat{W}\,(kh-2\pi m)\ ,\end{aligned}$$

if the kernel $w(\xi)$ is interpolating, as then $\sum_{m=-\infty}^{+\infty}\hat{W}\,(kh-2\pi m) = 1$. Indeed, this corresponds to a periodization of the spectrum $\hat{W}(kh)$ and hence to a sampling of $w(\xi)$. Sampling $w(\xi)$ yields a Dirac function whose transform is equal to 1. As a side note, this limit seems to be

$$\lim_{\beta\to 0}\,(k^*_{1\cdot\beta}\ h) = \sin(kh)\ ,$$

for the M'_4 scheme, as can be seen in Fig. 8.6, Fig. 8.11 and Fig. 8.12. This corresponds to the modified wavenumber of a second order centered finite dif-

ference scheme, as shown in Fig. 8.10. The limit of the amplification factor phase is then

$$\lim_{n\to\infty} \phi^r_{n\cdot\beta/n} = -\beta \lim_{\beta\to 0} \left(k^*_{1\cdot\beta}\, h\right) .$$

Remarkably, and contrary to the dissipation error, where $\lim_{n\to\infty} |\tilde{\rho}^r_{\beta/n}|^n = \lim_{\beta\to 0} |\tilde{\rho}^r_\beta|$, one observes that

$$\lim_{n\to\infty} \phi^r_{n\cdot\beta/n} \neq \lim_{\beta\to 0} \phi^r_{1\cdot\beta} = 0 ,$$

the latter equality resulting from Fig. 8.5.

The asymptotic limits that were derived for the dissipation and the dispersion errors do not assume any particular value of β. Moreover, the limits for ϵ^* and k^* no longer depend on β. Indeed, this is what can be observed for both cases $\beta = 1$ and $\beta = 3.7$ (the choice of the latter value is such that it is larger than 1 and "arbitrary"). Nevertheless, the convergence is clearly non monotonic when $\beta/n \geq 0.5$, as shown for example in Fig. 8.12 when $n \leq 2\beta = 7.4$. This results from the fact that the behavior of $|\tilde{\rho}^r|$ and k^* is non-monotonic with respect to β when $n = 1$ and $\beta \geq 0.5$ (see Fig. 8.2 and Fig. 8.6).

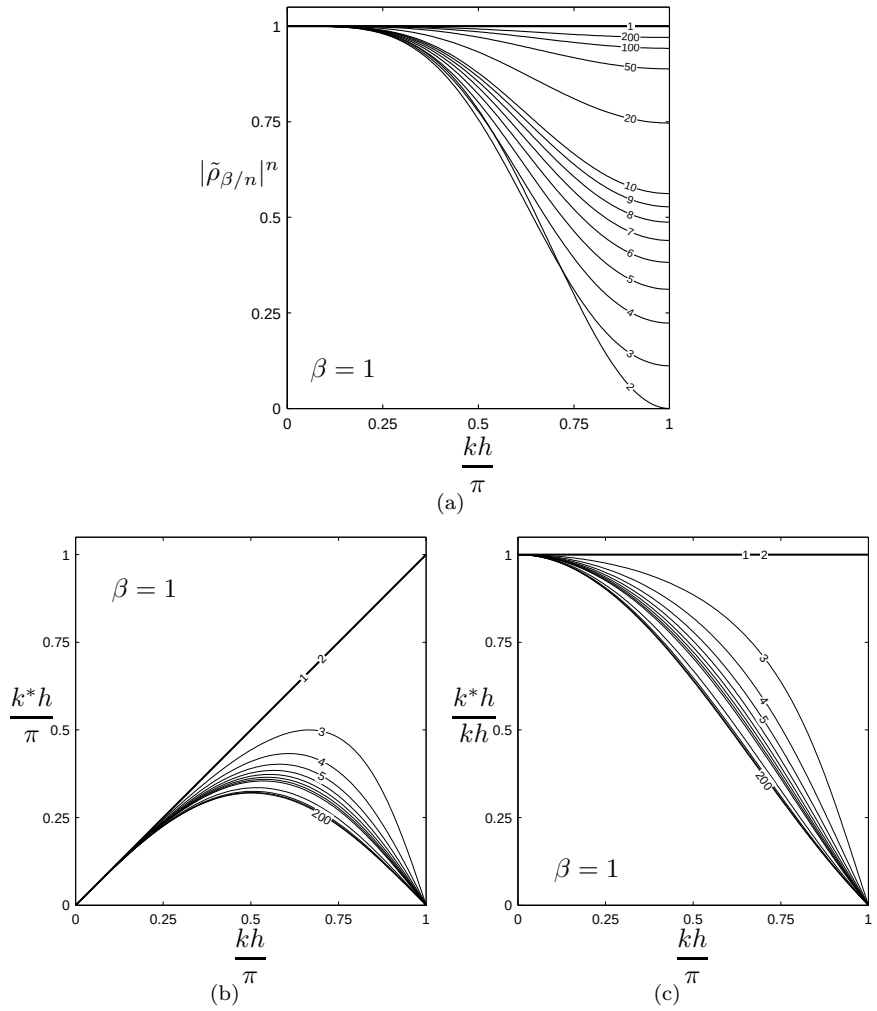


Figure 8.11: *Numerical dissipation error $|\tilde{\rho}|$ (a) and dispersion error k^* (b and c) obtained for M_4'; the scheme is successively applied n times with a shift β/n each time, for a total displacement of $\beta = 1$ (the curves are labeled with the associated value of n).*

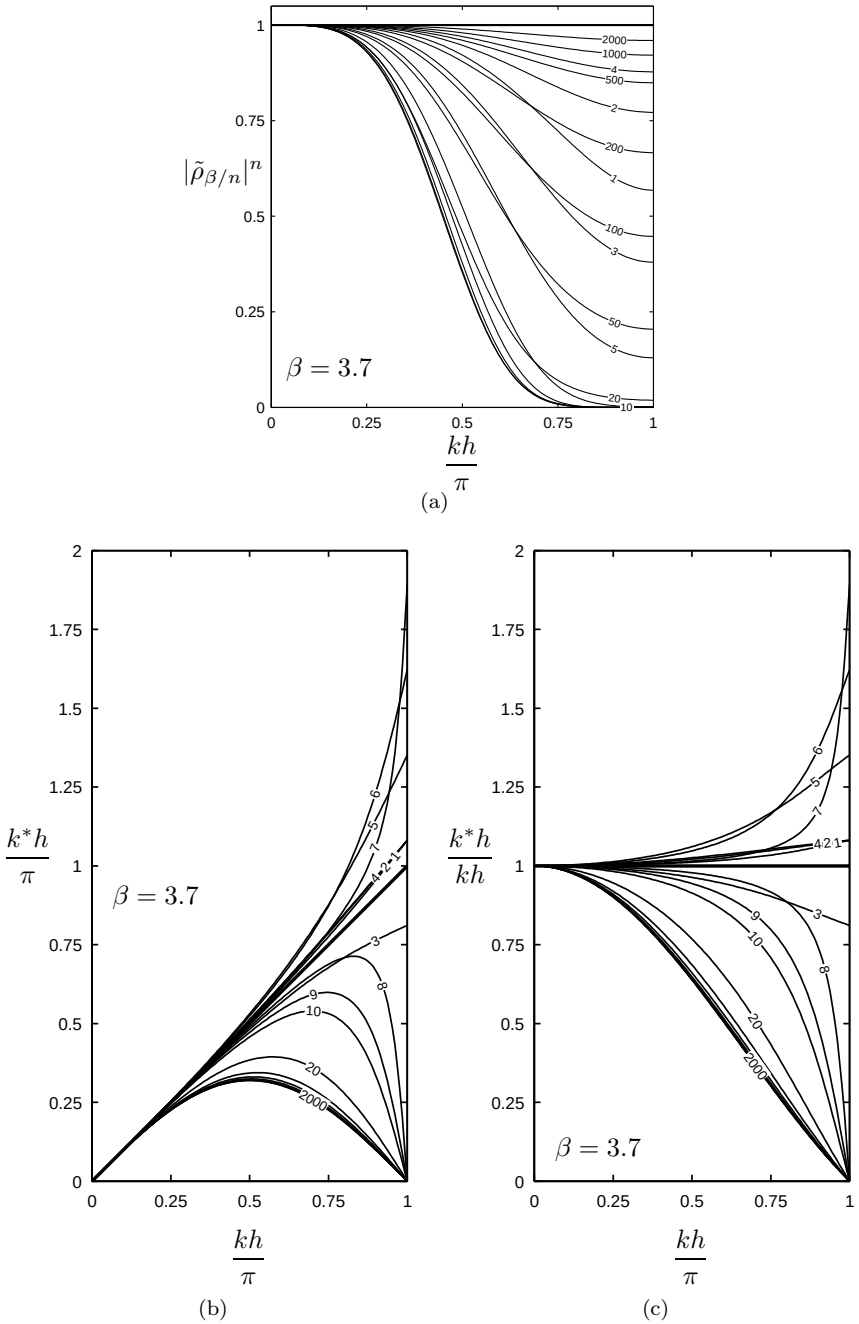


Figure 8.12: *Numerical dissipation error $|\tilde{\rho}|$ (a) and dispersion error k^* (b and c) obtained for M_4'; the scheme is successively applied n times with a shift β/n each time, for a total displacement of $\beta = 3.7$ (the curves are labeled with the associated value of n).*

8.2.3 Conclusion

Lagrangian methods have negligible numerical dispersion and dissipation errors, by definition. However, in practice, periodic redistribution of the particles is required in order to maintain an accurate representation of the described field (i.e., the vorticity), and the Lagrangian method hence implicitly adopts a Eulerian character, while revealing some typically grid-related properties, such as numerical dispersion and dissipation errors.

Those errors are closely related to the redistribution frequency, when considering the modified wave number and the effective numerical diffusion coefficient. As expected, the lower the redistribution frequency, the lower the numerical errors. As a consequence, the optimal redistribution frequency results from a balance between the latter numerical errors and the error induced by flow distortion when the frequency is low (i.e. when the redistribution is applied "too rarely"). Studying the effect of the particle distortion would require considering a non-uniform advection velocity (and perhaps a 2-D framework), contrary to the present analysis that was carried out for a uniform velocity.

Nevertheless, one of the advantages of Lagrangian methods is the absence of formal CFL-type condition, which allows using larger time steps. Hence, even if the redistribution occurs every time step, the average number of cells traveled by a particle before being redistributed is generally significantly higher than for a conventional grid-based method, in practice, and the associated numerical errors remain consequently lower. Comparisons of a remeshed vortex particle-mesh method with a pseudo-spectral method indeed show a very good agreement in terms of energy spectra and dissipation when simulating medium [37] and high Reynolds number flows [127], for both resolved and under resolved grid configurations.

Chapter 9

Conclusions

This thesis makes a contribution to wall-bounded VPM methods by proposing a novel approach for the integration of solid boundaries, based on an immersed interface technique initially introduced in [77]. Thanks to their high level of maturity, vortex methods already excel and outperform other CFD techniques for the simulation of *unbounded* vortical flows (study of the wake interactions in wind farms or the characterization of aircraft wakes in various atmospheric conditions). Yet, considering *wall-bounded* flows, some challenges still need to be addressed and further developments are required in order to make them competitive for a broader range of applications. While several well-performing techniques have been proposed in the literature, there is room for improvement, as to their application within vortex particle-mesh (VPM) methods. The main topic of this work has been to highlight some of the weaknesses of currently available VPM methods concerning the treatment of solid walls and to provide appropriate solutions so as to improve them.

9.1 Achievements

The motivation for the development of an immersed interface VPM method arose from investigations about the behavior of a 3-D hybrid Eulerian-Lagrangian solver.

Indeed, the starting point of this thesis was the development of such a hybrid method and it was driven by the consideration that the combination of a VPM method with a near-wall Eulerian solver is the most natural evolution for vortex methods aiming at the simulation of high Reynolds number wall-

bounded flows. The present hybrid approach, based on an overlapping domain technique, has been inspired from Daeninck [40] and combines a finite volume solver from the OpenFOAM software library with the VPM method that was developed in Lonfils [83].

The core developments of this thesis were initiated from the observation of spurious high-frequency oscillations in the drag evolution during the validation of the methodology on the flow past a sphere. Some indications seemed to show that this spurious phenomenon might be related to the treatment of the no-slip condition by the VPM approach, yet the causality has not been definitely established here. A consistency problem inside the VPM solver has been highlighted, occurring between the vortex panel method approach and the grid-based Poisson solver.

From this point of view, adopting an immersed interface approach is the appropriate response to the above consistency problem, as it allows performing all spatial differential operations on the grid, thus conforming to the basic philosophy of vortex particle-mesh methods, by using similar finite difference stencils for the wall treatment. Consequently, the various operations of the VPM method have been revisited with the target of incorporating these techniques at all computational levels, so as to maintain the accuracy up to the wall.

Regarding the development of immersed interface techniques for VPM methods, the main achievements of this work are:

- The present numerical framework involves only purely one-dimensional immersed interface corrections for the spatial derivatives, as they are computed at the intersection of the interface with the grid lines (i.e. at the control points). Moreover, there is no interdependence between the control points. This allows performing the correcting procedure one grid direction at a time, without considering the neighbors.

- A 2-D immersed interface Poisson solver has been developed with the aim of providing a consistent tool for the computation of a velocity field satisfying a no-through flow condition at the wall and a far-field condition enforcing a free stream flow. The approach is based on the James-Lackner algorithm for the enforcement of the far-field condition. Furthermore, in the context of the flow computation past an airfoil, a new scheme for the enforcement of the Kutta-Joukowsky has been introduced.

- In the context of the diffusion term correction, a "compatible extrapolation scheme" has been developed in order to account for Neumann boundary conditions, without significantly affecting the stability compared to the classical (uncorrected) scheme. The challenge here consists in reconciling the one-dimensional nature of the corrections with the intrinsically 2-D flux condition.

- The interpolation procedure between the particles and the grid has been adapted so as to account for the wall, while still relying on the classical techniques used by vortex methods (using high order interpolation kernels). The approach is based on the pre-computation of ghosts, which are grid values for the M2P procedure (mesh to particles) and particle values for the P2M interpolation (particles to mesh). The ghost particles are obtained by a level set extension technique.

- Finally, the previous tools have been successfully combined into an immersed interface-enabled vortex particle-mesh solver. The approach has been validated on some challenging test cases, such as the impulsively started flow past a cylinder or past an airfoil with a sharp trailing edge and the vortex shedding induced by the flow past a cylinder.

The presently developed tool takes advantage from both simulation approaches, the Lagrangian and the Eulerian methodologies. The spatial differential operators are performed in a more consistent and accurate manner near the wall, while keeping the advantage of working with particles for the advection. By doing so, the VPM method surely comes closer to grid-based methods and perhaps loses a bit from its "mesh-less" nature. Yet, the method is able to perform very well on test cases which are usually considered only achievable by purely Lagrangian methods, i.e. the simulation of impulsively started flows. In particular, the simulation results for the near-trailing edge flow dynamics of an impulsively started airfoil (formation of the starting vortex) are quite remarkable.

The eventual target for the simulation of wall-bounded flows still consists in developing a hybrid Eulerian-Lagrangian VPM solver, which is commonly accepted the most appropriate way to achieve the simulation of high Reynolds number flows using the VPM approach. Improving the treatment of the wall inside the VPM approach is one step towards that direction, considering hybrid approaches based on overlapping domains. Moreover, in its stand-alone version

(thus non-hybrid), the immersed interface VPM solver provides a more accurate tool for the simulation of moderate Reynolds number flows.

In addition to these achievements, some complementary studies have been performed, so as to provide more insight into two specific aspects of VPM methods, namely the temporal accuracy of the no-slip enforcing procedure and the quantification of the numerical errors of dispersion and dissipation introduced by the redistribution procedure. The conclusions of these investigations are as follows:

- The academical test case consisting of a dipole flow inside a cavity has been used as a "sand box" problem in order to study the no-slip enforcing procedure, with the aim of attempting to improve the associated temporal accuracy compared to the classical approach. The very simple problem geometry has allowed bypassing the errors due to arbitrary intersections between the wall and the grid, as introduced by immersed interface methods. Improving the time accuracy of the no-slip enforcing procedure is twofold. First, the effect of the splitting of the equations has been studied and an inconsistency of the classical methodology has been highlighted and remedied for the case where the vorticity flux is prescribed. Secondly, the vorticity flux approximation intended to yield the no-slip condition has been analyzed by studying the characteristics of the associated vortex sheet. Yet, the global accuracy of the approach could not be formally improved and some suggestions have been made for future research.

- As a conclusion of this work, we have studied the numerical errors of dispersion and dissipation that are introduced by the redistribution of the particles, in the 1-D case. Results have shown that the approach is less prone to those errors, when the redistribution frequency is reduced (provided the flow has not distorted the set of particles too much, in the 2-D and 3-D case), as one could expect.

9.2 Perspectives

The present results have shown that the immersed interface approach, which shares many aspects with grid-based methods, is able to compete with classical vortex methods. It thus offers a more consistent framework for vortex methods

in the context of wall-bounded flows. Yet, the present method remains a proof of concept and several improvements can be envisaged:

- The first and most obvious point consists in a generalization to 3-D flows. The use of one-dimensional stencil corrections is intrinsically prone to be generalized to multiple dimensions. Yet, accounting for the no-through flow condition using the streamfunction is less straightforward in 3-D and a velocity-based approach could be used. According to the suggestions made in Chapter 4, a Neumann boundary condition must then be enforced on the wall. This could be achieved by using the "compatible extrapolation scheme" from Chapter 5.

- At the present time, the computation of the Poisson solution is still quite expensive in terms of computational time (see Section 7.4), despite the limited number of iterations for the enforcement of the far-field condition. Solving the linear system $Ax = b$ with the wall corrections could be performed by relying on the decomposition of the matrix $A = A_0 + A_w$ into a contribution A_0 corresponding to a classical Poisson solver and another one A_w corresponding to the wall corrections. One may then solve the problem either iteratively by relaxation [80], either by using the Sherman-Morrison-Woodbury formula, as was done in [20] for a penalization method. Both approaches allow using a fast Poisson solver for the inversion of the matrix A_0, which should significantly accelerate the computation. Regarding the iterative approach, one could imagine merging the outer and inner iterations of the algorithm (iterations for the far-field condition and for the wall corrections) and hence performing a single iteration loop.

- The methodology that has been developed here is compatible with hierarchically refined grids. Such an approach has already been implemented in [83] in the framework of a vortex panel-based VPM approach. As mentioned therein, one could also envisage building the solver upon existing multi resolution libraries, as for example *Chombo* [89] or *Overture* [13].

- The immersed interface framework is also well-suited for applications in fluid-structure interaction and multi phase flows. However, accounting for moving and deforming bodies or interfaces requires keeping track of the interface and hence using an efficient control point computation/update, as the latter evolve in time and need to be recomputed at every time step.

For applications in multi phase flows, the underlying level set framework allows accounting for the surface tension, by means of the local surface curvature computation. As to fluid-structure problems, the control points could be considered as attached to their equilibrium position by virtual "springs" of prescribed stiffness (depending on the structural properties of the body), in the sense of the original immersed boundary method of Peskin [101]. A more advanced treatment consists in coupling the flow solver with a structural code, as classically done in the fluid-structure interaction literature. The advantage of the present immersed interface approach is that any type of boundary condition can be enforced on both sides of the interface.

- As an improved time accuracy for the no-slip enforcing procedure has not been achieved yet, some developments in this direction would allow consistently increasing the order of convergence for the time discretization, as it has been done for the spatial discretization.

- Finally, coming back to the initial development of this work and in the light of the lessons learned thereafter, it appears that the hybrid scheme could benefit from some of the interesting features of the immersed interface approach. Indeed, it is more consistent than panel-based methods and less oscillatory in time, when considering the drag for impulsively started flows. Nevertheless, the immersed interface VPM solver turns out to be less robust for under resolved cases in its current form. This is actually problematic, since the coupling approach becomes interesting, from a computational point of view, precisely when the VPM solver is under resolved near the wall. It is thus not clear yet to what extent the present method would enhance the Lagrangian part of the hybrid solver. Yet, the tools developed in this work could be considered in order to improve some of the computational operations from existing VPM solvers, e.g. the particle-mesh interpolation, etc.

  Furthermore, a different domain decomposition could be envisaged, by using a VPM domain that does not include the near-wall region but rather begins at a certain distance from it. The boundary condition on that surface delimiting the VPM domain from the near-wall region could then be enforced using the present immersed interface tools. However, maybe this requires using a Schwarz iteration for the computation of the boundary condition.

One could also consider another possibility for trying to remove the spurious oscillations from the drag signal (as observed in Chapter 2), while still using overlapping domains and the VPM solver in its original form [83] (thus using vortex panels). As shown in Fig. 9.1, a buffer layer could be introduced (in which the solutions from the Eulerian and the Lagrangian subdomains would agree well), as well as a blending zone near the wall. Inside the blending zone, in the direct vicinity of the wall, the VPM solver would not solve the Navier-Stokes equations (the proper vorticity is here provided by the Eulerian near-wall solver and as soon as it is interpolated to the VPM subdomain, it would be simply "frozen" for the next sub step). As one approaches the outer boundary of this zone, the VPM solver would be fully switched on and the vorticity would thus be properly updated. The only purpose of the VPM vorticity inside this blending region is its contribution to the computation of the velocity, by means of the Poisson solution. The errors usually made by the VPM solver near the wall due to its under resolved nature (and which are believed to contribute to the spurious oscillations) could thus be prevented.

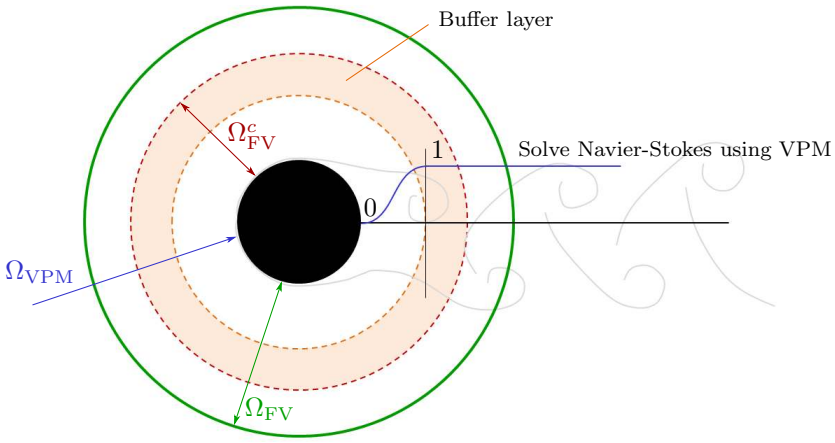


Figure 9.1: *Sketch of an alternative coupling approach for the hybrid Eulerian-Lagrangian solver using overlapping domains, a buffer layer and a near-wall blending region where the Navier-Stokes equations are only partially solved by the VPM solver, depending on the distance to the wall.*

Bibliography

[1] OpenFOAM. `http://www.openfoam.org`.

[2] D. Adalsteinsson and J. A. Sethian. The Fast Construction of Extension Velocities in Level Set Methods. *J. Comput. Phys.*, 148(1):2–22, January 1999.

[3] P. Angot, C.-H. Bruneau, and P. Fabrie. A penalization method to take into account obstacles in incompressible viscous flows. *Numer. Math.*, 81(4):497–520, 1999.

[4] S. Backaert, P. Chatelain, and G. Winckelmans. Vortex particle-mesh with immersed lifting lines for aerospace and wind engineering. In *Symposium on Particle Methods in Fluid Mechanics, Oct. 15-17*, DTU Copenhagen, 2012. Procedia IUTAM.

[5] M. Bar-Lev and H. Yang. Initial flow field over an impulsively started circular cylinder. *J. Fluid Mech.*, 72(04):625–647, 1975.

[6] J. Barnes and P. Hut. A hierarchical O(N log N) force calculations algorithm. *Nature*, 324:446–449, 1986.

[7] J. T. Beale and A. Majda. Rates of convergence for viscous splitting of the Navier–Stokes equations. *Math. Comput.*, 37(156):243–259, October 1981.

[8] M. Bergdorf. *Multiresolution Particle Methods for the Simulation of Growth and Flow.* PhD thesis, ETH Zuerich, 2007.

[9] M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. Multilevel adaptive particle methods for convection-diffusion equations. *Multiscale Model. Sim.*, 4(1):328–357, 2005.

[10] M. Bergdorf and P. Koumoutsakos. A Lagrangian particle-wavelet method. *Multiscale Model. Sim.*, 5(3):980–995, 2006.

[11] I. Borazjani and F. Sotiropoulos. On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming. *J. Exp. Biol.*, 213(1):89–107, 2010.

[12] I. Borazjani and F. Sotiropoulos. Numerical investigation of the hydrodynamics of anguilliform swimming in the transitional and inertial flow regimes. *J. Exp. Biol.*, 212(4):576–592, 2009.

[13] D. Brown, W. Henshaw, and D. Quinlan. Overture: An object-oriented framework for solving partial differential equations. In Y. Ishikawa, R. Oldehoeft, J. Reynders, and M. Tholburn, editors, *Scientific Computing in Object-Oriented Parallel Environments*, volume 1343 of *Lecture Notes in Computer Science*, pages 177–184. Springer Berlin / Heidelberg, 1997.

[14] D. Calhoun. A Cartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions. *J. Comput. Phys*, 176(2):231–275, 2002.

[15] C. Casciola, R. Piva, and P. Bassanini. Vorticity Generation on a Flat Surface in 3D Flows. *J. Comput. Phys.*, 129(2):345 – 356, 1996.

[16] P. Chatelain, A. Curioni, M. Bergdorf, D. Rossinelli, W. Andreoni, and P. Koumoutsakos. Billion vortex particle direct numerical simulations of aircraft wakes. *Comput. Method. Appl. M.*, 197:1296 – 1304, 2008.

[17] P. Chatelain and P. Koumoutsakos. A Fourier-based elliptic solver for vortical flows with periodic and unbounded directions. *J. Comput. Phys.*, 229(7):2425 – 2431, 2010.

[18] P. Chatelain and A. Leonard. Isotropic compact interpolation schemes for particle methods. *J. Comput. Phys.*, 227(6):3244 – 3259, 2008.

[19] R. Chatelin. *Numerical methods for 3D Stokes flow: variable viscosity fluids in a complex moving geometry; application to biological fluids*. PhD thesis, Universite Paul Sabatier - Toulouse III, 2013.

[20] R. Chatelin and P. Poncet. Hybrid grid-particle methods and Penalization: A Sherman-Morrison-Woodbury approach to compute 3D viscous flows using {FFT}. *J. Comput. Phys.*, 269(0):314 – 328, 2014.

[21] I.-L. Chern and Y.-C. Shu. A coupling interface method for elliptic interface problems. *J. Comput. Phys.*, 225(2):2138 – 2174, 2007.

[22] A. J. Chorin. Numerical study of slightly viscous flow. *J. Fluid Mech.*, 57:785–796, 3 1973.

[23] J. P. Christiansen. Numerical solution of hydrodynamics by the method of point vortices. *J. Comput. Phys.*, 13:363–379, 1973.

[24] D. K. Clarke, H. A. Hassan, and M. D. Salas. Euler calculations for multielement airfoils using Cartesian grids. *AIAA Journal*, 24(3):353–358, March 1986.

[25] H. J. H. Clercx and C.-H. Bruneau. The normal and oblique collision of a dipole with a no-slip boundary. *Comput. Fluids*, 35(3):245 – 279, 2006.

[26] H. J. H. Clercx and G. J. F. van Heijst. Dissipation of kinetic energy in two-dimensional bounded flows. *Phys. Rev. E*, 65:066305, Jun 2002.

[27] R. Cocle. *Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations : DNS and LES approaches*. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, 2007.

[28] R. Cocle, G. Winckelmans, and Daeninck. Combining the vortex-in-cell and parallel fast multipole methods for efficient domain decomposition simulations. *J. Comput. Phys.*, 227:2263–2292, 2008.

[29] W. Collins and S. Dennis. Flow past an impulsively started circular cylinder. *J. Fluid Mech.*, 60(01):105–127, 1973.

[30] W. Collins and S. Dennis. The initial flow past an impulsively started circular cylinder. *Q. J. Mech. Appl. Math.*, 26(1):53–75, 1973.

[31] G. Constantinescu and K. Squires. Numerical investigations of flow over a sphere in the subcritical and supercritical regimes. *Phys. Fluids*, 16(5):1449–1466, 2004.

[32] M. Coquerelle and G.-H. Cottet. A vortex level set method for the two-way coupling of an incompressible fluid with colliding rigid bodies. *J. Comput. Phys.*, 227(21):9121 – 9137, 2008.

[33] G.-H. Cottet. A vorticity creation algorithm for the Navier-Stokes equations in arbitrary domain. Technical report.

[34] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice.* Cambridge University Press, first edition, 2000.

[35] 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.

[36] G.-H. Cottet and A. Magni. TVD remeshing formulas for particle methods. *CR. Math.*, 347(23):1367 – 1372, 2009.

[37] G.-H. Cottet, B. Michaux, S. Ossia, and G. VanderLinden. A Comparison of Spectral and Vortex Methods in Three-Dimensional Incompressible Flows. *J. Comput. Phys.*, 175(2):702 – 712, 2002.

[38] G.-H. Cottet and P. Poncet. Simulation and control of three-dimensional wakes. *Comput. Fluids*, 33(5-6):697 – 713, 2004. Applied Mathematics for Industrial Flow Problems.

[39] G.-H. Cottet and L. Weynans. Particle methods revisited: a class of high order finite-difference methods. *CR. Math.*, 343(1):51 – 56, 2006.

[40] G. Daeninck. *Developments in hybrid approaches. Vortex method with kwnown separation location. Vortex method with near-wall Eulerian solver. RANS-LES coupling.* PhD thesis, Université catholique de Louvain, 2006.

[41] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. I. The case of an isotropic viscosity; II. The anisotropic case. *Math. Comput.*, 53(188):485–507, 1989.

[42] C. Doolan. Flow and Noise Simulation of the NASA Tandem Cylinder Experiment using OpenFOAM. *15th AIAA/CEAS Aeroacoustics Conference (30th AIAA Aeroacoustics Conference)*, AIAA 2009-3157, 2009.

[43] U. D'Ortona, G. Keetels, D. Kolomenskiy, K. Schneider, E. Leriche, H. Clercx, and B. Ben Beya. A comparison of different numerical methods to study the normal collision of a dipole with a non-slip boundary. *To be published, private communication.*

[44] B. Engquist, A.-K. Tornberg, and R. Tsai. Discretization of Dirac delta functions in level set methods. *J. Comput. Phys.*, 207(1):28 – 51, 2005.

[45] R. D. Falgout, J. E. Jones, and U. M. Yang. Pursuing scalability for hypre's conceptual interfaces. *ACM T. Math. Software*, 31(3):326–350, September 2005.

[46] R. D. Falgout and U. M. Yang. hypre: a Library of High Performance Preconditioners. In *Preconditioners, Lecture Notes in Computer Science*, pages 632–641, 2002.

[47] P. Farrell and J. Maddison. Conservative interpolation between volume meshes by local Galerkin projection. *Comput. Method. Appl. M.*, 200(1):89 – 100, 2011.

[48] P. Farrell, M. Piggott, C. Pain, G. Gorman, and C. Wilson. Conservative interpolation between unstructured meshes via supermesh construction. *Comput. Method. Appl. M.*, 198(33):2632 – 2642, 2009.

[49] M. Frigo and S. G. Johnson. The Design and Implementation of FFTW3. *P. IEEE*, 93(2):216–231, 2005. Special issue on "Program Generation, Optimization, and Platform Adaptation".

[50] M. Gazzola, P. Chatelain, W. M. van Rees, and P. Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *J. Comput. Phys.*, 230(19):7093 – 7114, 2011.

[51] L. Ge and F. Sotiropoulos. A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries. *J. Comput. Phys.*, 225(2):1782 – 1809, 2007.

[52] L. Georges. *Development and validation of a LES methodology for complex wall-bounded flows : application to high-order structured and industrial unstructured solvers.* PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, 2007.

[53] L. Georges, G. Winckelmans, and P. Geuzaine. Improving shock-free compressible RANS solvers for LES on unstructured meshes. *J. Comput. Appl. Math.*, 215(2):419–428, 2008.

[54] A. Gilmanov and F. Sotiropoulos. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *J. Comput. Phys.*, 207(2):457 – 492, 2005.

[55] L. Greengard and V. Rohklin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[56] L. F. Greengard. *The rapid evaluation of potential fields in particle systems*. PhD thesis, Yale University, 1987.

[57] S. Haykin and B. Van Veen. *Signals and Systems, 2nd Ed*. Wiley India Pvt. Limited, 2007.

[58] J. L. Hess. Panel Methods in Computational Fluid Dynamics. *Annu. Rev. Fluid Mech.*, 22(1):255–274, 1990.

[59] N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM e-books. SIAM, 2008.

[60] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. McGraw-Hill, New York, 1981.

[61] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.*, 62(1):40 – 65, 1986.

[62] R. James. The solution of Poisson's equation for isolated source distributions. *J. Comput. Phys.*, 25(2):71 – 93, 1977.

[63] L. N. Jenkins, D. Neuhart, C. McGinley, and M. Choudhari, M.M Khorrami. Measurements of unsteady wake interference between tandem cylinders. *AIAA journal*, 2006-3202, 006.

[64] T. A. Johnson and V. C. Patel. Flow past a sphere up to a Reynolds number of 300. *J. Fluid Mech.*, 378(-1):19–70, 1999.

[65] J. Katz and A. Plotkin. *Low Speed Aerodynamics*. Aerospace Series. Cambridge University Press, 2001.

[66] G. H. Keetels, U. D'Ortona, W. Kramer, H. J. H. Clercx, K. Schneider, and G. J. F. van Heijst. Fourier spectral and wavelet solvers for the incompressible Navier-Stokes equations with volume-penalization: Convergence of a dipole-wall collision. *J. Comput. Phys.*, 227(2):919 – 945, 2007.

[67] N. K. R. Kevlahan and O. V. Vasilyev. An adaptive wavelet collocation method for fluid-structure interaction at high Reynolds numbers. *SIAM J. Sci. Comput.*, 26(6):1894–1915, 2005.

[68] M. Khorrami, M. Choudhari, L. N. Jenkins, and C. McGinley. Unsteady Flowfield Around Tandem Cylinders as Prototype for Component Interaction in Airframe Noise. *AIAA Paper*, 2005-2866, 2005.

[69] J. Kim and P. Moin. Application of a fractionnal-step method to imcompressible Navier-Stokes equations. *J. Comput. Phys.*, 59:308–323, 1985.

[70] R. B. Kinney and M. A. Paolino. Flow Transient Near the Leading Edge of a Flat Plate Moving Through a Viscous Fluid. *J. Appl. Mech.*, 41:919–924, 1974.

[71] P. Koumoutsakos. Inviscid Axisymmetrization of an Elliptical Vortex. *J. Comput. Phys.*, 138(2):821 – 857, 1997.

[72] P. Koumoutsakos, A. Leonard, and F. Pépin. Boundary Conditions for Viscous Vortex Methods. *J. Comput. Phys.*, 113(1):52 – 61, 1994.

[73] P. Koumoutsakos and A. Leonard. High-resolution simulations of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296:1–38, 1995.

[74] K. Lackner. Computation of ideal MHD equilibria. *Comput. Phys. Commun.*, 12(1):33 – 44, 1976.

[75] M.-C. Lai, Z. Li, and X. Lin. Fast Solvers for 3D Poisson Equations Involving Interfaces in a Finite or the Infinite Domain. *J. Comput. Appl. Math.*, 191(1):106–125, June 2006.

[76] H. Lamb. *Hydrodynamics*. Cambridge Univ. Press, Cambrdige (UK), 6th edition, 1932.

[77] R. J. Leveque and Z. Li. The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources. *SIAM J. Numer. Anal.*, pages 1019–1044, 1994.

[78] Z. Li and K. Ito. Maximum Principle Preserving Schemes for Interface Problems with Discontinuous Coefficients. *SIAM J. Sci. Comput.*, 23(1):339–361, January 2001.

[79] M. J. Lighthill. Introduction. Boundary Layer Theory. In L. Rosenhead, editor, *Laminar Boundary Layers*, pages 46–113. Oxford University Press, 1963.

[80] M. N. Linnick and H. F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *J. Comput. Phys.*, 204(1):157 – 192, 2005.

[81] D. Lockard, M. Khorrami, M. Choudhari, F. Hutcheson, and T. Brooks. Tandem Cylinder Noise Predictions. *Proccedings of the 13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference)*, AIAA 2007-3450, 2007.

[82] D. Lockard. Tandem cylinder benchmark problem. *Technical presentation, computational aeroscience branch NASA LaRC*, 2009.

[83] T. Lonfils. *Numerical investigations of vortex systems: time-developing and space-developing simulations. Improvements of the vortex-in-cell method for external flows using both an immersed boundary approach and a multiple-space resolution technique.* PhD thesis, Université catholique de Louvain, 2010.

[84] A. Magni and G.-H. Cottet. Accurate, non-oscillatory, remeshing schemes for particle methods. *J. Comput. Phys.*, 231(1):152 – 172, 2012.

[85] S. Majumdar, G. Iaccarino, and P. Durbin. RANS solvers with adaptive structured boundary non-conforming grids. *Annual Research Briefs*, pages 353–366, 2001.

[86] Y. Marichal, C. Carton, L. Bricteux, M. Duponcheel, G. Winckelmans, and P. Geuzaine. DDES of the flow past a pair of cylinders in tandem configuration. *Proceedings of the AIAA BANC-I workshop (Stockholm, Sweden)*, June 2010.

[87] Y. Marichal, P. Chatelain, and G. Winckelmans. Unbounded Immersed Interface solver for Vortex Particle-Mesh methods. In *Symposium on Particle Methods in Fluid Mechanics, Oct. 15-17*, DTU Copenhagen, 2012. Procedia IUTAM.

[88] Y. Marichal, P. Chatelain, and G. Winckelmans. An immersed interface solver for the 2-D unbounded Poisson equation and its application to potential flow. *Comput. Fluids*, 96(0):76 – 86, 2014.

[89] D. Martin, P. Colella, and D. Graves. A cell-centered adaptive projection method for the incompressible Navier-Stokes equations in three dimensions. *J. Comput. Phys.*, 227(3):1863–1886, 2008.

[90] P. McCorquodale, P. Colella, G. Balls, and S. Baden. A Local Corrections Algorithm for Solving Poisson's Equation in Three Dimensions. *Communications in Applied Mathematics and Computational Science*, 2(1):57–81, 2007.

[91] G. Miller. An iterative boundary potential method for the infinite domain Poisson problem with interior Dirichlet boundaries. *J. Comput. Phys.*, 227(16):7917 – 7928, 2008.

[92] R. Mittal and G. Iaccarino. Immersed Boundary Methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

[93] J. Monaghan. Extrapolating B splines for interpolation. *J. Comput. Phys.*, 60(2):253 – 262, 1985.

[94] L. Neuhart, a. M. C. Jenkins, and M. Khorrami. Measurements of the Flowfield Interaction Between Tandem Cylinders. In *Aeroacoustics Conferences*, volume AIAA 2009-3275, 2009.

[95] J. N. Newman. *Marine hydrodynamics*. MIT Press, Cambridge, Mass., 1977.

[96] F. Noca, D. Shiels, and D. Jeon. Measuring Instantaneous Fluid Dynamic Forces on Bodies, Using Only Velocity Fields and Their Derivatives. *J. Fluid. Struct.*, 11(3):345 – 350, 1997.

[97] F. Noca, D. Shiels, and D. Jeon. A Comparison of Methods for Evaluating Time -dependent Fluid Dynamic Forces on Bodies, Using Only

VelocityFields and Their Derivatives. *J. Fluid. Struct.*, 13(5):551 – 578, 1999.

[98] M. 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, 2001.

[99] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based Fast Local Level Set Method. *J. Comput. Phys.*, 155(2):410–438, November 1999.

[100] C. S. Peskin. The immersed boundary method. *Acta Numerica*, 11:479–517, 1 2002.

[101] C. Peskin. Numerical analysis of blood flow in the heart. *J. Comput. Phys.*, 25(3):220–252, 1977.

[102] P. Ploumhans and G. S. Winckelmans. Vortex methods for high-resolution simulations of viscous flow past bluff bodies of general geometry. *J. Comput. Phys.*, 165:354–406, 2000.

[103] 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. *J. Comput. Phys.*, 178(2):427–463, 2002.

[104] P. Poncet. Analysis of an immersed boundary method for three-dimensional flows in vorticity formulation. *J. Comput. Phys.*, 228:7268–7288, 2009.

[105] P. Poncet. Topological aspects of three-dimensional wakes behind rotary oscillating cylinders. *J. Fluid Mech.*, 517:27–53, 9 2004.

[106] P. Poncet. Analysis of Direct Three-Dimensional Parabolic Panel Methods. *SIAM J. Numer. Anal.*, 45(6):2259–2297, October 2007.

[107] S. C. Reddy and L. N. Trefethen. Stability of the method of lines. *Numer. Math.*, 62(1):235–267, 1992.

[108] L. Rosenhead. The Formation of Vortices from a Surface of Discontinuity. *P R Soc Lond A-Conta*, 134(823):170–192, 1931.

[109] D. Rossinelli, M. Bergdorf, G.-H. Cottet, and P. Koumoutsakos. GPU accelerated simulations of bluff body flows using vortex particle methods. *J. Comput. Phys.*, 229(9):3316 – 3333, 2010.

[110] P. G. Saffman. *Vortex Dynamics*. Cambridge Univ. Press, New-York, 1992.

[111] J. K. Salmon and M. S. Warren. Skeletons from treecode closet. *SIAM J. Sci. Comput.*, 111(1):136–155, 1994.

[112] I. Schoenberg. Contributions to the Problem of Approximation of Equidistant Data by Analytic Functions. In C. Boor, editor, *I. J. Schoenberg Selected Papers*, Contemporary Mathematicians, pages 3–57. Birkhauser Boston, 1988.

[113] J. A. Sethian. Evolution, Implementation, and Application of Level Set and Fast Marching Methods for Advancing Fronts. *J. Comput. Phys.*, 169(2):503–555, May 2001.

[114] D. Shiels, A. Leonard, and A. Roshko. Flow-induced vibration of a circular cylinder at limiting structural parameters. *J. of Fluid. Struct.*, 15(1):3–21, 2001.

[115] C.-W. Shu. High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems. *SIAM Rev.*, 51(1):82–126, February 2009.

[116] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA-Paper 92-0439*, (1):5–21, 1992.

[117] P. Spalart, S. Deck, M. Shur, K. Squires, M. Strelets, and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theor. Comp. Fluid Dyn.*, 20:181–195, 2006.

[118] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5(3):506–517, 1968.

[119] M. Strelets. Detached Eddy Simulation of Massively Separated Flows. *AIAA 2001-0879*, 2001.

[120] P. Swarztrauber. A direct method for the discrete solution of separable elliptic equations. *SIAM J. Numer. Anal.*, 11:1136–1150, 1974.

[121] P. Swarztrauber, R. A. Sweet, and J. C. Adams. Fishpack: Efficient Fortran Subprograms for the Solution of Elliptic Partial Differential Equations. Technical report, National Center for Atmospheric Research, http://www.cisl.ucar.edu/css/software/fishpack/technote.ps, July 1999.

[122] P. Swarztrauber and R. Sweet. The Fourier and cyclic reduction methods for solving Poisson's equation. In J. Schetz and A. Fuhs, editors, *Handbook of fluid dynamics and fluid machinery*. John Wiley & Sons, New York, 1996.

[123] W. Sweldens and P. Schroeder. Building your own wavelets at home, 1996. Wavelets in Computer Graphics, 15-87. ACM SIGGRAPH Course notes.

[124] F. Thirifay and G. Winckelmans. Development of a Lagrangian method for combustion and application to the planar methane air jet diffusion flame. *J. Turbul.*, 3(59):1468–5248, 2002.

[125] A. Tornberg and B. Engquist. Regularization Techniques for Numerical Approximation of PDEs with Singularities. *J. Sci. Comput.*, 19:527–552, 2002.

[126] Y.-H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys.*, 192(2):593 – 623, 2003.

[127] W. M. van Rees, A. Leonard, D. Pullin, and P. Koumoutsakos. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *J. Comput. Phys.*, 230(8):2794 – 2805, 2011.

[128] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.*, 12,6:620–631, 1998.

[129] A. Wiegmann and K. P. Bube. The Explicit-Jump Immersed Interface Method: Finite Difference Methods For PDE With Piecewise Smooth Solutions. *SIAM J. Numer. Anal.*, 37:827–862, 2000.

[130] J. Williamson. Low-storage Runge-Kutta schemes. *J. Comput. Phys.*, 35(1):48 – 56, 1980.

[131] G. Winckelmans, R. Cocle, L. Dufresne, and R. Capart. Vortex methods and their application to trailing wake vortex simulations. *CR. Phys.*, 6(4-5):467 – 486, 2005. Aircraft trailing vortices.

[132] G. Winckelmans. Vortex methods. In E. Stein, R. de Borst, and T. Hughes, editors, *Encyclopedia of computational mechanics*, volume 3 (Fluids). John Wiley & sons, 2004.

[133] J. Wu and J. Wu. Boundary Vorticity Dynamics Since Lighthill's 1963 Article: Review and Development. *Theor. Comp. Fluid Dyn.*, 10(1-4):459–474, 1998.

Appendix A

One-sided stencils

The computation of the correction terms for the direction-splitting approach requires a systematic way to provide one-sided finite difference stencils. The present appendix provides the required tools, along with an analysis of their accuracy.

Assume we have grid points with positions x_p and x_α is the position of the interface. The relative position of the interface with respect to the the grid points is arbitrary.

Consider a function $u(x) \in C_{N+2}$ and define $\Delta\tilde{x}_p \triangleq (x_p - x_\alpha)/h$ (h being the grid spacing), $u_p \triangleq u(x_p)$ and $u_\alpha^{(k)} \triangleq (d^k u/dx^k)(x_\alpha)$.

Case 1 The most frequently required stencil in the splitting approach consists in expressing the k^{th} derivative of u at x_α as a function of the provided q^{th} derivative of u at x_α and of N grid point values

$$u_\alpha^{(k)} = (S_q^k)_\alpha \; u_\alpha^{(q)} + \sum_{p=1}^{N} (S_q^k)_p \; u_p + \mathcal{O}(h^s) \;\; ,$$

with s the order of the leading error term. The procedure described hereafter generalizes the approach followed in [80], where only u_α may be prescribed, not its derivatives.

The way to handle this problem requires first to form the following linear combination using the Taylor series of the function $u(x)$ evaluated at the grid

nodes and developed around x_α

$$
\begin{aligned}
&C_1\, u_1 + \; C_2\, u_2 + ... + C_N\, u_N \\
&= \quad u_\alpha \quad [\quad C_1 \quad + \quad C_2 \quad + ... + \quad C_N \quad] \\
&+ \quad h\, u_\alpha^{(1)} \quad [\quad C_1\, \Delta\tilde{x}_1 \quad + \quad C_2\, \Delta\tilde{x}_2 \quad + ... + \quad C_N\, \Delta\tilde{x}_N \quad] \\
&+ \quad \frac{h^2}{2}\, u_\alpha^{(2)} \quad [\quad C_1\, \Delta\tilde{x}_1^2 \quad + \quad C_2\, \Delta\tilde{x}_2^2 \quad + ... + \quad C_N\, \Delta\tilde{x}_N^2 \quad] \\
&+ \quad ... \\
&+ \quad \frac{h^N}{N!}\, u_\alpha^{(N)} \quad [\quad C_1\, \Delta\tilde{x}_1^N \quad + \quad C_2\, \Delta\tilde{x}_2^N \quad + ... + \quad C_N\, \Delta\tilde{x}_N^N \quad] \\
&+ \frac{h^{(N+1)}}{(N+1)!}\, u_\alpha^{(N+1)} \quad [\quad C_1\, \Delta\tilde{x}_1^{(N+1)} \quad + \quad C_2\, \Delta\tilde{x}_2^{(N+1)} \quad + ... + C_N\, \Delta\tilde{x}_N^{(N+1)} \quad] \\
&+ \quad \mathcal{O}(h^{N+2})\,.
\end{aligned}
$$

The values for the coefficients C_p have to be chosen so that all terms in brackets vanish, except for both brackets that multiply the k^{th} and q^{th} derivatives. This leads to the definition of the following $(N+1) \times N$ matrix

$$
\mathbf{M} \triangleq \begin{bmatrix}
1 & 1 & \dots & 1 \\
\Delta\tilde{x}_1 & \Delta\tilde{x}_2 & \dots & \Delta\tilde{x}_N \\
\Delta\tilde{x}_1^2 & \Delta\tilde{x}_2^2 & \dots & \Delta\tilde{x}_N^2 \\
\vdots & \vdots & \ddots & \vdots \\
\Delta\tilde{x}_1^N & \Delta\tilde{x}_2^N & \dots & \Delta\tilde{x}_N^N
\end{bmatrix} .
$$

Next, let us define $\mathbf{M}_q$ as the square matrix including all rows of $\mathbf{M}$, except for the row with the terms $\Delta\tilde{x}_p^q$ (row $q+1$). The row $q+1$ is noted $\mathbf{r}_q^T$ and the stencil can then be expressed as

$$
\begin{aligned}
(S_q^k)_p &= \frac{k!}{h^k} (\mathbf{M}_q^{-1} \mathbf{e}_q^k)_p \\
(S_q^k)_\alpha &= -\frac{h^{q-k} k!}{q!} \mathbf{r}_q^T \mathbf{M}_q^{-1} \mathbf{e}_q^k \,,
\end{aligned}
$$

where $\mathbf{e}^k$ is a $(N+1)$ vector defined by $(\mathbf{e}^k)_i = \delta_{ki}$ and $\mathbf{e}^k_q$ is equal to $\mathbf{e}^k$, without the entry $q+1$. The order of the leading error term is then $s = N+1-k$, which is independent of q. This means that we can impose whatever derivative at x_α, without affecting the order of the leading error term.

For growing N, the conditioning of $\mathbf{M}_q$ deteriorates. However, in our case, $N = 3$ most of the time and the procedure remains applicable without any difficulty.

Case 2 Another similar case is the expression of the k^{th} derivative of u at x_α as a function of N grid point values, without other information.

$$u^{(k)}_\alpha = \sum_{p=1}^{N} T^k_p \, u_p + \mathcal{O}(h^s) \ ,$$

The stencil reads

$$T^k_p \ = \ \frac{k!}{h^k} (\mathbf{M}_N^{-1} \mathbf{e}^k_N)_p \ .$$

Here, the order of the leading error term is $s = N-k$.

Appendix B

Additional immersed interface tools

In this appendix, some additional corrected stencils are provided for the computation of the velocity field and of the wall data required by the mesh-particle interpolation (see Chapter 6).

B.1 Computation of the velocity field

Assuming that the streamfunction Ψ has been computed as the solution of the Poisson equation $\nabla^2\Psi = -\omega$ using the tools from Chapter 4, the velocity field $\mathbf{u} = (u, v)$ is computed as

$$\mathbf{u} = \nabla \times (\Psi \hat{\mathbf{e}}_z) \quad \text{or also} \quad u = \frac{\partial \Psi}{\partial y} \text{ and } v = -\frac{\partial \Psi}{\partial x} . \tag{B.1}$$

The wall correction procedure from Chapter 4 may also be applied here. The classical second order centered scheme for the first derivative can be used by simply adding a correction term when the stencil crosses the wall. Considering an irregular grid point $\mathbf{x}_{ij} = (x_i, y_j)$ and the associated control point $\mathbf{x}_\alpha = (x_i, y_\alpha)$ (along a vertical grid line) with $y_j < y_\alpha < y_{j+1}$, the scheme for u reads

$$\begin{aligned} u_{i,j} &= R_{-1}\, \Psi_{i,j-1} + R_{+1}\, (\Psi_{i,j+1} - J_\alpha^+) \\ u_{i,j+1} &= R_{-1}\, (\Psi_{i,j} - J_\alpha^-) + R_{+1}\, \Psi_{i,j+2} \ , \end{aligned}$$

with $R_{-1} = -1/(2\Delta y)$ and $R_{+1} = 1/(2\Delta y)$. The definitions of the correction terms J_α^+ and J_α^- do not depend on the type of derivative that is being descretized. In Chapter 4, the second derivative was corrected (as opposed to the present first derivative), but the expressions for J_α^+ and J_α^- are still applicable here. Both corrections J_α^+ and J_α^- require the evaluation of the partial derivatives of Ψ at $\mathbf{x}_\alpha$, here up to order 2, in order to preserve the local second order accuracy of the scheme (as opposed to the third order accuracy required in Chapter 4). It should also be observed that only control points on grid lines $x = cst$ are needed here (similarly, v only uses control points on $y = cst$ grid lines).

Assume now that the body is in the domain $y < y_\alpha$. Since $\Psi = \overline{\Psi}$ at $\mathbf{x}_\alpha$ (and inside the body), we can write the following one-sided stencil from Appendix A

$$\left(\frac{\partial^k \Psi}{\partial y^k}\right)_\alpha^+ = (S_0^k)_\alpha \ \overline{\Psi} + \sum_{p=1}^{2} (S_0^k)_p \ \Psi_{i,j+p+1} + \mathcal{O}(h^{3-k}) \ ,$$

As a matter of convenience, one may actually reuse the correction terms J_α^+ and J_α^+ determined for the computation of $\nabla^2\Psi$, (this increases the accuracy, though it is not needed here). The field v may be computed using a similar procedure.

B.2 Computation of the wall data required by the mesh-particle interpolation

In a 2-D vortex particle-mesh method, the following evolution equations have to be solved for all particles

$$\begin{aligned} \frac{d\mathbf{x}_p}{dt} &= \mathbf{u}_p \\ \frac{d\omega_p}{dt} &= \nu \left(\nabla^2 \omega\right)_p \ . \end{aligned}$$

The right hand sides of both equations are first computed on a grid using (corrected) finite differences (the velocity computation is detailed in previous Section B.1 whereas, the computation of the Laplacian of the vorticity is described in Chapter 5). Then, the computed information has to be mapped onto the particles using the M2P interpolation. Applying the M2P approach from

Chapter 6 to those grid fields requires the knowledge of some additional data at the wall. Recalling also from Chapter 5 the notation ξ for the current direction coordinate and η for the transverse coordinate at a control point $\mathbf{x}_\alpha$, we need the following quantities to apply the M2P approach to $\mathbf{u}$ and $\nabla^2\omega$

$$\mathbf{u}_\alpha \ , \quad \left(\frac{\partial \mathbf{u}}{\partial \xi}\right)_\alpha \ , \quad \left(\nabla^2\omega\right)_\alpha \quad \text{and} \quad \frac{\partial}{\partial \xi}\left(\nabla^2\omega\right)_\alpha \ .$$

The computation of $\mathbf{u}_\alpha$ requires the computation of $(\partial\Psi/\partial\xi)_\alpha$ and $(\partial\Psi/\partial\eta)_\alpha$ for u_α and v_α, according to Eq. (B.1). Using the stencils from Appendix A, the derivative in the current direction is simply computed as

$$\left(\frac{\partial \Psi}{\partial \xi}\right)_\alpha = (S_0^1)_\alpha \, \overline{\Psi} + \sum_{p=1}^{3} (S_0^1)_p \, \Psi_{\xi,p} + \mathcal{O}(h^3) \ ,$$

with $\Psi_{\xi,p}$ the solution values at the grid points $\mathbf{x}_{\xi,p}$ shown in Fig. B.1.

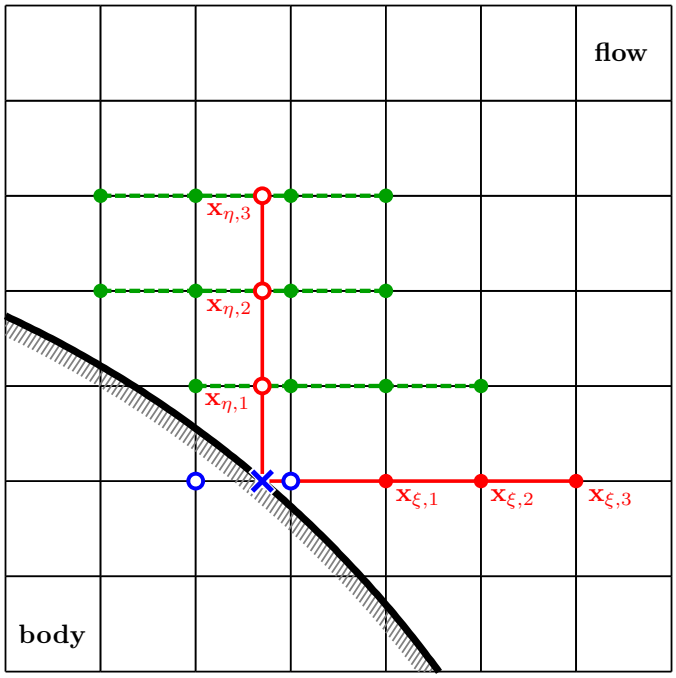


Figure B.1: *Sketch of the different stencil configurations allowing to compute the partial derivatives in the ξ and η directions: control point (blue cross) and the associated irregular points (blue circles); stencil nodes for the current direction ($\mathbf{x}_{\xi,p}$, red bullets); stencil points for the transverse direction ($\mathbf{x}_{\eta,p}$, red circles); stencil nodes required for the interpolation at $\mathbf{x}_{\eta,p}$ (green bullets).*

Similarly to the compatible extrapolation approach from Chapter 5, the derivative in the transverse direction is computed as

$$\left(\frac{\partial \Psi}{\partial \eta}\right)_\alpha = (S_0^1)_\alpha \, \overline{\Psi} + \sum_{p=1}^{3} (S_0^1)_p \, \Psi_{\eta,p} + \mathcal{O}(h^3)$$
$$\Psi_{\eta,p} = \sum_{s=1}^{4} T_s^0 \, \Psi_{\eta,p,s} + \mathcal{O}(h^4) \, ,$$

with $\Psi_{\eta,p,s}$ the solution values at the grid nodes (green bullets in Fig. B.1) that are required to interpolate the value $\Psi_{\eta,p}$ at the points $\mathbf{x}_{\eta,p}$. Both derivatives give us an expression for the wall velocity u_α and v_α.

In order to compute the derivative of $\mathbf{u}$ in the current direction, we first need to compute the velocity field $\mathbf{u}_{ij}$ based on Ψ_{ij}, following Appendix B.1. The next step follows the lines of the above derivation of Ψ. For the u-component, we thus have

$$\left(\frac{\partial u}{\partial \xi}\right)_\alpha = (S_0^1)_\alpha \, u_\alpha + \sum_{p=1}^{3} (S_0^1)_p \, u_{\xi,p} + \mathcal{O}(h) \, .$$

The order of the leading error term is formally 3 when both u_{ij} and u_α are exact. If only u_{ij} was exact, the error would have been $\mathcal{O}(h^2)$, since $(S_0^1)_\alpha \sim h^{-1}$ and the error for u_α is $\mathcal{O}(h^3)$. However, u_{ij} is computed using second order finite differences and hence the error for $(\partial u/\partial \xi)_\alpha$ is $\mathcal{O}(h)$. The v-component is treated equivalently.

For the vorticity we have a Neumann boundary condition $(\partial \omega/\partial n)_\alpha$ and the compatible extrapolation procedure from Chapter 5 has to be applied in order to compute $(\nabla^2 \omega)_\alpha$. Eq. (5.7) provides an expression for ω_α, thus transforming a Neumann boundary condition into a Dirichlet condition. The desired wall quantity can be written as a function of the current and transverse second derivatives

$$(\nabla^2 \omega)_\alpha = \left(\frac{\partial^2 \omega}{\partial \xi^2}\right)_\alpha + \left(\frac{\partial^2 \omega}{\partial \eta^2}\right)_\alpha \, .$$

The latter are provided by

$$\begin{aligned}\left(\frac{\partial^2\omega}{\partial\xi^2}\right)_\alpha &= (S_0^2)_\alpha\,\omega_\alpha + \sum_{p=1}^{3}(S_0^2)_p\,\omega_{\xi,p} + \mathcal{O}(h^2)\\ \left(\frac{\partial^2\omega}{\partial\eta^2}\right)_\alpha &= (S_0^2)_\alpha\,\omega_\alpha + \sum_{p=1}^{3}(S_0^2)_p\,\omega_{\eta,p} + \mathcal{O}(h^2)\\ \omega_{\eta,p} &= \sum_{s=1}^{4} T_s^0\,\omega_{\eta,p,s} + \mathcal{O}(h^4)\;,\end{aligned}$$

The associated derivative in the current direction also requires the precomputation of the field $(\nabla^2\omega)_{ij}$ using the tools from Chapter 5. It is then given by

$$\frac{\partial}{\partial\xi}(\nabla^2\omega)_\alpha = (S_0^1)_\alpha\,(\nabla^2\omega)_\alpha + \sum_{p=1}^{3}(S_0^1)_p\,(\nabla^2\omega)_{\xi,p} + \mathcal{O}(h)\;.$$

Again, the accuracy of the leading error term deteriorates because of the second order field $(\nabla^2\omega)_{ij}$.

Appendix C

Fourier transforms : additional expressions

In this appendix, the Fourier transforms used in Chapter 8 are defined and the expression of some redistribution kernels in the Fourier space are also provided.

C.1 Definitions of the Fourier transforms

The following definitions are taken from [57]:

Fourier Transform (FT): non periodic continuous functions $f(x)$

$$f(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{F}(k)\, e^{jkx}\, dk \longrightarrow \hat{F}(k) = \int_{-\infty}^{\infty} f(x)\, e^{-jkx}\, dx$$

$$x \in \mathbb{R} \qquad\qquad k = \frac{2\pi}{\lambda} \in \mathbb{R}$$

Fourier Series (FS): periodic continuous functions $f(x)$

$$f(x) = \sum_{l=-\infty}^{\infty} \tilde{F}[l]\, e^{jlk_0 x} \xrightarrow{k_0 \triangleq \frac{2\pi}{L}} \tilde{F}[l] = \frac{1}{L} \int_0^L f(x)\, e^{-jlk_0 x}\, dx$$

$$x \in \mathbb{R},\ \text{Period } L \qquad\qquad l \in \mathbb{Z}$$

Discrete Time Fourier Transform (DTFT): non periodic discrete functions $f[p]$

$$f[p] = \frac{1}{2\pi}\int_{-\pi}^{\pi} \hat{F}(K)\, e^{jKp}\, dK \longrightarrow \hat{F}(K) = \sum_{p=-\infty}^{\infty} f[p]\, e^{-jKp}$$

$$p \in \mathbb{Z} \qquad\qquad K \in \mathbb{R},\ \text{Period } 2\pi$$

Discrete Time Fourier Series (DTFS): periodic discrete functions $f[p]$

$$f[p] = \sum_{l=0}^{N-1} \tilde{F}[l]\, e^{jlpK_0} \xrightarrow{K_0 \triangleq \frac{2\pi}{N}} \tilde{F}[l] = \frac{1}{N}\sum_{p=0}^{N-1} f[p]\, e^{-jlpK_0}$$

$$p \in \mathbb{Z},\ \text{Period } N \qquad\qquad l \in \mathbb{Z},\ \text{Period } N$$

C.2 Some redistribution kernels with their Fourier transforms

For a detailed derivation of the following schemes refer to Cottet & Koumoutsakos [34].

- M_1 kernel :

$$M_1(\xi) = \begin{cases} 0 & \text{if } |\xi| > \frac{1}{2}\,, \\ 1 & \text{if } |\xi| \leq \frac{1}{2}\,, \end{cases}$$
$$\mathrm{FT}\{M_1\}(k) = \frac{2}{k}\,\sin\left(\frac{k}{2}\right)\,.$$

- M_2 kernel :

$$M_2(\xi) = \begin{cases} 0 & \text{if } |\xi| > 1\,, \\ 1-|\xi| & \text{if } |\xi| \leq 1\,, \end{cases}$$
$$\mathrm{FT}\{M_2\}(k) = \left(\frac{2}{k}\,\sin\left(\frac{k}{2}\right)\right)^2\,.$$

- M_3 kernel :

$$M_3(\xi) = \begin{cases} 0 & \text{if } |\xi| > \frac{3}{2} \,, \\ \frac{1}{2}(\frac{3}{2} - |\xi|)^2 & \text{if } \frac{1}{2} < |\xi| \leq \frac{3}{2} \,, \\ \frac{1}{2}(\frac{3}{2} + |\xi|)^2 - \frac{3}{2}(\frac{1}{2} + |\xi|)^2 & \text{if } |\xi| \leq \frac{1}{2} \,, \end{cases}$$

$$\mathrm{FT}\{M_3\}(k) = \left(\frac{2}{k}\,\sin\left(\frac{k}{2}\right)\right)^3 .$$

- M_4 kernel :

$$M_4(\xi) = \begin{cases} 0 & \text{if } |\xi| > 2 \,, \\ \frac{1}{6}(2 - |\xi|)^3 & \text{if } 1 < |\xi| \leq 2 \,, \\ \frac{1}{6}(2 - |\xi|)^3 - \frac{4}{6}(1 - |\xi|)^3 & \text{if } |\xi| \leq 1 \,, \end{cases}$$

$$\mathrm{FT}\{M_4\}(k) = \left(\frac{2}{k}\,\sin\left(\frac{k}{2}\right)\right)^4 .$$

- M_4' kernel (see Monaghan [93]) :

$$M_4'(\xi) = \begin{cases} 0 & \text{if } |\xi| > 2 \,, \\ \frac{1}{2}(2 - |\xi|)^2(1 - |\xi|) & \text{if } 1 < |\xi| \leq 2 \,, \\ 1 - \frac{5}{2}|\xi|^2 + \frac{3}{2}|\xi|^3 & \text{if } |\xi| \leq 1 \,, \end{cases}$$

$$\mathrm{FT}\{M_4'\}(k) = -2\cos\left(\frac{k}{2}\right)\left(\frac{2}{k}\,\sin\left(\frac{k}{2}\right)\right)^3 + 3\left(\frac{2}{k}\,\sin\left(\frac{k}{2}\right)\right)^4 .$$

- M_6^{**} kernel (see Bergdorf [8]) :

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

- Λ_2 kernel :

$$\Lambda_2(\xi) = \begin{cases} 0 & \text{if } |\xi| > \frac{3}{2} \,, \\ \frac{1}{2}(1-|\xi|)(2-|\xi|) & \text{if } \frac{1}{2} < |\xi| \leq \frac{3}{2} \,, \\ 1-|\xi|^2 & \text{if } |\xi| \leq \frac{1}{2} \,, \end{cases}$$

- Λ_3 kernel :

$$\Lambda_3(\xi) = \begin{cases} 0 & \text{if } |\xi| > 2 \,, \\ \frac{1}{6}(1-|\xi|)(2-|\xi|)(3-|\xi|) & \text{if } 1 < |\xi| \leq 2 \,, \\ \frac{1}{2}(1-|\xi|^2)(2-|\xi|) & \text{if } |\xi| \leq 1 \,, \end{cases}$$

Appendix D

Vorticity flux associated to a no-slip condition

This appendix complements Chapter 3, in particular Section 3.3, and forms a brief reminder of the developments made in [72] about the way to link the vorticity flux and the vortex sheet in the context of enforcing a no-slip condition at a solid wall in vortex methods.

Let us consider the 2-D flow past a solid body with a prescribed velocity boundary condition $\mathbf{u} = \mathbf{u}_b(\mathbf{x}, t)$ on its boundary $\partial\Omega_b$. Consider also the vorticity field at time $t + \Delta t$ resulting from the advancement of the solution from t to $t + \Delta t$ by solving Eq. (3.2) using a no-through flow condition on $\partial\Omega_b$ and starting from an admissible vorticity field at time t (i.e. in addition to the no-through flow condition, the solution also satisfies the no-slip condition on $\partial\Omega_b$ at t).

The velocity field at $t+\Delta t$, satisfying the no-through flow condition $\mathbf{u}\cdot\mathbf{n} = \mathbf{u}_b\cdot\mathbf{n}$ ($\mathbf{n}$ being the outward pointing normal vector to $\partial\Omega_b$), can be obtained by solving the Poisson Eq. (3.3) as a function of the vorticity field at $t+\Delta t$. Yet, as explained in Section 3.3, this vorticity field has likely become non-admissible, since it has been advanced in time only enforcing the no-through flow condition. The velocity field hence presents a non-zero tangential slip velocity at the wall that can also be seen as an infinitely thin vortex sheet $\Delta\gamma$.

This vortex sheet is also obtained, along with the velocity, when solving the Poisson equation. However, the flow domain $\Omega_f = \mathbb{R}^2/\Omega_b$ being not simply-connected, the uniqueness of the Poisson equation solution requires the addition

of an integral constraint on the vortex sheet circulation $\Gamma_{\Delta\gamma}$ (see [65])

$$\Gamma_{\Delta\gamma} = \oint_{\partial\Omega_b} \Delta\gamma \, ds \, . \tag{D.1}$$

According to Lighthill's model [79], the vortex sheet is part of the flow, as it represents the amount of vorticity (yet, it is still singular) that complements the actual vorticity field so as to yield the no-slip condition just below the panel (on the body side).

Considering the flow circulation $\Gamma_f \triangleq \int_{\Omega_f} \omega \, d\mathbf{x}$ and Kelvin's theorem in Ω_f, we further obtain

$$\frac{d\Gamma_f}{dt} = \int_{\Omega_f} \nu\nabla^2\omega \, d\mathbf{x} = -\oint_{\partial\Omega_b} \nu\frac{\partial\omega}{\partial n} \, d\mathbf{x} \, ,$$

because $\omega \to 0$ at infinity. If the body is subjected to a solid body rotation speed $W_b(t)$, this movement can be represented by a uniform extension of the vorticity $\omega = 2W_b(t)$ inside the body Ω_b, thus implying a body circulation $\Gamma_b \triangleq \int_{\Omega_b} \omega \, d\mathbf{x} = 2W_bS_b$, with S_b the area of the body. Applying again Kelvin's theorem in the whole domain $\mathbb{R}^2$, we obtain

$$\frac{d\Gamma_f}{dt} = -2S_b \, \frac{dW_b}{dt} \, .$$

Integrating this equation over the time interval $[t, t+\Delta t]$ results in

$$\Delta\Gamma_f \triangleq \int_t^{t+\Delta t} \frac{d\Gamma_f}{dt'} \, dt' = -2S_b \, [W_b(t+\Delta t) - W_b(t)] \, .$$

Based on the interpretation of the vortex sheet as a part of the flow, we can model the increment $\Delta\Gamma_f$ of circulation coming from the body and that must enter the fluid as

$$\Delta\Gamma_f = \oint_{\partial\Omega_b} \Delta\gamma \, d\mathbf{x} \, ,$$

which leads to

$$-\oint_{\partial\Omega_b} \left(\int_t^{t+\Delta t} \nu\frac{\partial\omega}{\partial n} \, dt' \right) \, d\mathbf{x} = \oint_{\partial\Omega_b} \Delta\gamma \, d\mathbf{x} \, .$$

We may furthermore assume that this relation holds locally, given the interpretation of the vortex sheet as the vorticity feeding the boundary layer (by

means of the vorticity flux), so as to cancel the residual slip velocity. Thus we have

$$\Delta\gamma = -\int_t^{t+\Delta t} \nu \frac{\partial \omega}{\partial n}\, dt \quad \text{or} \quad \frac{\partial(\Delta\gamma)}{\partial t} = -\nu \frac{\partial \omega}{\partial n}\,,$$

if the vortex sheet is given a time continuous meaning over the time step (i.e. $\Delta\gamma(t)$ is a smooth function of the time and $\Delta\gamma \to 0$ when $\Delta t \to 0$), as suggested in [33]. Despite being derived for a simply-connected domain, the above relation is assumed to remain valid for any incompressible viscous flow in the presence of solid walls and the additional assumption that the flux is constant over the time step leads to the following widely used expression

$$-\nu \frac{\partial \omega}{\partial n} = \frac{\Delta\gamma}{\Delta t}\,.$$

When $W_b(t)$ is constant, the increment of total circulation is $\Delta\Gamma_f = 0$, but the local vorticity flux is of course still non zero.

Appendix E

Time integration schemes for VPM methods

Several time integration schemes complementing Chapter 3 are presented here. This appendix considers time integration schemes that solve the Navier-Stokes equations, on the one hand, with a *prescribed* vorticity flux condition at the wall and, on the other hand, with a *no slip* condition at the wall (the vorticity flux is then linked to the residual vortex sheet). The computation of the wall contributions (for both aforementioned flux types) within the different schemes is provided in Appendix E.7. In Appendix E.8, some details are also given about the "PW" scheme [102] used for the computation of the analytical solution associated to the near-wall diffusion.

For the following, computational steps of the RK2 variants that differ from the original RK2 solver from Section 3.2 are colored dark red, in order to facilitate the distinction.

E.1 Low storage RK3

In Chapter 3, a VPM solver enforcing a no-through flow condition at the solid wall was presented, using a mid-point second order Runge-Kutta scheme. The following VPM solver scheme is based on a low storage third order Runge-Kutta scheme from [130]. The notations are similar to those of Chapter 3:

Sub step 1 : from t^n to $t^{n+\frac{1}{3}} \triangleq t^n + \frac{1}{3}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$(\mathbf{G_u})_p \leftarrow \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{3}} = \mathbf{x}_p^n + \frac{1}{3}\,\Delta t\,\,(\mathbf{G_u})_p$$

- *Diffusion :*
$$\omega_{ij}^n \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^n \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^n = q_\omega^n$$
$$\left(\nabla^2\omega\right)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_p^n$$
$$(G_{\nabla^2\omega})_p \leftarrow \nu\left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{3}} = \omega_p^n + \frac{1}{3}\,\Delta t\,\,(G_{\nabla^2\omega})_p$$
$$\omega_p^{n+\frac{1}{3}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{3}}} \omega_{ij}^{n+\frac{1}{3}}$$

Sub step 2 : from $t^{n+\frac{1}{3}}$ to $t^{n+\frac{3}{4}} \triangleq t^n + \frac{3}{4}\Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{3}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{3}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{3}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{3}}} \mathbf{u}_p^{n+\frac{1}{3}}$$
$$(\mathbf{G_u})_p \leftarrow -\frac{5}{9}\,\,(\mathbf{G_u})_p + \mathbf{u}_p^{n+\frac{1}{3}}$$
$$\mathbf{x}_p^{n+\frac{3}{4}} = \mathbf{x}_p^{n+\frac{1}{3}} + \frac{15}{16}\,\Delta t\,\,(\mathbf{G_u})_p$$

- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{3}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{3}} \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n+\frac{1}{3}} = q_\omega^{n+\frac{1}{3}}$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{3}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{3}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{3}}$$
$$(G_{\nabla^2\omega})_p \leftarrow -\frac{5}{9}\,\,(G_{\nabla^2\omega})_p + \nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{3}}$$
$$\omega_p^{n+\frac{3}{4}} = \omega_p^{n+\frac{1}{3}} + \frac{15}{16}\,\Delta t\,\,(G_{\nabla^2\omega})_p$$
$$\omega_p^{n+\frac{3}{4}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{3}{4}}} \omega_{ij}^{n+\frac{3}{4}}$$

Sub step 3 : from $t^{n+\frac{3}{4}}$ to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :* $\omega_{ij}^{n+\frac{3}{4}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{3}{4}}$

  $\mathbf{u}_{ij}^{n+\frac{3}{4}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{3}{4}}} \mathbf{u}_p^{n+\frac{3}{4}}$

  $$(\mathbf{G_u})_p \leftarrow -\frac{153}{128}\,(\mathbf{G_u})_p + \mathbf{u}_p^{n+\frac{3}{4}}$$

  $$\mathbf{x}_p^{n+1} = \mathbf{x}_p^{n+\frac{3}{4}} + \frac{8}{15}\,\Delta t\,(\mathbf{G_u})_p$$

- *Diffusion :* $\omega_{ij}^{n+\frac{3}{4}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{3}{4}}$ with $-\nu\,\dfrac{\partial\omega}{\partial n}\Big|_{\partial\Omega}^{n+\frac{3}{4}} = q_\omega^{n+\frac{3}{4}}$

  $\left(\nabla^2\omega\right)_{ij}^{n+\frac{3}{4}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{3}{4}}} \left(\nabla^2\omega\right)_p^{n+\frac{3}{4}}$

  $$(G_{\nabla^2\omega})_p \leftarrow -\frac{153}{128}\,(G_{\nabla^2\omega})_p + \nu\left(\nabla^2\omega\right)_p^{n+\frac{3}{4}}$$

  $$\omega_p^{n+1} = \omega_p^{n+\frac{3}{4}} + \frac{8}{15}\,\Delta t\,(G_{\nabla^2\omega})_p$$

  $\omega_p^{n+1} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1}$

Redistribution : after n^r time steps, reinitialize the set of particles.

E.2 DRK2-END

Decomposed RK2 scheme computing the wall contribution at the END of the time step.

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,\mathbf{u}_p^n$$

- *Diffusion :*
$$\omega_{ij}^n \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^n \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^n = 0$$
$$\left(\nabla^2\omega\right)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2}} = \omega_p^n + \frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \mathbf{u}_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\;\mathbf{u}_p^{n+\frac{1}{2}}$$

- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n+\frac{1}{2}} = 0$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} = \omega_p^n + \Delta t\,\nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1,*}$$

- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+1}$ (see App. E.7 step **B**)
$$\omega_{ij}^{n+1} = \omega_{ij}^{n+1,*} + (\omega_w)_{ij}^{n+1}$$
$$(\omega_w)_{ij}^{n+1} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+1}} (\omega_w)_p^{n+1}$$
$$\omega_p^{n+1} = \omega_p^{n+1,*} + (\omega_w)_p^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles:

E.3 DRK2-SUB

Decomposed RK2 scheme computing the wall contributions at the end of every SUB step.

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,\mathbf{u}_p^n$$

- *Diffusion :*
$$\omega_{ij}^n \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^n \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n} = 0$$
$$\left(\nabla^2\omega\right)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2},*} = \omega_p^n + \frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2},*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2},*}$$

- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+\frac{1}{2}}$ (see App. E.7 step **A**)
$$\omega_{ij}^{n+\frac{1}{2}} = \omega_{ij}^{n+\frac{1}{2},*} + (\omega_w)_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \mathbf{u}_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\,\mathbf{u}_p^{n+\frac{1}{2}}$$

- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n+\frac{1}{2}} = 0$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} = \omega_p^n + \Delta t\,\nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1,*}$$

- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+1}$ (see App. E.7 step **B**)
$$\omega_{ij}^{n+1} = \omega_{ij}^{n+1,*} + (\omega_w)_{ij}^{n+1}$$
$$(\omega_w)_{ij}^{n+1} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+1}} (\omega_w)_p^{n+1}$$
$$\omega_p^{n+1} = \omega_p^{n+1,*} + (\omega_w)_p^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles.

E.4 DRK2-CSUB

Decomposed RK2 scheme computing Convected wall contributions at the end of every SUB step.

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,\mathbf{u}_p^n$$
- *Diffusion :*
$$\omega_{ij}^n \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^n \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^n = 0$$
$$\left(\nabla^2\omega\right)_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2},*} = \omega_p^n + \frac{\Delta t}{2}\,\nu\left(\nabla^2\omega\right)_p^n$$
$$\omega_p^{n+\frac{1}{2},*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2},*}$$
- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+\frac{1}{2}}$ (see App. E.7 step **A**)
$$(\omega_w)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^n} (\omega_w)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+\frac{1}{2}} = \omega_p^{n+\frac{1}{2},*} + (\omega_w)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+\frac{1}{2}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \omega_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \mathbf{u}_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\,\mathbf{u}_p^{n+\frac{1}{2}}$$
- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n+\frac{1}{2}} = 0$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} = \omega_p^n + \Delta t\,\nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1,*}$$
- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+1}$ (see App. E.7 step **B**)
$$(\omega_w)_{ij}^{n+1} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} (\omega_w)_p^{n+1}$$
$$\omega_p^{n+1} = \omega_p^{n+1,*} + (\omega_w)_p^{n+1}$$
$$\omega_p^{n+1} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles.

E.5 SRK2

Second order Split RK2 scheme.

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,\mathbf{u}_p^n$$
$$\tilde{\omega}_p^{n+\frac{1}{2}} = \omega_p^n$$
$$\tilde{\omega}_p^{n+\frac{1}{2}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \tilde{\omega}_{ij}^{n+\frac{1}{2}}$$
- *Diffusion :*
$$\tilde{\omega}_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\tilde{\omega}\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\,\frac{\partial\omega}{\partial n}\bigg|_{\partial\Omega}^{n} = q_\omega^n$$
$$\omega_{ij}^{n+\frac{1}{2}} = \tilde{\omega}_{ij}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\,\nu\left(\nabla^2\tilde{\omega}\right)_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \mathbf{u}_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\,\mathbf{u}_p^{n+\frac{1}{2}}$$
$$\tilde{\omega}_p^{n+1} = \omega_p^n$$
- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\,\frac{\partial\omega}{\partial n}\bigg|_{\partial\Omega}^{n+\frac{1}{2}} = q_\omega^{n+\frac{1}{2}}$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1} = \tilde{\omega}_p^{n+1} + \Delta t\,\nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles.

E.6 DSRK2-CSUB

Decomposed second order Split RK2 scheme computing Convected wall contributions at the end of every SUB step.

Predictor : from t^n to $t^{n+\frac{1}{2}} \triangleq t^n + \frac{1}{2}\Delta t$

- *Advection :*
$$\omega_{ij}^n \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^n$$
$$\mathbf{u}_{ij}^n \xrightarrow{\text{M2P to } \mathbf{x}_p^n} \mathbf{u}_p^n$$
$$\mathbf{x}_p^{n+\frac{1}{2}} = \mathbf{x}_p^n + \frac{\Delta t}{2}\,\mathbf{u}_p^n$$
$$\tilde{\omega}_p^{n+\frac{1}{2}} = \omega_p^n$$
$$\tilde{\omega}_p^{n+\frac{1}{2}} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+\frac{1}{2}}} \tilde{\omega}_{ij}^{n+\frac{1}{2}}$$
- *Diffusion :*
$$\tilde{\omega}_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\tilde{\omega}\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n} = 0$$
$$\omega_{ij}^{n+\frac{1}{2},*} = \tilde{\omega}_{ij}^{n+\frac{1}{2}} + \frac{\Delta t}{2}\,\nu\left(\nabla^2\tilde{\omega}\right)_{ij}^{n+\frac{1}{2}}$$
- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+\frac{1}{2}}$ (see App. E.7 step **A**)
$$\omega_{ij}^{n+\frac{1}{2}} = \omega_{ij}^{n+\frac{1}{2},*} + (\omega_w)_{ij}^{n+\frac{1}{2}}$$

Corrector : from t^n to $t^{n+1} \triangleq t^n + \Delta t$

- *Advection :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2}}$$
$$\mathbf{u}_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \mathbf{u}_p^{n+\frac{1}{2}}$$
$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t\,\mathbf{u}_p^{n+\frac{1}{2}}$$
$$\tilde{\omega}_p^{n+1} = \omega_p^n$$
$$\tilde{\omega}_p^{n+1} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \tilde{\omega}_{ij}^{n+1}$$
- *Diffusion :*
$$\omega_{ij}^{n+\frac{1}{2}} \xrightarrow{\nabla^2(\cdot)} \left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \quad \text{with} \quad -\nu\left.\frac{\partial\omega}{\partial n}\right|_{\partial\Omega}^{n+\frac{1}{2}} = q_\omega^{n+\frac{1}{2}}$$
$$\left(\nabla^2\omega\right)_{ij}^{n+\frac{1}{2}} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} \left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} = \tilde{\omega}_p^{n+1} + \Delta t\,\nu\left(\nabla^2\omega\right)_p^{n+\frac{1}{2}}$$
$$\omega_p^{n+1,*} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1,*}$$
- *Near-wall diff. :* Compute the wall contribution $(\omega_w)_{ij}^{n+1}$ (see App. E.7 step **B**)
$$(\omega_w)_{ij}^{n+1} \xrightarrow{\text{M2P to } \mathbf{x}_p^{n+\frac{1}{2}}} (\omega_w)_p^{n+1}$$
$$\omega_p^{n+1} = \omega_p^{n+1,*} + (\omega_w)_p^{n+1}$$
$$\omega_p^{n+1} \xrightarrow{\text{P2M from } \mathbf{x}_p^{n+1}} \omega_{ij}^{n+1}$$

Redistribution : after n^r time steps, reinitialize the set of particles.

E.7 Computation of the wall contributions

Prescribed flux :

$$\textbf{A} \quad \bar{q}_\omega^n = \frac{1}{\Delta t/2}\int_{t^n}^{t^{n+\frac{1}{2}}} q_\omega \, dt \quad \xrightarrow{\text{PW}(\Delta t/2)} \quad (\omega_w)_{ij}^{n+\frac{1}{2}}$$

$$\textbf{B} \quad \bar{q}_\omega^{n+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} q_\omega \, dt \quad \xrightarrow{\text{PW}(\Delta t)} \quad (\omega_w)_{ij}^{n+1}$$

No-slip enforcement :

$$\textbf{A} \quad \omega_{ij}^{n+\frac{1}{2},*} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+\frac{1}{2},*}, \ \Delta\gamma^{n+\frac{1}{2},*}$$
$$\bar{q}_\omega^n = \frac{\Delta\gamma^{n+\frac{1}{2},*}}{\Delta t/2} \xrightarrow{\text{PW}(\Delta t/2)} (\omega_w)_{ij}^{n+\frac{1}{2}}$$

$$\textbf{B} \quad \omega_{ij}^{n+1,*} \xrightarrow{\text{solve Eq. (3.3)}} \mathbf{u}_{ij}^{n+1,*}, \ \Delta\gamma^{n+1,*}$$
$$\bar{q}_\omega^{n+\frac{1}{2}} = \frac{\Delta\gamma^{n+1,*}}{\Delta t} \xrightarrow{\text{PW}(\Delta t)} (\omega_w)_{ij}^{n+1}$$

E.8 Explicit integral formulas for the near-wall diffusion (PW)

The following 2-D diffusion problem can be solved analytically in the half plane $y \geq 0$

$$\begin{aligned} \frac{\partial \omega_w}{\partial t} &= \nu \nabla^2 \omega_w \\ \omega_w(x,y,0) &= 0 \\ -\nu \frac{\partial \omega_w}{\partial n}(x,0,t) &= \begin{cases} 0 & \text{if } |x| > \frac{b}{2} \\ \bar{q}_\omega = cst & \text{if } |x| \leq \frac{b}{2} \end{cases}, \end{aligned}$$

which corresponds to a panel located on the x axis in the segment $[-\frac{b}{2}, \frac{b}{2}]$ and diffusing a uniform and constant flux $\bar{q}_\omega$ into the flow. According to Ploumhans & Winckelmans [102], we can write

$$\frac{\partial \omega_w}{\partial t}(x,y,t) = \bar{q}_\omega \frac{1}{\sqrt{\pi}} \frac{1}{\sqrt{4\nu t}} \exp\left(\frac{-y^2}{4\nu t}\right) [\text{erfc}(s)]_{(x+\frac{b}{2})/\sqrt{4\nu t}}^{(x-\frac{b}{2})/\sqrt{4\nu t}}.$$

Let us consider a particle with coordinates $\mathbf{x}_p = (x_p, y_p)$ in the local reference system of the panel. As a result of the near-wall diffusion occurring between

$t = 0$ and $t = \Delta T$, this particle receives an increment of circulation defined by

$$\begin{aligned}
\Delta\Gamma_p &\triangleq \int_0^{\Delta T} \frac{d\Gamma_p}{dt}\, dt = \int_0^{\Delta T} \int_{x_p-\frac{h}{2}}^{x_p+\frac{h}{2}} \int_{y_p-\frac{h}{2}}^{y_p+\frac{h}{2}} \frac{\partial \omega_w}{\partial t}\, dy\, dx\, dt \\
\frac{d\Gamma_p}{dt} &= \bar{q}_\omega \left[\mathrm{erfc}(s)\right]_{(y_p+\frac{h}{2})/\sqrt{4\nu t}}^{(y_p-\frac{h_l}{2})/\sqrt{4\nu t}} \\
&\quad \cdot \frac{1}{2}\sqrt{4\nu t}\left(\left[\mathrm{ierfc}(s)\right]_{(x_p-\frac{b}{2}+\frac{h}{2})/\sqrt{4\nu t}}^{(x_p-\frac{b}{2}-\frac{h}{2})/\sqrt{4\nu t}} - \left[\mathrm{ierfc}(s)\right]_{(x_p+\frac{b}{2}+\frac{h}{2})/\sqrt{4\nu t}}^{(x_p+\frac{b}{2}-\frac{h}{2})/\sqrt{4\nu t}}\right) ,
\end{aligned}$$

with $h_l/2 = y_p$ if $0 \leq y_p \leq h$ and $h_l/2 = h/2$ otherwise. The time integral must be computed numerically using a Gauss quadrature (3 or 4 points). If the particles are not aligned with the local panel lattice, a least-square correction can be applied in order to ensure conservation

$$\Delta\Gamma_{p,\mathrm{conserv}} = \Delta\Gamma_p + \frac{(\Delta\Gamma_p)^2}{\sum_p (\Delta\Gamma_p)^2}\left(b\,\bar{q}_\omega\,\Delta T - \sum_p \Delta\Gamma_p\right) .$$

The average vorticity associated to this increment of circulation is then simply

$$\Delta\omega_w \triangleq \frac{\Delta\Gamma_p}{h^2} .$$

The diffusion from the panel affects only a limited number of particles. Typically, a check is performed so as to identify the particles that lie in the domain of influence of the panel defined by a characteristic diffusion length, i.e. a particle p belongs to this domain (and hence receives an increment of circulation), if

$$|\mathbf{x}_p| \lesssim 5\sqrt{4\nu\Delta T} .$$

Every particle has to account for all panels that include its position inside their domain of influence.

Given a boundary flux $\bar{q}_\omega$ (or equivalently a set of panels m with a prescribed flux $(\bar{q}_\omega)_m$), the application of this scheme in order to compute the grid field $(\omega_w)_{ij}$, associated to the near-wall diffusion from $t = 0$ to $t = \Delta T$, is succinctly referred to as

$$\bar{q}_\omega \quad \xrightarrow{\mathrm{PW}(\Delta T)} \quad (\omega_w)_{ij} ,$$

where PW stands for *Ploumhans & Winckelmans* [102].

Appendix F

DDES of the flow past a pair of cylinders in tandem configuration

The content of this chapter is based on a proceedings paper [86] written for the "*AIAA BANC-I workshop*" in June 2010 (Stockholm). It is entitled "*DDES of the flow past a pair of cylinders in tandem configuration*"; the authors are Y. Marichal, C. Carton, L. Bricteux, M. Duponcheel, G. Winckelmans and P. Geuzaine.

Abstract This paper concerns the unsteady turbulent flow simulation around a pair of cylinders in tandem configuration at a high Reynolds number. The study of this flow and its acoustics are relevant as it includes similar physical phenomena as those related to landing gear noise. The flow simulation is here performed using a parallel implicit compressible flow solver for unstructured tetrahedral meshes developed at Cenaero. As turbulent model, the Delayed Detached Eddy Simulation approach (DDES) of Spalart-Allmaras is used. In order to capture the flow physics properly, a special care is devoted to the spatial discretization, here using a kinetic-energy conserving central scheme in the LES part of the DDES and an AUSM scheme in the RANS part. The grid is set-up with multiple refinement zones to capture properly the free shear layers emanating from the upstream cylinder, the wake interaction with the downstream cylinder and the vortex shedding of the downstream cylinder. The

Mach number of the flow corresponds to that of the experiments: $M = 0.1285$. The Reynolds number based on the upstream velocity and the cylinder diameter is set to $Re = 1.66 \times 10^5$. The spanwise extent of the mesh is set to $L_z = 4\,D$ with D the cylinder diameter. A very refined case is also investigated, then limited to $L_z = D$. Mean flow diagnostics and turbulence quantities are provided and compared to available experimental data [82].

F.1 Introduction

During approach or landing, the noise generated by an aircraft is issued from different sources. Among these sources, there are of course the engines but also all the lift enhancing devices and the landing gears. The noise generated by these later sources is called airframe noise. This airframe noise constitutes a non negligible part of the total noise generated by a manoeuvering aircraft. In this work, the focus lies on the landing gear. The source of noise results from the turbulent flow over this bluff body. Various parts of a landing gear can be modeled as closely spaced nearly cylindrical bodies. For this purpose, the study of the flow over a pair of cylinders in tandem configuration is relevant. Several experimental studies have already been performed on this flow. For instance, aerodynamic data were gathered in the work of Khorrami et al. [68]. In the work of Jenkins et al. [63], a PIV investigation was performed. A 2-D URANS investigation was also performed by Doolan [42]. The originality of the present work consists in using a DDES approach for the turbulence modeling and in studying the influence of the cylinder spanwise length.

F.2 Numerical method

The simulation of the studied case was performed with a compressible parallel hybrid finite volume - finite element solver developed at Cenaero [53]. The convective fluxes are described by a finite volume formulation by means of a second order energy conserving centered scheme designed to perform LES on unstructured grids. The local use of upwind schemes, for example in strong gradient regions, is also provided (e.g. AUSM, JST, Roe,...). The diffusive terms are discretized using a classical $P1$ Galerkin formulation resulting in a global second order spatial numerical scheme. The time discretization is also second order and fully implicit (Three-point Backward Difference, 3BDF).

In order to solve the system efficiently, a Newton-Krylov-Schwarz method is used (the non linear system has to be solved at every time step by means of a dual time stepping technique). As the Reynolds number of the flow is very high, turbulence modeling is required. The Delayed Detached Eddy Simulation (DDES) approach [117] has been used.

F.3 Problem description

F.3.1 Tandem cylinders

A sketch of the geometrical configuration of the pair of cylinders is illustrated in Fig. F.1. The distance between the cylinders is set to $L/D = 3.7$. Two geometries have been studied, differing from each other in the spanwise extension L_z of the periodic computational domain : $L_z = 4\,D$ and $L_z = D$. This is to be compared with the geometry used for the two experimental campaigns that have been caried out : The BART campaign (Basic aerodynamic research tunnel test, see e.g. Lockard et al. [81], Neuhart et al. [94]) was made on a configuration with an extension of $L_z = 12.4\,D$, the QFF campaign (quiet flow facility, see e.g. Lockard et al. [81]) was made on a configuration with an extension of $L_z = 16\,D$. This justifies the use of a periodic computational domain, as the influence of the side walls can be ommited for such an extension. The extent of the bounding box corresponds to $105\,D$ in the flow direction and to $70\,D$ in the direction perpendicular to the flow. This aims at preventing any border effects due to the finite size of the domain.

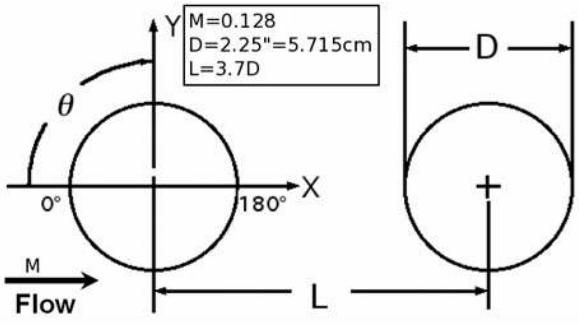


Figure F.1: *Flow configuration*

F.3.2 Simulated conditions

The Reynolds and Mach numbers were taken as those of the experiment: $Ma = 0.1285$, $Re = 1.66 \times 10^5$. At this Reynolds number, the range of scales in the flow is very large and this requires a strong turbulent model to take into account the effect of the unresolved scales. In bluff body aerodynamics, a well suited approach is DDES (Delayed Detached Eddy Simulation, see [117]) which allows to treat the boundary layers with a RANS model. On the contrary, in the core of the flow, the LES is a well performing approach where only the large scales of physical interest are captured, meaning that more turbulent fluctuations are captured. The efficiency of this approach lies in the fact that the RANS method requires far less grid points in the streamwise and spanwise direction in the boundary layers, compared to LES. The model is based on a specific RANS formulation for the near-wall regions of the flow that behaves as a LES model in the core of the flow. In our case, the Spalart-Allmaras model [116] has been adopted (see e.g. Strelets et al. [119], Spalart et al. [117]). The use of this turbulent model requires specific spatial descretization schemes : In the LES zone an energy-conserving centered scheme is used and, in the RANS zone near to the wall, an AUSM upwind scheme is required to ensure the stability of the simulation. The AUSM scheme is also used at a certain distance of the cylinders in order to smooth out the fluctuations due to the mesh coarsening.

Three cases have been studied with different spanwise lengths and mesh refinement parameters. The principal mesh has a spatial extent $L_z = 4\,D$ and has been designed to fit with the available computational resources (mesh $M1$). In order to assess the dependence of the results on the spanwise extent of the periodic computational domain, a second mesh with the same resolution was build with $L_z = D$ (mesh $M2$). A third mesh with $L_z = D$ was also generated with a very fine resolution (mesh $M3$) : it allows to show that the results are essentially mesh independent. As there is a change in the spanwise extent of the periodic domain between $M1$ and $M2$, the flow configuration is not identical. However, only the modes with the largest wavelength are affected. We can check if the small structures of the flow are captured properly by comparing the results from $M2$ with those of the refined mesh $M3$. A comparison for the large scales of the flow is less needed as those are less dependent of the grid size but rather of the spanwise extent of the mesh.

As will be seen in Section F.3.3, the boundary layers are wall-resolved and there is no need for any supplementary wall model. The free flow (external

boundaries) is modeled by computing Riemann fluxes using the upstream flow conditions and by taking into account the flow direction. The combination between the important global size of the domain and the use of the upwind AUSM scheme far from the cylinders avoids the need for an artificial buffer layer. Indeed, it was observed that the present boundary condition implementation performed quite well in convecting turbulent structures outside the domain and the acoustic reflection was negligible.

The time discretization uses a fully implicit integration that allows to choose a time step independently of acoustic considerations. In order to capture properly the physics, the number defined by $CFL = \frac{\max\|\vec{u}\|\Delta t}{\Delta x}$ which behaves roughly like the classical CFL number, should be of order unity (Δx is the streamwise wall mesh size). This guarantees good convergence of the solver. The current time values of $\Delta t* = \frac{\Delta t U}{D}$, CFL and the sampling parameters are given in Table F.1 (U is the upstream velocity). The mean values have been computed over $N_{\Delta t}$ time steps after having reached a statistically converged regime, which corresponds to $T_{tot}U/D$ convective times and $T_{tot}f_s$ main shedding periods. The statistical data have been computed by taking samples over every time step.

Mesh	Δt^*	CFL	$N_{\Delta t}$	$T_{tot}U/D$	$T_{tot}f_s$	f_s	f_{min}	f_{max}
$M1$	$7.69 \cdot 10^{-3}$	0.74	17655	135.3	39.9	226	5.67	50k
$M2$	$7.69 \cdot 10^{-3}$	0.76	17143	131.5	41.8	244	5.83	50k
$M3$	$3.84 \cdot 10^{-3}$	0.86	34247	131.5	39.7	232	5.84	100k

Table F.1: *Temporal parameters (frequencies are expressed in Hz).*

F.3.3 Grid generation

The number of grid points and the characteristics of the three meshes are summarized in Table F.2 and Table F.3. The last table provides mesh sizes of the cells adjacent to the walls (for the upstream and downstream cylinders). The surface mesh on the upstream cylinder is designed to be pseudo-structured (cartesian mesh) in order to be aligned with the mean flow direction. It was also chosen to handle the downstream cylinder in the same way. Δx gives the size of the cells in the flow direction, Δy is the thickness of the first layer (perpendicular to the wall surface) and Δz the mesh size in the spanwise direction.

The quantity $\Delta y^+_{max} = \max \frac{u_\tau \Delta y}{\nu}$ indicates if the boundary layer velocity gradient is well resolved ($u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity based on the wall shear stress τ_w). It is approximately $O(1)$ on both cylinders, as it should be for wall-resolved computations. The mesh refinement procedure can be visualized in the top part of Fig. F.2, displaying in the upper part the mesh corresponding to $M1$ and $M2$ and in the lower part the finer mesh $M3$.

The thickness of the different boundary mesh layers follows a geometric progression with a rate of 1.1 as the wall distance increases. As can be seen in Fig. F.2, there are also refinement zones in the wake of the cylinders, but as the mesh has an octree topology inside the volume, only grid size transitions corresponding to a factor 2 are allowed. The most refined zone covers the recirculation of the upstream cylinder in order to capture as many as possible structures coming from the upstream cylinder as they directly influence the flow configuration at the downstream cylinder. A first choice consisted in defining two refinement patches covering only the detached shear layers from cylinder 1 but convergence considerations showed that the whole wake should be refined. The physical reason for this is the recirculation phenomenon reinserting small structures from the coarser zone into the more refined patch. In this case, the grid transition factor is too high to capture the reinsertion properly.

Contrary to the volume mesh, the surface mesh on the external borders of the domain is unstructured and isotropic. Therefore, transition regions are provided to connect the core grid to the internal and external boundaries.

Mesh	N_{points}	N_{cells}	N_{cpu}	L_z/D
$M1$	$9.85\,M$	$58.1\,M$	400	4
$M2$	$2.67\,M$	$15.2\,M$	64	1
$M3$	$10.9\,M$	$63.4\,M$	256	1

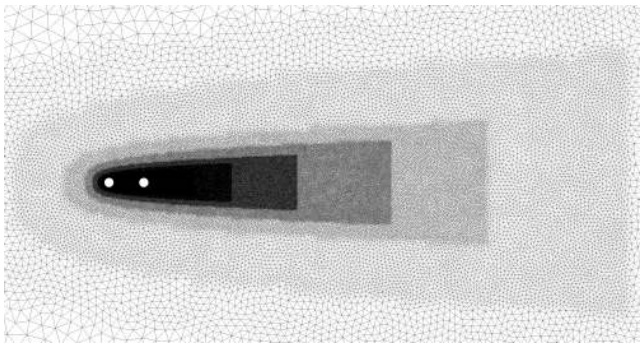
Table F.2: *Mesh description : global quantities.*

F.4 Results

The results obtained numerically in the present work are compared as much as possible with the experimental data. A first insight into the flow configuration for the case $M1$ ($L_z = 4\,D$) is given in Fig.F.3, showing a snapshot of an iso-value surface of the λ_2 criterion. In Fig.F.4, another snapshot is represented,

Mesh		$\Delta x/D$	$\Delta y/D$	$\Delta z/D$	Δy^+_{max}
$M1$	up	$1.75 \cdot 10^{-2}$	$1.75 \cdot 10^{-4}$	$1.75 \cdot 10^{-2}$	2.39
	down	$2.62 \cdot 10^{-2}$	$3.50 \cdot 10^{-4}$	$3.50 \cdot 10^{-2}$	4.64
$M2$	up	$1.75 \cdot 10^{-2}$	$1.75 \cdot 10^{-4}$	$1.75 \cdot 10^{-2}$	2.41
	down	$2.62 \cdot 10^{-2}$	$3.50 \cdot 10^{-4}$	$3.50 \cdot 10^{-2}$	4.62
$M3$	up	$7.70 \cdot 10^{-3}$	$1.50 \cdot 10^{-4}$	$1.05 \cdot 10^{-2}$	2.01
	down	$1.31 \cdot 10^{-2}$	$2.10 \cdot 10^{-4}$	$1.75 \cdot 10^{-2}$	2.68

Table F.3: *Mesh description for cells adjacent to the wall : upstream cylinder (up), downstream cylinder (down).*



showing a spanwise cut of the z-component of the vorticity is shown for the case $M3$. One can appreciate the complexity of the flow and the multi scales of physical importance : very thin boundary layers, detached shear layers, large coherent vortices, turbulent vortices, etc.

F.4.1 Pressure coefficients

The first quantitative diagnostics to be compared with the experiments are the pressure coefficients displayed in Fig. F.5 and defined by:

$$C_p = \frac{p_\infty - p}{\frac{1}{2}\rho U^2}\,. \tag{F.1}$$

A good agreement is globally observed between the different results. For the upstream cylinder, the peaks are slightly overestimated by the computations. In the wake region, the BART experiment over-predicts the pressure coefficient. The other results are very close to each other. It however appears that the predicted C_p in the wake region is not as high as it should be which has

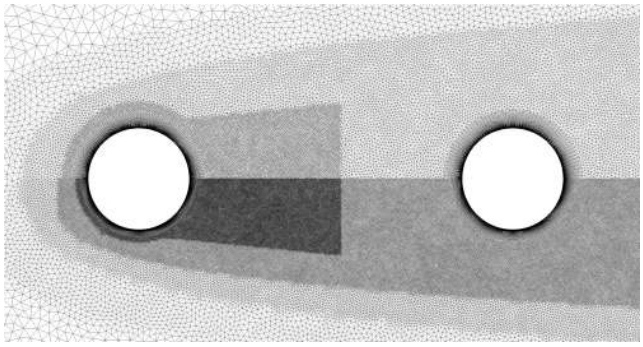


Figure F.2: *Top : Illustration of the different refinement zones for mesh M3; bottom : Zoom on the region close to the cylinders. Comparison between mesh M2 (top) and mesh M3 (bottom).*

a tremendous impact on the drag coefficient as will be seen later. For the downstream cylinder, the agreement is quite good between all the curves. It is however worth to note that the BART experiment exhibits also a behaviour that is different from that of the other results, mainly in the stagnation region. The RMS value of the surface pressure fluctuation is defined as:

$$C_p^{RMS} = \frac{\sqrt{\overline{p'^2}}}{\frac{1}{2}\rho U^2} \tag{F.2}$$

with $\overline{p'^2}$ the time average of the squared pressure fluctuations. This diagnostic is plotted in Fig. F.6.

One can observe, that for both cylinders, the shape of the curves is very similar. When $L_z = D$, the amplitude of the fluctuations is similar to that of the experiments. However it should not be the case as all the turbulent scales of the experiment cannot be captured. This similarity in amplitude happens therefore by chance as it is based on a different physical phenomenon. Indeed the flow topology is closer to what happens with a 2-D case where the amplitude of the shedding is maximal. For the two cases with $L_z = D$, the more refined mesh ($M3$) leads to more important pressure unsteady fluctuations as it captures more small scales. The case with $L_z = 4D$ leads to the lower fluctuations : this is due to the fact that the energy of the turbulent fluctuations are also distributed along the z direction. The 2-D effect is thus less stringent and the z averaged pressure fluctuations have a lower value. The downstream cylinder numerical results do not depart much from each other near $\theta = 0^o$

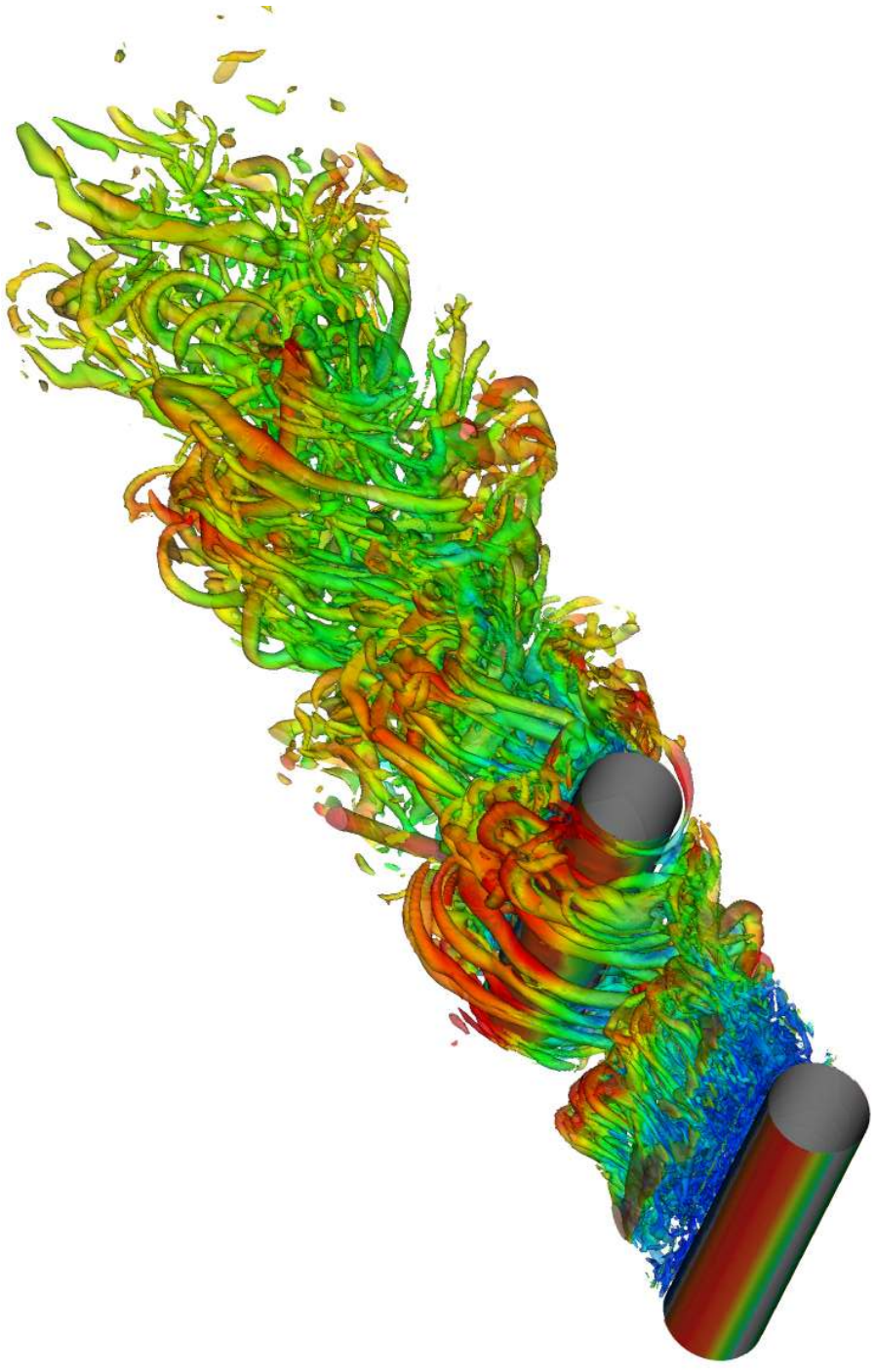


Figure F.3: *Velocity norm plot on $\lambda_2 = -10$ iso-value surface (case $M1$).*

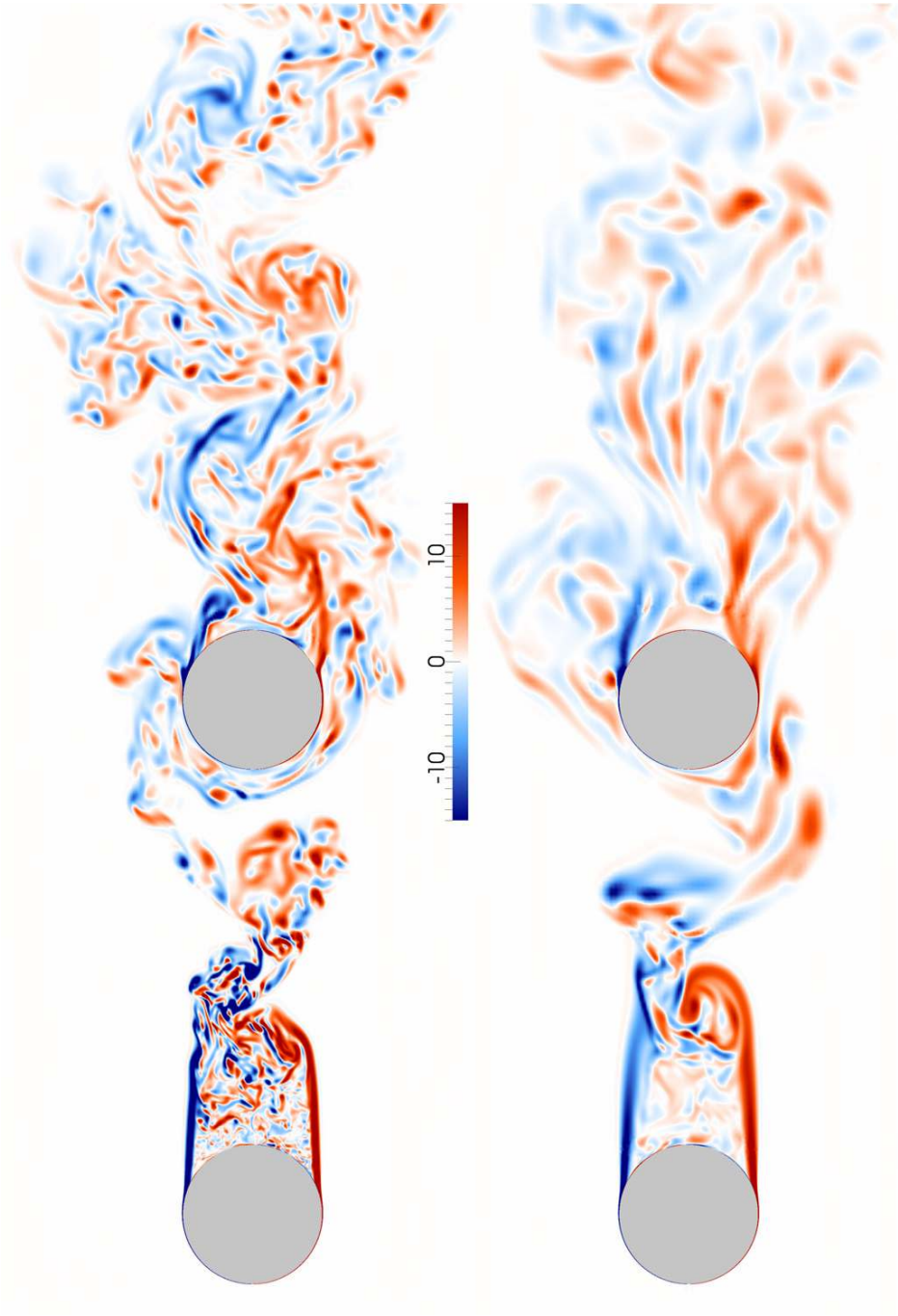


Figure F.4: *Snapshot of the vorticity $\frac{\omega_z D}{U}$ (with saturation of the color scale) in one plane perpendicular to z axis; case $M3$ (left), case $M1$ (right).*

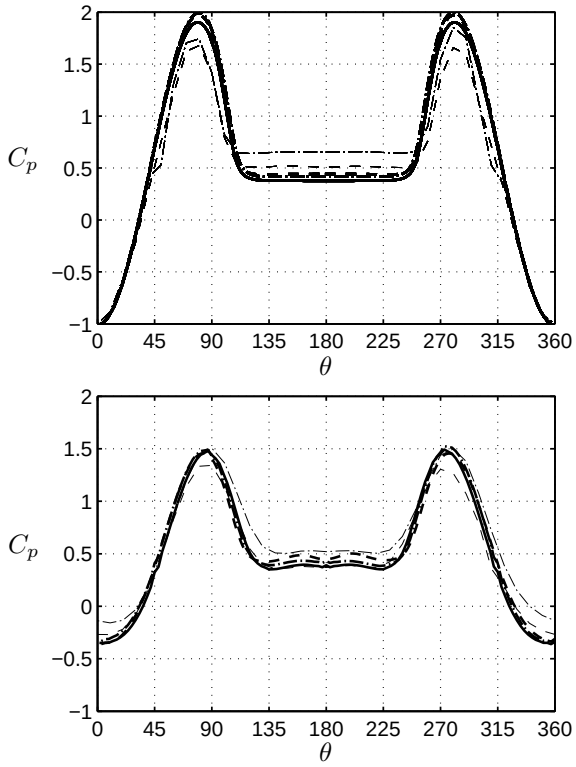


Figure F.5: *Top : Pressure coefficient azimuthal distribution on the upstream cylinder. BART experiment (dash-dot thin); QFF experiment (dash thin); case $M1$ (solid-thick); case $M2$ (dash-dot thick); case $M3$ (dash thick). Bottom : Pressure coefficient azimuthal distribution on the downstream cylinder. BART experiment (dash-dot thin); QFF experiment (dash thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick).*

because, there, the fluctuations mostly come from the upstream wake. On the contrary, behind the downstream cylinder, the 2-D effects play again an important role for the two cases with $L_z = D$ and the fluctuations are globally higher than those for $L_z = 4\,D$.

F.4.2 Velocity profiles

The streamwise velocity profiles are reported in Fig. F.7 for the region between the two cylinders (upper part). The amplitude of the velocity found in the case with $L_z = 4\,D$ is close to that of the experiment. However, the location of the peaks is quite different. This reflects a different size in the recirculation zone lying in the gap region, probably due to a separation point location

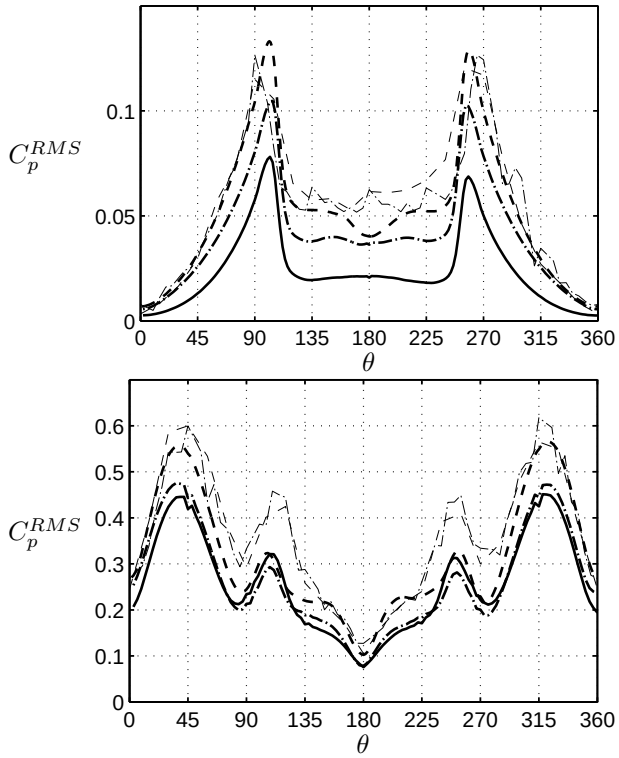


Figure F.6: *Top : RMS of the pressure coefficient azimuthal distribution on the upstream cylinder. BART experiment (dash-dot thin); QFF experiment (dash thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick). Bottom : RMS of the pressure coefficient azimuthal distribution on the downstream cylinder. BART experiment (dash-dot thin); QFF experiment (dash thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick).*

that differs from the experimental one. Nevertheless, the results for $L_z = D$ seem to have reached convergence. The velocity minimum is well captured in amplitude and position and so is the stagnation point. However, the velocity maximum peak location and amplitude does not follow the experiment. This repeatedly shows that the spanwise confinement is not negligible and, as a consequence, the good agreement between experience and computation concerning the stagnation point location comes about by chance. With this spanwise dimension, it can not capture properly the 3-D effects in the wake. Indeed, the confinement implies a larger velocity maximum as the spanwise dimension is not large enough to redistribute the kinetic energy, see also the 2-D URANS results from Doolan et al. [42]. This is enforced by the fact that, for $L_z = 4\,D$, even if the peak location is not correct, its amplitude is in good agreement

with the experiment showing that the flow topology is more likely driven by 3-D turbulent effects. In the region after the downstream cylinder (lower part of Fig. F.7), it can be observed that the case with $L_z = 4D$ is closer to the experiment than the cases with $L_z = D$. In this case, we think that using an even longer spanwise cylinder length should further improve the results.

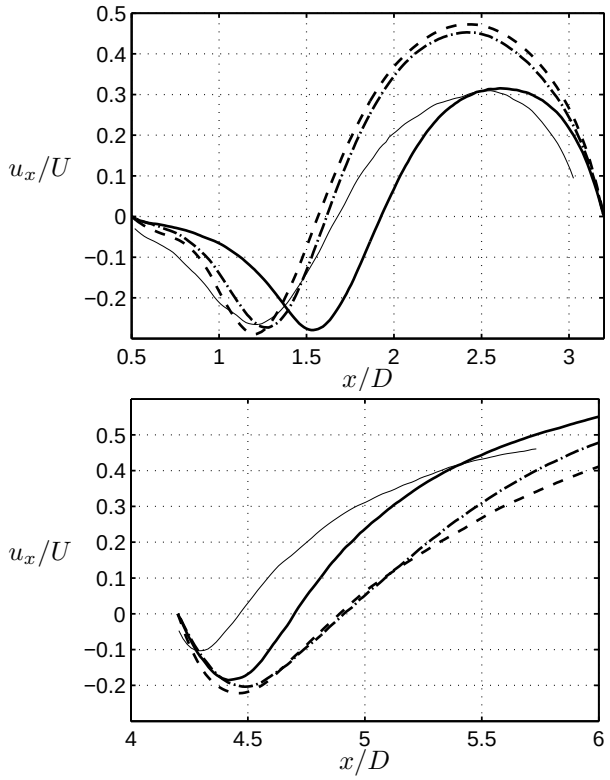


Figure F.7: *Top : Velocity profile in the gap between cylinders and along $y/D = 0$. BART experiment (solid-thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick). Bottom : Velocity profile after downstream cylinder and along $y/D = 0$. BART experiment (solid-thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick).*

F.4.3 Turbulent kinetic energy profiles

The 2-D turbulent kinetic energy is defined as:

$$\mathrm{TKE}_{2D} = \frac{1}{2}\left(\overline{u'_x u'_x} + \overline{u'_y u'_y}\right), \tag{F.3}$$

with $\overline{u'_x u'_x}$ and $\overline{u'_y u'_y}$ the time average of the squared velocity fluctuations respectively in the streamwise direction and in the direction perpendicular to the flow. This diagnostic is given along $y/D = 0$ in the upper part of Fig. F.8. The turbulent kinetic energy has similar properties as those of the pressure fluctuations. Indeed the numerical results show that the two cases with $L_z = D$ ($M2$ and $M3$) capture well the amplitude of the peak between the cylinders despite a slightly higher amplitude for the refined case $M3$. This, in turn, can be explained with similar arguments as those developed for the pressure fluctuations. The case $M1$ predicts a peak position which is in phase with the experiment. The amplitude is lower because it cannot capture all scales from the experiment. The amplitude is also lower compared with the two other cases as the spanwise confinement is more pronounced there. One can observe that the turbulent kinetic energy computed behind the second cylinder for the case $M1$ is the only one of the three cases to provide the correct tendency even if the predicted peak lies slightly behind the experiment. A comparison can be made with the streamwise velocity profile behind the downstream cylinder (lower part of Fig. F.7). Indeed there seems to be a link between the recirculation zone length and the position of the TKE_{2D} peak.

F.4.4 Temporal evolution of the lift and drag coefficients

The lift and drag coefficients are defined by

$$C_L = \frac{F_L}{\frac{1}{2}\rho\, U^2\, L_z\, D}, \quad C_D = \frac{F_D}{\frac{1}{2}\rho\, U^2\, L_z\, D}. \tag{F.4}$$

Their time evolution for the case with $L_z = 4D$ ($M1$) is reported in Fig. F.9 for the upstream cylinder and in Fig. F.10 for the downstream cylinder. These plots allows to check that the transient effects are well eliminated after 50 time units (defined by D/U). In the present work (case $M1$), the averaging process was carried for a time span of 135 convective times, starting at $150\frac{D}{U}$, which should be enough for statistics accumulation. The obtained time averaged values are $C_D = 0.33$ for the upstream cylinder and $C_D = 0.35$ for the downstream cylinder. This is indeed strange as one would expect the drag coefficient of the downstream cylinder to be lower than that of the upstream one. The experimental data of Jenkins et al. [63] provides $C_D = 0.49 - 0.52$ for the upstream cylinder and $C_D = 0.24 - 0.35$ for the downstream cylinder. The drag is still in the range of the experiments for the second cylinder while it

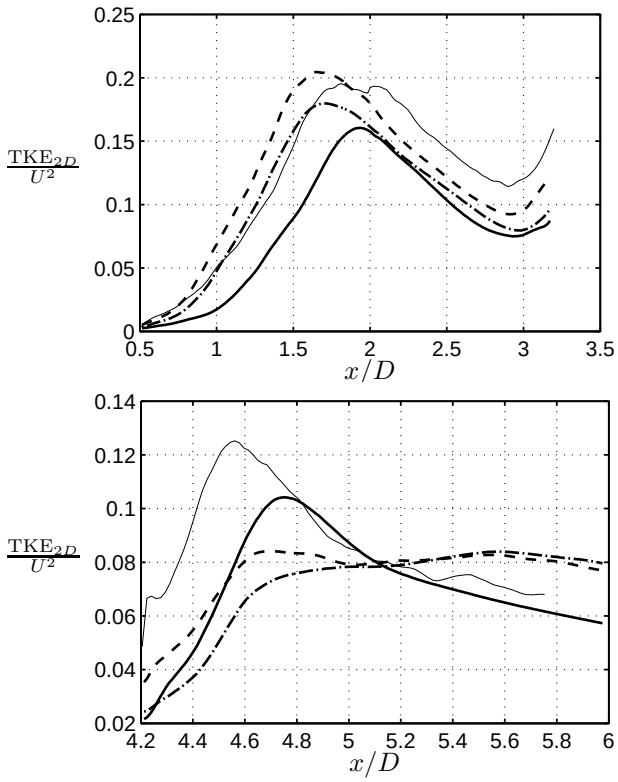


Figure F.8: *Top : 2-D turbulent kinetic energy profile (TKE) in the gap between cylinders and along $y/D = 0$. BART experiment (solid-thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick). Bottom : 2-D turbulent kinetic energy profile (TKE) after downstream cylinder and along $y/D = 0$. BART experiment (solid-thin); case $M1$ (solid thick); case $M2$ (dash-dot thick); case $M3$ (dash thick).*

is under predicted for the upstream cylinder. This is mainly due to the wake pressure behind the upstream cylinder which is not as low as in the experiment. This is closely linked to the separation point location, as will be seen in the next section.

F.4.5 Friction coefficient

The friction coefficient is defined as:

$$C_f = \frac{\tau_w}{\frac{1}{2}\rho U^2}\,. \tag{F.5}$$

The results are presented in Fig. F.11. As experimental results are not available, only numerical results are shown. The three cases are really close to each other

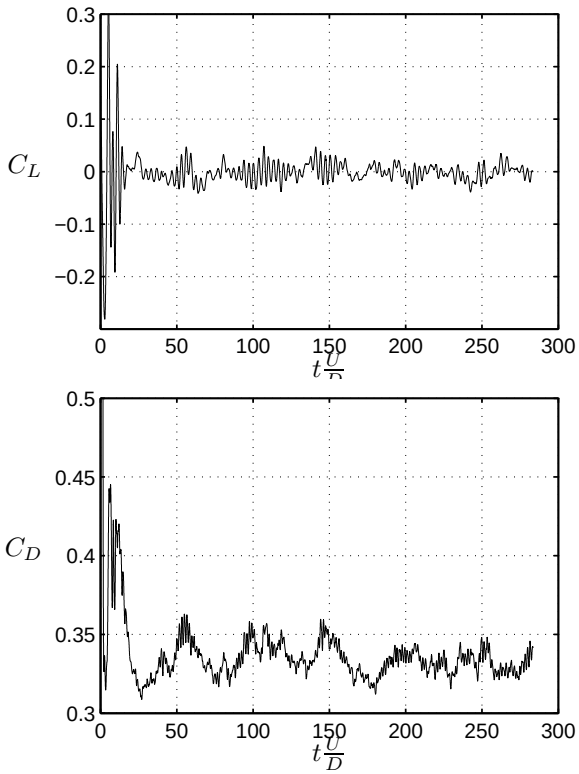


Figure F.9: *Top : Temporal evolution of the lift coefficient C_L for the upstream cylinder. Bottom : Temporal evolution of the drag coefficient C_D for the upstream cylinder.*

meaning that this diagnostic is quite independent of the spanwise dimension. By taking a closer look at the separation point on the upstream cylinder, defined as the root of the friction coefficient near $\theta = 105^o$ and $\theta = 255^o$, one can see that there is only a slight difference in its location for the three cases. Results are given in Table F.4. Independently of the spanwise dimension, the turbulent model predicts a separation location which is nearly the same for the three cases. This makes sense as the boundary layer is mostly homogenous in the spanwise direction and thus approximately two-dimensional.

It should also be observed that the friction coefficient is one order of magnitude lower than the pressure coefficient which means that its contribution to the drag is only minor, as expected for a bluff body flow. However, the separation point location directly influences the wake pressure. As can be seen in Fig. F.5, the region behind the separation point is quasi isobar. Therefore, the wake pressure is directly defined by the pressure at the separation point.

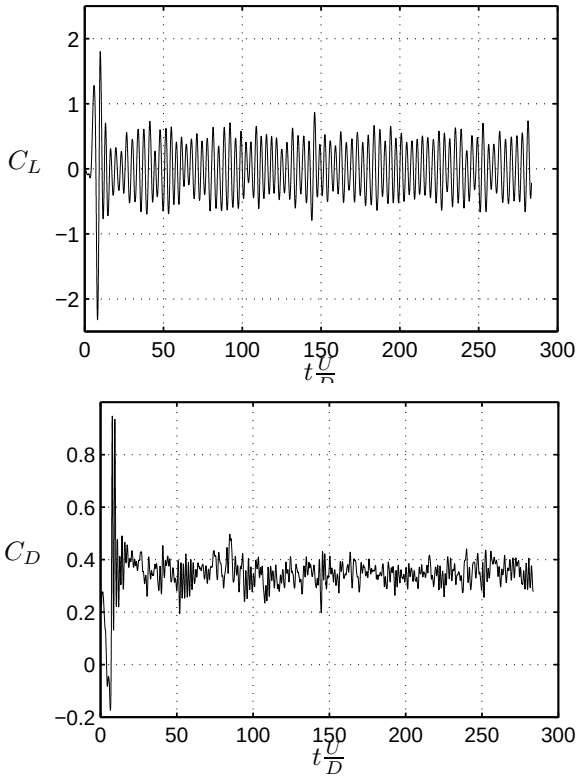


Figure F.10: *Top : Temporal evolution of the lift coefficient C_L for the downstream cylinder. Bottom :Temporal evolution of the drag coefficient C_D for the downstream cylinder.*

A more upstream location of the experimental separation point can be seen in the upper part of Fig. F.11. Different reasons can be invoked for this pressure difference: The turbulent model and the experimental tripping device. The turbulent model uses a tripping term which triggers a turbulent boundary layer. This relies on the assumption that the boundary layer is fully turbulent. On the contrary, the tripping device induces a turbulent transition of the boundary layer. The resulting transition length is probably not negligible and the boundary layer is probably not fully turbulent thus resulting in a earlier separation.

The mesh transition between the boundary layer mesh and the volume mesh can also influence the behavior of the turbulent model. Such a big difference between prediction and experiment shows that the flow is very sensitive to the boundary layer modeling since a slight modification of the input parameters has a large impact on the separation location and thus on global flow topology.

As a consequence, much care has to be taken while designing the mesh near the wall and choosing an appropriate turbulence model.

Mesh	$\theta\,[^o]$
$M1$	103.5
$M2$	104.9
$M3$	103.9

Table F.4: *Location of the main separation point on the upstream cylinder.*

F.4.6 Power spectral density

For the case $M1$, the power spectral density is taken from the wall pressure time signal at two different locations, the first at 135^o on the upstream cylinder and the second at 45^o on the downstream cylinder (see Fig. F.12). The main peak position of the upstream cylinder defines the shedding frequency. The computations predict a shedding frequency of $226\,Hz$ whereas, for the experiment, the resulting shedding frequency has a value of $178\,Hz$. This over-prediction is probably due to the over-prediction of the recirculation zone length that directly influences the shedding dynamics.

F.4.7 Mean flow streamlines

Fig. F.13 gives a comparison between the streamline configuration associated with each of the three cases. The two cases with $L_z = D$ ($M2$ and $M3$) have a similar flow topology except from the near wake of the upstream cylinder. Nevertheless, the lengths of the recirculation zone agree quite well. On the contrary, the case $M1$ ($L_z = 4\,D$) has a longer upstream recirculation zone and a shorter downstream recirculation zone. Indeed, if one remembers Fig. F.7, the upstream recirculation zone length is over-predicted for the case $M1$. This tendency can be related to the separation point position given in Table F.4. As the separation location is nearly the same for the three cases, only the spanwise dimension can be the determinant factor. For $L_z = D$, the 2-D effects imply a shorter recirculation bubble. It is also important to observe that the refinement implies a slightly shorter recirculation zone. It thus contributes to the correct tendency, as expected. For the same separation location, the 3-D effects force the case $M1$ to provide a larger recirculation zone. In brief, the combination of a

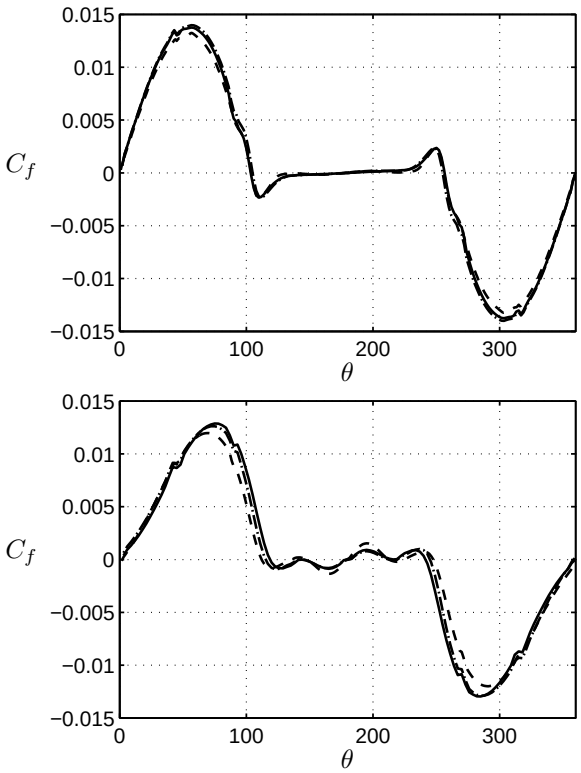


Figure F.11: *Top : Friction coefficient for the upstream cylinder. Case $M1$ (solid-thick); case $M2$ (dash-dot thick); case $M3$ (dash thick). Bottom :Friction coefficient for the downstream cylinder. Case $M1$ (solid-thick); case $M2$ (dash-dot thick); case $M3$ (dash thick).*

more refined mesh and a better capture of the separation point (more upstream location) would probably result in a shorter, and correct, recirculation length.

F.4.8 Mean flow vorticity and turbulent viscosity

The averaged spanwise component of the vorticity at the upstream cylinder is given in Fig. F.14 along with a comparison between the three cases. The represented colormap is saturated in order to allow better visualization of the separating shear layers. On the left hand side, the global aspect shows that the topology of the flow is slightly different if one compares the two cases with $L_z = D$ ($M2$ and $M3$) with the case with $L_z = 4\,D$ ($M1$). In the cases $M2$ and $M3$, the shear layers are slightly pointing downward, whereas the shear layers for the case $M1$ are nearly horizontal. Hence the recirculation region is longer

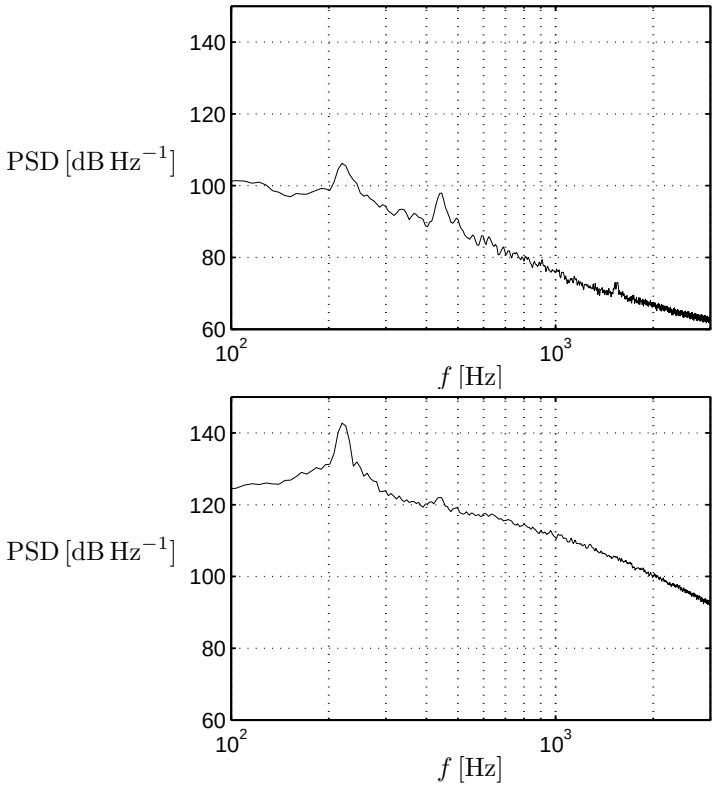


Figure F.12: *Top : Power spectral density of the surface pressure fluctuations taken at* 135^o *on the upstream cylinder. Bottom : Power spectral density of the surface pressure fluctuations taken at* 45^o *on the downstream cylinder.*

for $M1$ than for $M2$ and $M3$. This again comes from the confinement effect. There is also a slight difference in the thickness of the shear layer between $M2$ and $M3$ which can be observed on the right hand side of the Fig. F.14 (close-up on the upper shear layer of the upstream cylinder). As expected, the coarser the mesh is, the more the shear layer is diffused. Meanwhile, one could argue that this shear layer is still to thick, as no Kelvin-Helmhlotz instabilities are observed, even in the refined case $M3$, see Fig. F.4. Again, this is closely linked to the nature of the boundary layer. If it is fully turbulent at separation, the mean shear layer is too short and too thick to allow K-H instabilities (the most instable wavelength can be shown to be proportional to the thickness). However, if it is in a laminar or transitional state, the thickness will be reduced and K-H instabilities could more likely occur. In addition, as the shear stress is quite important in the shear layer, the turbulent viscosity production term

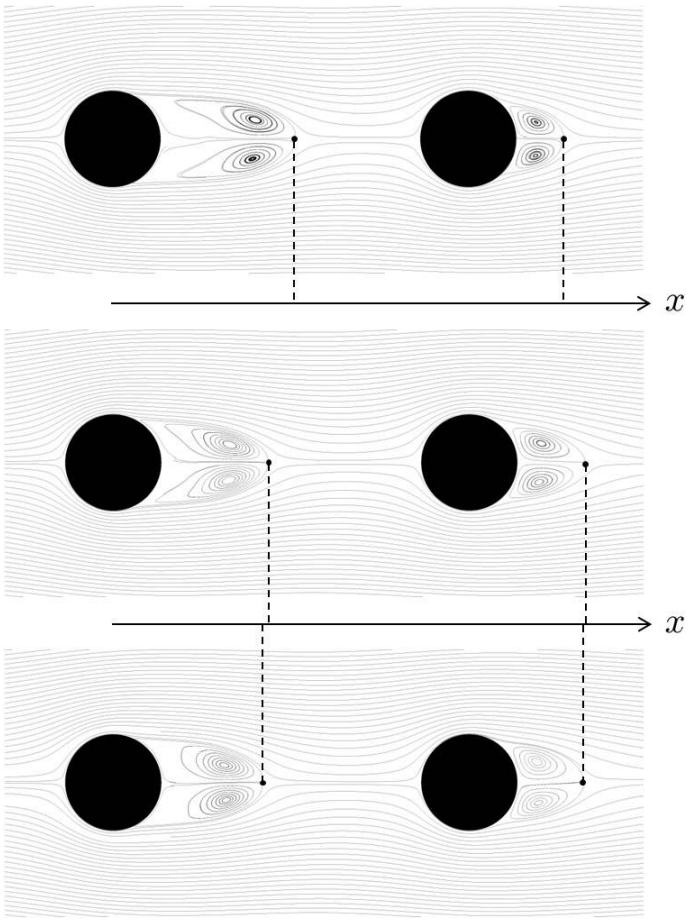


Figure F.13: *Mean flow Streamlines. The estimated stagnation point locations are indicated by a bullet. Top : case $M1$; Center : case $M2$; Bottom : case $M3$*

in the Spalart-Allmaras model [116] overwhelms the destruction term and local turbulence equilibrium is no more verified. The model behaviour lies between a RANS model and a LES approach. This also contributes to the thickening of the shear layer, see also Fig. F.15 for the turbulent viscosity near the separation. It is smaller for the refined case $M3$, as expected.

F.5 Computational resources

The computations were run on the Cenaero Ernest cluster. Each 8-slot nodes contains two quad-core *Intel Xeon* processors offering nearly 2 GB per core. The cores are connected with an Infiniband interconnect bringing the peak

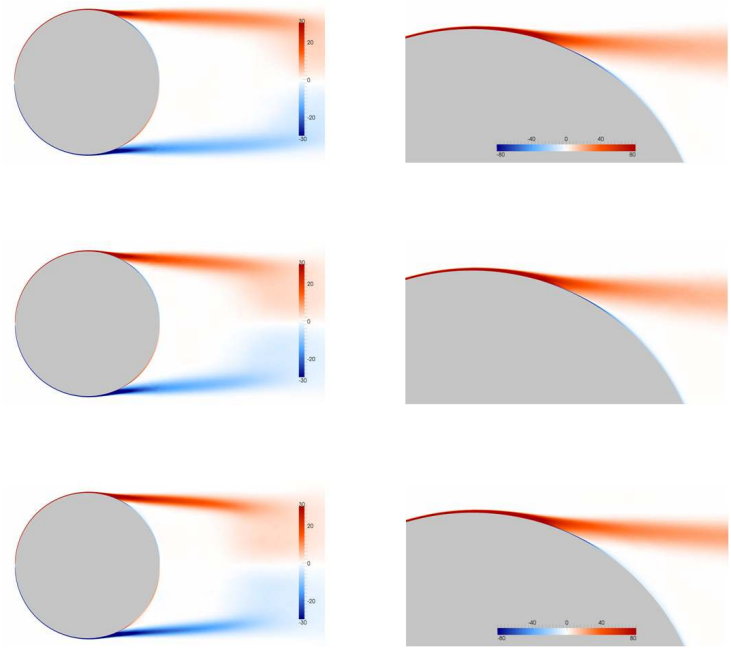


Figure F.14: *Time and spanwise averaged spanwise vorticity component for case $M1$ (top), $M2$ (middle), $M3$ (bottom) and zoomed view (right column).*

capacity of Cenaero cluster to 17 TFlops. The CPU times as well as the memory requirements are given in Table F.5. $T^{WC}_{\Delta t}$ is the wall clock time needed to perform a simulation of one time step, $T^{WC}_{1\,\mathrm{s}}$ is the wall clock time needed to perform a simulation of 1 s of physical time. max Mem is the maximal node memory load percentage (with $15.7\,GB$ per node).

Mesh	$T^{WC}_{\Delta t}$[s]	$T^{WC}_{1\,\mathrm{s}}$[days]	max Mem[%]
$M1$	38.9	45	48
$M2$	69.2	80	50
$M3$	75.2	174	52

Table F.5: *Computational ressources.*

F.6 Conclusions

The DDES of the turbulent flow past a pair of cylinders in tandem configuration and at high Reynolds number (1.66×10^5) has been performed. Three cases

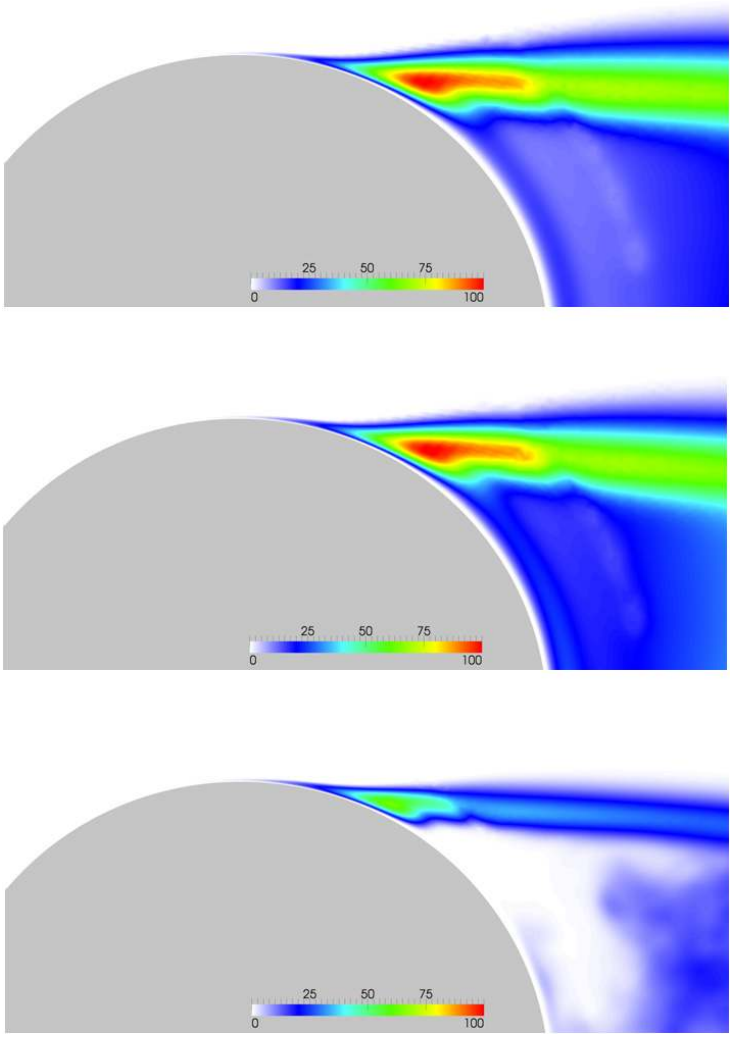


Figure F.15: *Normalized turbulent viscosity* $\frac{\nu_t}{\nu}$ *for case* $M1$ *(top),* $M2$ *(middle),* $M3$ *(bottom).*

were simulated to investigate the effect of the spanwise periodic length and the grid resolution. The most realistic case is indeed that with $L_z = 4\,D$ which is closer to the reference experiments. The two cases with $L_z = D$ are closer to a 2-D case and were run to study the influence of the grid refinement and of the confinement. Several quantitative diagnostics were provided and compared with the experiments. The agreement between the results and the reference experiment is quite fair. The averaged profiles are consistent with those of the experiment. However, the size of the recirculation zones are not the same as in the experiment. Moreover, the condition of the simulation and the experiment are not the same: the turbulence model induce a tripping on a boundary layer which is seen as fully turbulent (as in this zone it is a RANS behaviour) from the stagnation point while, in reality, this boundary layer experience a transition only after the location of the tripping device. This can explain why the location of the separation point of the upstream cylinder is further downstream of that of the experiment. This results in a too long recirculation region and in a too high back pressure (hence also a too drag coefficient).

Choosing a longer spanwise length for the cylinders should further improve the results. For instance, with $L_z = 8\,D$ one comes close to the length of the experimental setup ($L_z \geq 12\,D$) without having the important influence of the wind tunnel walls.