On the assessment of Immersed Boundary Methods for Fluid-Structure Interaction modelling: application to waste water pumps design and the inherent clogging issues

By

Mathieu Specklin

M.Sc, Diplôme d'Ingénieur

A dissertation submitted in fulfillment of the requirements for the award of the degree of

Doctor of Philosophy (Ph.D.)

Supervisor: Dr. Yan Delauré

School of Mechanical and Manufacturing Engineering

Dublin City University

May 2018

DECLARATION

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

ID No.: 14210488

Date:

I dedicate this work to my four grandparents, who were extraordinary people in extraordinary times...

To my grandfather Paul Specklin, historian, textile engineer and head of the R & D in Dollfus & Noack, who handed down to me curiosity and passion for science and history.

To my grandmother Blandine Specklin, née Martin, happy educator, the most joyful and the sweetest woman on earth, who gave me the love of singing and dancing. She taught me tolerance and forgiveness.

To my grandfather Dr. Bernard Schwaederlé, beloved physician and mayor of his village, who passed me his love of nature and wide spaces.

To my grandmother Elisabeth Schwaederlé, née Nonenmacher, dedicated nurse, the kindest and bravest person I've ever met.

They were certainly not perfect as human being, but they were in their grandparenthood.

Acknowledgments

My first thanks are going to my supervisor Yan Delaure and to Stephen Marry who made this research possible. I am grateful to Ben Breen, Robbie Connolly, Mark Kehoe and Stefan Berten for having followed and supported my work with interest. In general, I would like to thank the whole Sulzer group for their warm welcome. Special thanks to Abdulaleem Albadawi for his good collaboration and support all along the PhD. I am also thanking my two examiners Harold Esmonde and Sofiane Khelladi for accepting to review my work. Finally, the financial support of the Irish Research Council (IRC) under its Employment Based Research Scholarship programme (Grant No. EBPPG/2013/63) is acknowledged.

A very special thank-you to my lovely Clemence for her continuous support and reassurance during my times of doubts, and God knows they were plentiful! I would like to thank the old friends who can be counted on. Thanks to the Georges and their wives. Thanks to Ewen & Jing. Thanks to la Grosse Team. Thanks to the friends from VKI. Thanks to the 2A from MMK. Please all stay as you are. I would like to also thank the new friends. I was delighted to find again my twin brother Thomas, please stay in the family this time. Thanks to Blue Bear my singing partner. Thanks to Chris for being Chris and sorry for all the smokes I asked you. Thanks to Maria for her constant motivation to party. Thanks to James for showing us the real Ireland. Thanks to Greg for being so patient and open-minded with us. Thanks to the super-nice Greeks Dimitra, Stratos and Kiki, and thanks to the super-nice Germans, Nico and Morten. Thanks to Lucia & Max, Julie, Apo, Mathieu, etc, for bringing some additional fun in

Dublin. Thanks to the Brazilian waitresses from the DCU sport center. Thanks to all of you again for your constant positivism. The list is not exhaustive, and I am thanking all the new faces met in Dublin, with whom I have shared some fun. In general, I am grateful to Ireland and its amazing scenery, to the Irish sense of humour and to the Irish sense of welcoming. Finally, I thank my whole family (and God knows they are plentiful as well), and especially those who came visiting: Nico (at least), Dom, my parents, Philippe, Edith, Florie, David, Fleur & Thomas, and Thierry. And a special thank-you to Mamie, you are the best!

Because a PhD is not only made of joy and pleasure, but also of suffering, now is time for blames: firstly, as it appears that I was the last one among my friends to graduate, I would like to "thank" all of you for your "it's normal", "you will see" and "you'll be fine". I blame Dublin Bus and the clamps. I blame the Irish weather. I blame the Irish generosity which makes impossible to have only a few drinks. I blame Clemence for forcing me to go to London. I blame Thomas for making me running, I blame Blue Bear for not sharing his food, I blame Dimitra for choosing Chris' behind, and I blame Chris for having slept so many times on our couch. I blame Maria for her constant motivation to party. I blame James for his Mars bars and his lifts. I blame Greg for not showing up at half of the parties. I blame my family for not having a clue about what a PhD really is. A special dedication goes to my father who invited me for skiing the week before the viva. No blames for Mamie, you're still the best!

Contents

List of Figures

List of Tables

List of Abbreviations

Acronyms

AMI	Arbitrary Mesh Interface
BEP	Best Efficiency Point
DES	Detached Eddy Simulation
Fr	Froude number
FSI	Fluid-Structure Interaction
GGI	Generalized Grid Interface
IB(M)	Immersed Boundary (Method)
LB(M)	Lattice-Boltzmann (Method)
LB-IBM	Immersed Boundary Method coupled to a Lattice-Boltzmann solver
LES	Large Eddies Simulation
NS	Navier-Stokes
NS-IBM	Immersed Boundary Method coupled to a Navier-Stokes solver
pSA	penalized Spalart-Allmaras model (or equation)
RANS	Reynolds Average Navier-Stokes
Re	Reynolds number
SA	Spalart-Allmaras model (or equation)
VOF	Volume of Fluid

Abstract

On the assessment of Immersed Boundary Methods for Fluid-Structure Interaction modelling: application to waste water pumps design and the inherent clogging issues

By

Mathieu Specklin

The meshing stage of a Computation Fluid Dynamics (CFD) problem is of crucial importance. In realistic engineering applications, issues arise when dealing with complex, sharp and moving boundaries, removing the possibility of automatic creation of a high quality structured mesh for instance. For Fluid-Structure Interaction (FSI) problems, the standard body-fitted meshing approaches are also limited to low structure deformation and to simple geometries. In light of these limitations, Immersed Boundary Methods (IBMs) have shown to be good alternatives for a broad range of problems.

The present work sets out to build a set of numerical methods based on IBMs to simulate both the motion of rigid bodies and the transport of thin flexible solids. This research focuses on the specific area of waste water pumps. Firstly, IBMs are used to provide estimates of the hydrodynamic performances of centrifugal pumps. Secondly, this type of method is used to give a first answer regarding the characterization of the physical mechanism that leads to clogging in such pumps.

The numerical tool is coupled to two different solvers for the fluid equations: (i) Navier-Stokes (NS) and (ii) Lattice-Boltzmann (LB). In the NS context, a sharp IBM based on a penalization method is developed and implemented in the open source library OpenFOAM® in order to model the flow around rigid bodies. The complete model includes correction of the boundary conditions at the fluid-solid interface to improve the accuracy of the solution and coupling with turbulence models. To model the transport and the deformation of flexible structures, a diffuse penalty based IBM coupled to a solid model based on the variational derivative of the deformation energy is considered. In the LB context, the diffuse IBM available in the open source library Palabos® is assessed for one-way coupling problems with rigid bodies. The latter is extended and coupled to the same solid model as above in order to study flexible structures.

The capabilities and the accuracy of the two IBMs are assessed and compared with several test cases dealing with rigid bodies. For the sharp NS-IBM, numerical results of academic cases highlight the benefits brought by the corrections of the interface boundary conditions. In engineering cases, the method leads to results in good agreement with experimental data and numerical data from standard body-fitted simulations. Finally, for the IBMs aimed at modelling the flexible structures, both physical approaches compare well with previous numerical models in literature, and are giving promising results regarding the clogging mechanism.

Chapter 1

Introduction

1.1 Background

Waste Water Treatment Plants (WWTP) accounted for approximately 1% of the worlds total energy consumption in 2006, with increases predicted to exceed 20% by 2020 [139]. Since pump and blower motors can consume more than 80% of a WWTPs energy needs, there are still to this day important gains to be made by improving hydraulic efficiency and research and development continues to play a significant role in pump design. A typical design process for waste water pumps is based on a recursive loop (as illustrated in Figure 1.1). The process covers four separate design objectives: flow performances, machining constraints, mechanical constraints and solids handling. Given that a small change in any one of the stages can severely impact the three others, the general cost and time of such a design process is substantial.

Pump clogging although not directly linked to hydraulic efficiency has become an equally important aspect of sewage pump design due to the increased occurrence of certain resilient waste products. In the city of Sydney alone, 500 tonnes of wet wipes are removed from the sewers each year [127] (see Figure 1.2). Australia's government is spending up to $25 million to clear sewage pipes each year [26]. The city of New York has spent more than $18 million between 2010 and 2015 on clogging problems [94]. The materials involved in the clogging are generally flushed wet wipes, which compose

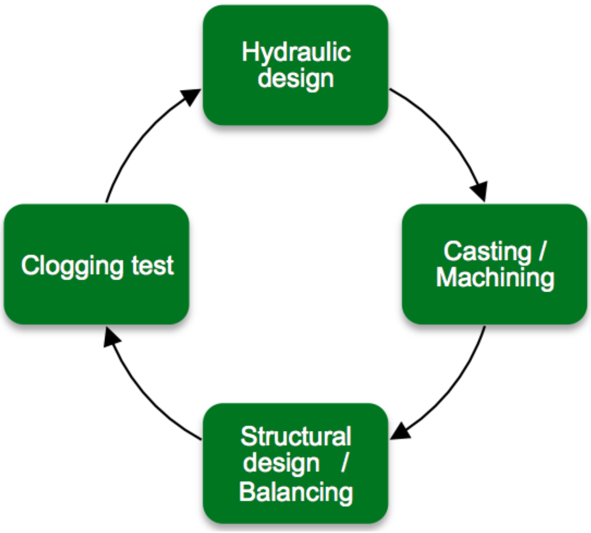


Figure 1.1: Design process of a waste water pump.

75% of the sewer blockages [127]. The blockage of a waste water centrifugal pump is illustrated in Figure 1.3. The rate of accumulation of rags and fibrous clumps is known to depend on the flow conditions and certain hydrodynamic properties of the pumps. As it is difficult to characterize and quantify experimentally the mechanisms that lead to clogging, a statistical approach is generally used, by releasing hundreds of rag or textiles in a test tank to extract a relevant ratio passed/blocked.

Figure 1.2: 500 tonnes of wet wipes are removed from Sydney sewers each year [127]. That is the equivalent to 4 blue whales.

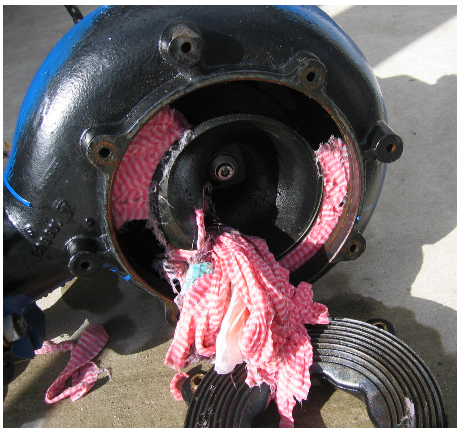


Figure 1.3: Waste water pump blocked by textiles after test experiments. Photograph provided by Sulzer.

Numerical simulation is a promising tool, which could accelerate the design processes and overcome the issues inherent to experimental testing (cost and time mainly), if it was possible to model the process involved in rag flow and pump interactions. The research presented in this work has been built as a collaboration with Sulzer Pump Solutions Ireland Ltd. with a dual objective: (i) to develop innovative computational models in order to improve the design cycle of a waste water pump and (ii) to understand the different mechanisms that can lead to a clogging scenario.

The optimal hydrodynamic design for a waste water pump is a good compromise between hydraulic efficiency and anti-clogging performances. Although the hydrodynamics of pumps is already well known [48] [15], the effectiveness of more subtle pump design changes are difficult to assess without accurate computational tools capable of capturing both solid and liquid phases. One of the major challenges for accurate Computational Fluid Dynamics (CFD) modelling lies in the creation of adequate computational meshes in realistic engineering configurations. Standard body-conforming models meet their limits when faced with the type of Fluid-Structure Interaction (FSI) problems considered here. Issues arise when dealing with complex, sharp and moving boundaries. For example it is not always possible to rely on automatic mesh generation tools to achieve high quality structured mesh for the Navier-Stokes framework. If the aim is to capture in addition the interaction with thin and flexible structures, the mesh morphing and re-meshing required in standard approaches are limited to moderate changes in shape and performed to the detriment of the computational time. In this context, Immersed Boundary Method (IBM) types of method are a promising alternative for this sort of problems. The aim of IBM is to take into account the presence of a secondary component in the fluid through a modification of the conservation equations, instead of fitting the grid to the secondary medium. One of the main interests of IBM lies in the drastic simplification of the model and the meshing stage, and especially in its suitability for use with Cartesian grids of perfect orthogonal quality.

1.2 Core of the research

1.2.1 Main objectives

The main aim of the research reported in this manuscript was to develop versatile numerical models capable of simulating the transport of slender flexible textiles and rags in centrifugal waste water pumps. The research focused on IBMs and involved two complementary objectives:

1. The development of IBM solutions for single-phase flows at low to high Reynolds number with moving rigid boundaries of arbitrary shape.
2. The development of a robust diffuse interface IBM for the study of the transport and the deformation of thin flexible solids immersed in fluids.

Two types of numerical solvers for the fluid equations, Navier-Stokes and Lattice-Boltzmann, were considered in this research. An IBM developed within the former is referred as NS-IBM, while an IBM developed in the latter is referred as LB-IBM.

The first objective listed above is subdivided into different tasks:

- the development of a sharp interface type of IBM in the open-source library OpenFOAM® (Navier-Stokes solver), which allows a better representation of arbitrary geometries. The IBM is based on a penalization (or Penalty) approach to impose the desired velocity at the immersed interface. It is a continuous IBM, as the momentum forcing is applied before the discretization of the equations.

- the development of correction schemes to improve the imposition of the boundary conditions at the immersed interface, when the grid is not fitting the solid geometry. These reconstruction schemes aim at increasing the order of accuracy of the sharp NS-IBM.

- the combination of the method with relevant turbulence models.

- the validation of the sharp NS-IBM for a wide range of FSI problems, involving fixed and moving boundaries, simple and complex geometries, laminar and turbulent flows. The validation involves assessments of the orders of accuracy, comparisons with experimental and numerical results from the literature, and comparisons with equivalent body-fitting simulations.

- the comparison of the sharp NS-IBM with the diffuse interface type of IBM developed by Inamuro [97], [125] for a Lattice-Boltzmann solver and available in the open-source library Palabos®. The two types of method (see Figure 1.4 for illustration) are assessed in terms of accuracy mainly, and also in terms of stability and computational cost.

The second objective is subdivided into the following tasks:

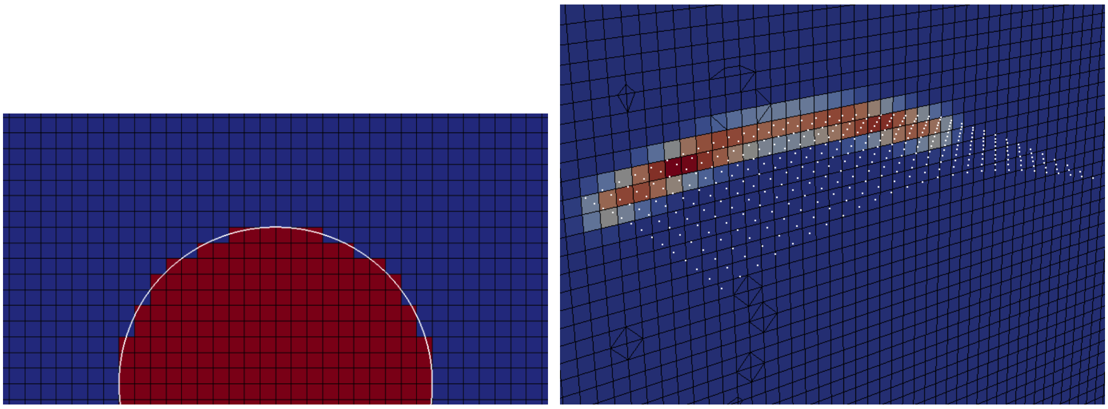
- the coupling of Inamuro's diffuse IBM in the Lattice-Boltzmann framework with a solid solver capable of modelling the deformation of a two-dimensional flexible structure.

- the validation of the resulting LB-IBM for a wide range of FSI problems, involving transport and deformation at low to high Reynolds number.

- the comparison between the diffuse LB-IBM with a diffuse NS-IBM based on a penalization method. The same solid solver is used for both approaches and the comparison focuses on the coupling mechanism which accounts for the action of the fluid on the thin flexible solid.

- the development and the assessment of solid-solid interaction models to handle collision between slender flexible structures and other rigid solids.

- the preliminary study of the physical processes leading to clogging.



(a) sharp interface IBM

(b) diffuse interface IBM

Figure 1.4: Representation of a sharp IBM and a diffuse IBM. In the former case, an Eulerian fluid cell is either fully in the fluid (blue) and is not influenced by the presence of the solid, or fully in the solid (red). In the latter case, the influence of the solid (represented by the white dots) is spread in the fluid domain in the vicinity of the solid points.

1.2.2 Framework

The framework of the research project is summarized in Figure 1.5. At the end of the project, two computational methods fully based on IBM have been delivered. The two methods built respectively in Navier-Stokes and Lattice-Botlzmann solvers are able to accurately simulate Fluid-Structure Interaction problems, both with rigid solids of complex shape and with flexible slender structures. This research is the main contribution to the project.

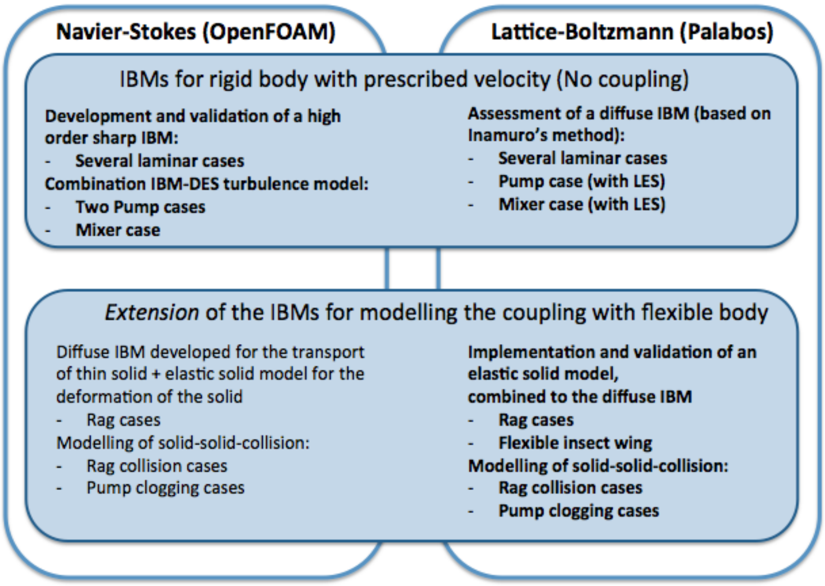

Navier-Stokes (OpenFOAM)
Lattice-Boltzmann (Palabos)
IBMs for rigid body with prescribed velocity (No coupling)
Development and validation of a high order sharp IBM:
- Several laminar cases
Combination IBM-DES turbulence model:
- Two Pump cases
- Mixer case
Assessment of a diffuse IBM (based on Inamuro's method):
- Several laminar cases
- Pump case (with LES)
- Mixer case (with LES)
Extension of the IBMs for modelling the coupling with flexible body
Diffuse IBM developed for the transport of thin solid + elastic solid model for the deformation of the solid
- Rag cases
Modelling of solid-solid-collision:
- Rag collision cases
- Pump clogging cases
Implementation and validation of an elastic solid model, combined to the diffuse IBM
- Rag cases
- Flexible insect wing
Modelling of solid-solid-collision:
- Rag collision cases
- Pump clogging cases

Figure 1.5: Summary and framework of the research project. The bold part represents what has been done during this research. The standard part represents what has been done by another member of the research team.

1.2.3 Novelties

The novelty of this research lies in different aspects of the work, which are highlighted below:

- The novelty from the sharp NS-IBM is in the formulation of the model. The method considers a correction for the pressure at the immersed interface in order to eliminate the influence of the pressure from the interior penalized solid domain. The pressure is corrected near the interface in the spirit of a penalization technique. Implicit and explicit schemes are investigated. Pressure treatments have not been considered before for continuous penalty types of IBM and results presented in the thesis confirm that improvements are achievable by correcting the pressure at the immersed interface. A new versatile reconstruction method for the velocity has also been developed and validated for fixed and moving boundaries. For turbulent cases, a reconstruction based on a power-law has been coupled to a hybrid RANS-LES model, which is based on a Reynolds Averaged Navier-Stokes (RANS) formulation of the Reynolds stresses near the immersed wall, and a Large Eddy Simulation (LES) model outside the boundary layer region. The simulation sensitivity to the power law has been tested for a cylindrical mixing tank at $Re = 5.18 \cdot 10^{-4}$. For the RANS part of the hybrid turbulence model, a Spalart-Allmaras (SA) model is used. A versatile formulation of the SA equation is proposed with a penalization of the turbulent viscosity at the immersed wall. This penalty Spalart-Allmaras (pSA) model can be used in a full RANS turbulence model or with a hybrid DES turbulence model. The novelty of this part of the research lies as well in the application of the model. Apart from recent work of Posa et al. [105], [104], IBM have not previously been applied to flows in high speed rotating machinery. Finally, one can also note that it is the first time that a sharp IBM with corrections accounting for the rasterization effect has been implemented in the OpenFOAM® library.

- The diffuse LB-IBM used to model the interactions between thin flexible structures and fluids is based on the Multi Direct Forcing Method (MDFM) developed by Wang et al. [147] initially in the Navier-Stokes context, and later adapted to the Lattice-Boltzmann framework by Inamuro et al. [97], [125]. In this part of the research, the main novelty lies in the combination of the method with an elastic model for two-dimensional membranes. Secondly, some first advancements have been made on the interactions between rigid and flexible solids. Except for the collision method introduced by Huang et al. [55] for one-dimensional filaments, solid-solid collision have never either been investigated nor modelled in this context. Finally, the application of the two diffuse IBMs considered in this research for the transport and the deformation of 2D flexible solids also represents a first attempt at testing the suitability of the coupling approaches to this type of problems. Similar applications includes the recent preliminary work by Jensen et al. [63], [62] towards the simulation of clogging effects in waste water pumps, which uses a Discrete Element Method (DEM) coupled to standard CFD to handle this type of problems. Also to be noted is the work of Akcabay [3], which presents an

IBM developed to model the behaviour of cloth in washing machine. However no attention has been paid on solid-solid collision within this work, and unphysical penetration might occur with coarse grids.

- This research proposes also a versatile comparison of different IBMs. Firstly, a comparison is made between sharp and diffuse interface types of method for the study of single-phase FSI problems. Secondly, within the context of diffuse IBM for the transport of flexible solids in fluids, a comparison has been carried out between two different types of coupling mechanism: (i) a penalty IBM within a Navier-Stokes solver, and a direct-forcing IBM within a Lattice-Boltzmann solver.

1.3 Thesis outline

The body of the thesis is organised as follow:

- Chapter 2 presents an overview of the Immersed Boundary Methods. The first section of the chapter reviews the existing numerical techniques for interface problems in CFD, including IBM, while the following sections review the literature on IBM in both Navier-Stokes and Lattice-Boltzmann frameworks.

- Chapter 3 presents the sharp IBM developed and implemented in OpenFOAM® for modelling rigid moving boundaries in fluids. Particular attention is paid to the corrections of the physical variables at the immersed interface to improve the accuracy of the method. The IBM, including all the different correction schemes, is referred as xIBM.

- In Chapter 4, the previously presented sharp IBM is validated against laminar cases and common benchmarks from the literature at low Reynolds number.

- In Chapter 5, the capabilities of the method is assessed with realistic engineering cases in turbulent mixing flows.

- Chapter 6 proposes a comparison between the sharp IBM and the diffuse IBM implemented in the open-source Lattice-Boltzmann library (Palabos®). For this purpose, the diffuse IBM developed by Inamuro is firstly introduced.

- Chapter 7 presents the development and the validation of a numerical model based on the previous diffuse IBM in the Lattice-Boltzmann solver for the simulation of the interactions between a fluid and thin flexible structures. A diffuse IBM already implemented in OpenFOAM® for similar applications is used for comparison purposes.

- Finally, chapter 8 is divided in two parts. The first section is an attempt to answer the clogging issues which can be encountered in waste water pumps. The second section highlights the main conclusions and contributions of the present research. Recommendations for future works are also included.

Chapter 2

Overview of the Immersed Boundary Methods and their applications

2.1 Boundary treatments and other methods for interface problems in CFD

One of the pillars of standard CFD techniques is the discretization of space. This discretization process needs to fit all the boundaries of the domain where the fluid equations are solved. The meshing stage can become highly challenging when dealing with complex boundaries for Navier-Stokes (NS) solvers, or with moving boundaries for Lattice-Boltzmann (LB) solvers for instance. Significant issues arise as well when the grid needs to be adapted to a secondary component transported in the fluid. In this context, versatile CFD methods have been developed over the past two decades for both physical approaches in order to overcome these difficulties.

2.1.1 In the Navier-Stokes context

The problem of a wall or interface immersed in a fluid can be written in the following way for an incompressible flow studied in the context of the Navier-Stokes equation:

$$\begin{cases} \rho\frac{\partial \mathbf{u}}{\partial t} + \rho\nabla\cdot(\mathbf{u}\mathbf{u}) - \mu\nabla^2\mathbf{u} = -\nabla p & \text{in} \quad \Omega_f \\ \nabla\cdot\mathbf{u} = 0 & \text{in} \quad \Omega_f \\ \mathbf{u} = \mathbf{u_{ib}} & \text{on} \quad \Gamma \end{cases} \tag{2.1}$$

where Ω_f represents the fluid domain, as opposed to Ω_s which is the immersed body domain, and Γ stands for the interface between both, as illustrated in Figure 2.1. The system of equations in 2.1 is given for a no-slip boundary condition at the interface, which is the most common physical condition found in FSI problems, but other types of condition can be considered. The Navier-Stokes solver must satisfy the governing equation in the flow domain, and the correct boundary conditions at the interface Γ. This can be done by adapting the mesh to the interface geometry or by modifying the fluid equations in its vicinity to recover the desired boundary conditions. Different methods have been developed to satisfy these constraints. They can be classified in four main families:

- Standard body-fitted methods
- Immersed Boundary Methods (IBM) and Immersed Interface Methods (IIM)
- Cut-cell methods
- Overset methods

Standard body-fitted approaches

In standard techniques, the mesh is fitted to the geometry as illustrated in Figure 2.2 for a Finite Volume discretization of a turbomachine. In realistic engineering applications, achieving a body-fitted mesh of high quality can be particularly troublesome when a lack

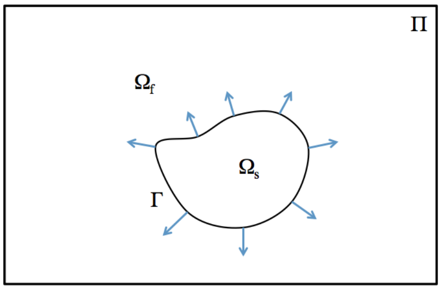


Figure 2.1: Sketch of the different domains

of periodicity, symmetry and/or clear block structure makes automated structured grid mesh generation by specialized pre-processors very difficult or simply unavailable. One solution in this case is to rely on hybrid meshing to combine regular hexahedral cells where possible, prism layers near solid surfaces where boundary layer must be accurately measured and other polyhedral cells to facilitate transition at interfaces. It can often be difficult however to avoid distorted mesh with associated numerical instabilities and loss of accuracy. This is particularly the case when strong gradients in mesh sizes cannot be avoided due to large cell count and high refinements near walls in high Reynolds flow. Also it should be noted that although polyhedral mesh blocks provide useful flexibility when hexahedral cells are used in the core of the flow domain and can greatly simplify meshing, the resulting hybrid mesh structure can be a problem in turbulence modelling with Large Eddy Simulation (LES) or Hybrid LES and Reynolds Averaged Navier Stokes (RANS) simulations, where the eddy viscosity model typically relies on a coefficient whose calibration is related to the mesh type and size.

In the context of moving boundaries, the grid must be adapted at every time step, which increases drastically the computational time. However, for certain types of motion, more specific methods have been developed in order to reduce the mesh morphing and the re-meshing. Hence for simple translation or rotation of a component, sliding mesh methods can be used. With this technique, two different mesh parts are defined, one for the fixed background, and one for the moving zone which fits the geometry of a rotating component for instance. All the non-conformal cells are then adapted at each time-step

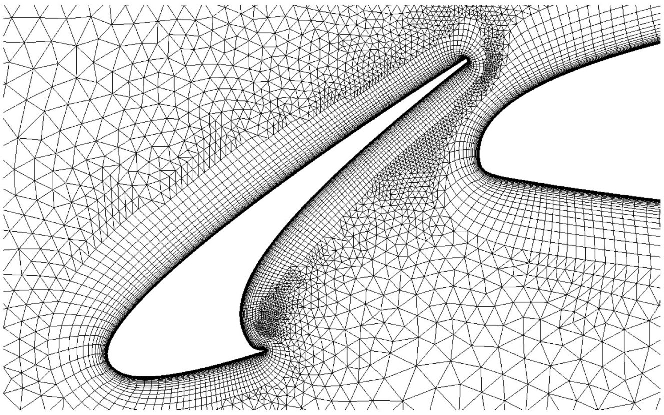


Figure 2.2: Sketch of a body-fitted Finite Volume mesh used to resolve the flow inside a turbomachine.

at the interface between the two mesh zones. Novel sliding mesh methods have been aimed at removing this requirement to adjust the topology of the mesh at the interface. This is case of the Generalised Grid Interface (GGI) method, which evaluates instead a set of weighting factors in order to properly balance the flux at the GGI interface.

For Fluid-Structure Interaction (FSI) problems, the Arbitrary Lagrangian Eulerian (ALE) method is widely used. With this approach, the fluid equations are also solved on a deforming grid, which follows the movement of the fluid-structure interface. However ALE are cumbersome in their formulation and almost impossible to apply to FSI problems with large deformations [119].

Immersed Boundary Methods (IBM)

The aim of IBMs is to solve a single set of conservation equations for several components (gas, liquids or solids). In IBMs, the presence of the secondary body is taken into account by the addition of a forcing term in the momentum equation introduced in Equation 2.1. This forcing term is added to the fluid equations in order to reproduce the boundary conditions at the interface between the fluid and the immersed surface, without having to generate a mesh which conforms to the latter.

This type of method can handle multi-phase flow and fluid-structure interaction problems in a single-phase framework. The main interests lie thus in the simplification of the model, and in its suitability for use with Cartesian mesh of perfect orthogonal quality for Navier-Stokes solvers. This strongly reduces the time required to setup the computational case, which can be significant with complex and moving geometries. On such simple mesh however, the spatial discretization of the problem does not generally correspond to the geometry of the components. Special care may then have to be taken at the interfaces between the media, in order to reproduce the effects of the real geometry. Figure 2.3 illustrates the difference between the two types of mesh in the case of a flow past a cylindrical obstacle, for both an IBM and a standard body-conforming approach.

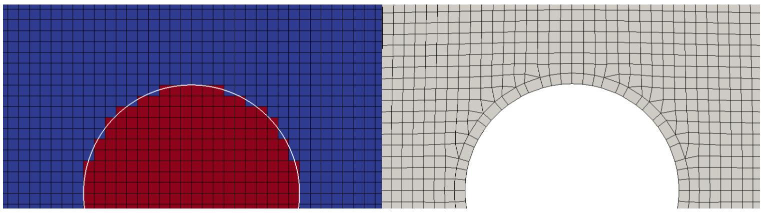


Figure 2.3: Sketch of the grids used to resolve the flow around a cylindrical obstacle. (Left) Cartesian grid used for an IBM, distinguishing both fluid (in blue) and solid (in red) domains. The white line represents the interface of the immersed solid. (Right) Equivalent grid in a body-conforming approach

With the definition above, all the IBM can also be classified under the Fictitious Domain Methods (FDM). This group consists of solution techniques for which the fluid equations for a geometrically complex domain are solved in a simpler (fictitious) domain embedding the original one, with the boundary conditions of the latter enforced in the new domain [51].

Immersed Interface Methods (IIM)

The Immersed Interface Method shares the same mathematical formulation as the Immersed Boundary Method, by representing the boundary immersed in a fluid as a singular force. However the IIM, first introduced by Li [84], is used when the solution of the fluid equations is of interest on both side of the interface. The problem is not to solve the Navier-Stokes equations in Ω_f only, as introduced in 2.1, but in Ω_s as well with jump conditions at the interface. In the IIM, these conditions are directly incorporated in the discretization schemes, which makes the method comparable to the discrete forcing type of IBM (see Section 2.2.1). This discrete formulation of the IIM can be adapted to achieve any desired order of accuracy.

The cut-cell methods

The cut-cell methods are a hybrid type of methods. Similarly to IBM, they are applied to uniform Cartesian grids. However, in cut-cell methods, the cells cut by the interface are reshaped according to the local geometry of the immersed wall. This process, shown in Figure 2.4, generates at the end an unstructured body-fitted grid, with triangular, trapezoidal or pentagonal cells of arbitrary size near the interface.

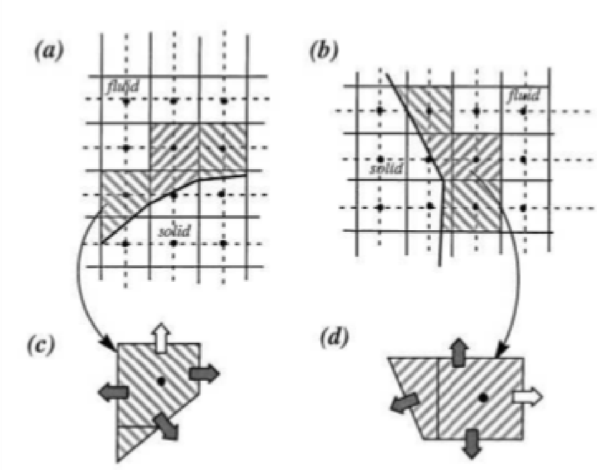


Figure 2.4: Sketch of the cutting process of the cut-cell method proposed by Ye et al. [158] and associated new face fluxes. Figure is taken from [158].

A major advantage of the cut-cells methods is that, inherently to the Finite Volumes Method, the local conservation of mass and momentum is guaranteed by resorting to a body-fitted approach. However realistic engineering applications with complex body shapes make the cutting process difficult especially in 3D problems. Furthermore the large change of size between neighboring cells tends to destabilize computations, while the presence of tiny cut cells adds to the requirement of very small time step. To tackle these issues, particular cut-cell methods have been developed to manage the topology of the cut cells, as for instance the merging of the smallest cells [158].

Overset methods

The principle of the overset methods lies in the combination of geometrically simple overlapping grids. A grid is generally made to fit each component of a system's geometry. In the overlapping regions, boundary information is exchanged between the grids via interpolation of the flow variables. The settings of an overset method is thus decomposed into two steps: (i) the grid generation for each components and (ii) the determination of interpolation weights for the transfer of data between grids. These two steps may be necessary at each time step if dealing with moving physical boundaries. In Figure 2.5, an overset mesh is illustrated as a combination of two grids. A perfect Cartesian grid is used for the background fluid, while a curvilinear grid has been generated around the flexible surface. The overlapping zone is clearly visible on the figure. It is important to note also that the overlapping is not total over the domain, as in both grids newly useless cells are removed for computational efficiency. The main advantage of overset methods is double: (i) it simplifies drastically the mesh generation and (ii) avoid the stretching of the grid as well as the use of high skewed cells.

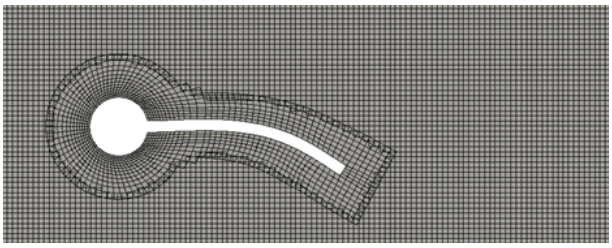


Figure 2.5: Sketch of the combination of two grids used for an overset approach.

2.1.2 In the Lattice-Boltzmann context

Although, the Lattice-Boltzmann method is also a discrete model for the dynamics of fluids, its formulation and assumptions differ from the Navier-Stokes approach. In the next section the basic principles of the method are introduced.

Basic definitions for the Lattice-Boltzmann method

The method follows on from the combination of cellular automata algorithms and the kinetic theory of gases. Cellular automata is a family of methods which investigates the behavior of an entity that occupies a specific position on a discrete space and that interacts with its identical neighbors. Regarding the physics, the kinetic theory of gases proposes a molecular description of the fluid, where the state of a molecule is defined through a position vector $\mathbf{x}$ and a velocity or momentum vector ξ. In order to be able to describe a great number of particles, statistical approach are used in which a system is represented by an ensemble of many copies. In this context, probability density functions (pdf) $f(\mathbf{x}, \xi, t)$ gives the probability of finding a particular molecule with a given position and momentum in a six-dimensional phase space.

The Boltzmann equation governs the motion of the molecules by expressing a balance between transport of these particles and collision between particles:

$$\frac{\partial f}{\partial t} + \xi_i \frac{\partial f}{\partial x_i} = C(f) \tag{2.2}$$

where the left hand side of Equation 2.2 stands for the transport term and the right hand side represents the collision term between particles. A very common model for collision is the single relaxation time BGK (Bhatnagar-Gross-Krook) approximation, where the collision term is modeled as a relaxation term towards equilibrium, and reads:

$$C(f) = -\frac{1}{\tau}(f - F) \tag{2.3}$$

The relaxation time τ in Equation 2.3 is related to the physical viscosity, while F

stands for the Maxwell-Boltzmann equilibrium distribution, which describes a particular probability distribution for a system of particles assumed to have reached a thermodynamic equilibrium.

Finally, the Lattice-Boltzmann models are merely discretized versions of the Boltzmann equation. The LBM reduces the number of possible particle spatial position and microscopic momenta to a finite amount. In this context, the space is discretized on a regular lattice grid, while a set of discrete velocities is defined to describe the momenta. For instance, the D2Q9 model illustrated in Figure 2.6 is a common Lattice-Boltzmann model for 2D problems, containing 9 discrete velocities. The particle distribution function introduced above is now as well a set of 9 variables f_α, representing frequencies of occurrence. The macroscopic variables can be recovered by the following equations:

$$\rho = \sum_\alpha f_\alpha \tag{2.4}$$

$$\rho \mathbf{u} = \sum_\alpha f_\alpha \mathbf{c}_\alpha \tag{2.5}$$

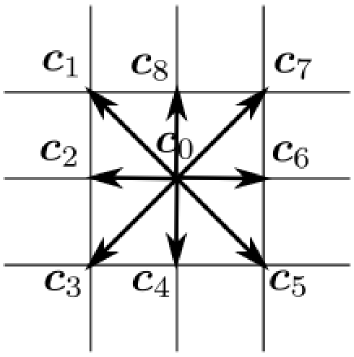


Figure 2.6: Sketch of the D2Q9 model with lattice velocities $\mathbf{c}_\alpha$, where α ranges from 0 to 8. These velocities are convenient in that all of their Cartesian components are 0 or ± 1. Their magnitudes differ however.

The discretized Lattice-Boltzmann equation with BGK approximation reads:

$$f_\alpha(\mathbf{x} + \mathbf{c}_\alpha \Delta x, t + \Delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}(f_\alpha(\mathbf{x}, t) - f_\alpha^{eq}(\mathbf{x}, t)) \tag{2.6}$$

where Δx is the lattice spacing and Δt is the time step during which the particles travel one lattice spacing. The local equilibrium distribution functions f^{eq} are obtained from the Maxwell-Boltzmann equilibrium distribution (see Equation 2.3). For a D2Q9 lattice model, the equilibrium distribution function is defined as:

$$f_\alpha^{eq}(\mathbf{x}) = \omega_\alpha \rho(\mathbf{x})[1 + 3\frac{\mathbf{c}_\alpha \cdot \mathbf{u}}{c_s^2} + \frac{9}{2}\frac{(\mathbf{c}_\alpha \cdot \mathbf{u})^2}{c_s^4} - \frac{3}{2}\frac{\mathbf{u}^2}{c_s^2}] \tag{2.7}$$

In Equation 2.7 the weights ω_α are 4/9 for the rest particles, 1/9 for $\alpha = 1, 2, 3, 4$ and 1/36 for $\alpha = 5, 6, 7, 8$, while c_s is the basic speed on the lattice (one lattice unit per time step). Rothman & Zaleski [110] show how this idealized microscopic dynamics give rise to the isotropic macroscopic hydrodynamics described by the Navier-Stokes equations. The model detailed above is sufficient to recover the continuity equation and the momentum equations with second-order accuracy in space and time, although the left hand side of Equation 2.6 resembles a forward Euler scheme. It is in addition possible to use implicit time discretization. With the BGK approximation, this model cannot recover the energy equation, and is therefore most often used to simulate incompressible fluids only. In a practical way, Equation 2.6 can be split in a streaming step and a collision step. The numerical procedure can be then defined in three successive steps:

1. Compute the macroscopic variables on each grid node, and evaluate each of the nine equilibrium terms.

2. Evaluate the collision term, which is proportional to the difference between f_α and f_α^{eq}

3. Evaluate the transport term by propagating the result from the collision one lattice cell in each direction, and update the populations to their value at time $t + 1$.

For further details on the Lattice-Boltzmann modeling, the reader is invited to look at the books of Sukop & Thorne [124], and Rothman & Zaleski [110].

Standard model for wall treatment: Bounce-Back modeling

Given the regular characteristic of a lattice grid, the space discretization of a Lattice-Boltzmann simulation constitutes an important restriction as it cannot adapt to the arbitrary shape of boundaries. Due to its ease of implementation and because it is consistent with the stair-case format of the lattice grids, the bounce-back walls are the most classic boundary condition used for no-slip wall. The theory behind the model is that at a microscopic level, after collision with a wall, the molecules are scattered back in average to the direction they are coming from. Regarding the implementation for the bounce-back nodes, the normal collision-streaming process is omitted and the algorithm merely reads:

$$f_\alpha(\mathbf{x}, t + \Delta t) = f_\beta(\mathbf{x}, t) \tag{2.8}$$

where α is the index of unknown particle populations on the wall, and β is the opposite direction $\mathbf{c}_\alpha = -\mathbf{c}_\beta$. The process is detailed in Figure 2.7.

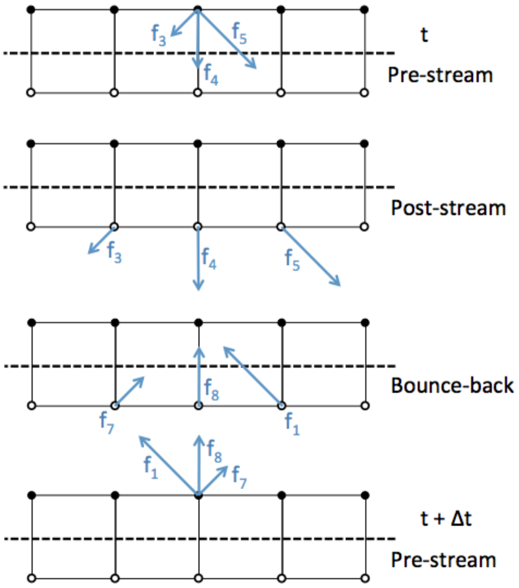


Figure 2.7: Sketch of the bounce-back algorithm. • and ◦ are respectively the fluid nodes and bounce-back nodes.

The effect of the bounce-back modeling is like having a wall halfway between the fluid and the solid bounce-back nodes. Walls represented by this method are approximated by a stair-case shape, as illustrated in Figure 2.8. With this method, it is also possible to divide the solid nodes between those which are approximated as bounce-back and the interior ones which are isolated. A removal of the latter from the discrete domain avoid then unnecessary computations at inactive nodes.

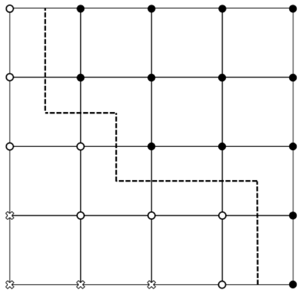


Figure 2.8: Sketch of a lattice grid with fluid nodes •, bounce-back nodes ◦ and interior nodes x. The virtual halfway wall is represented as a dashed line.

In the case of a no-slip wall with a non-zero velocity, a simple bounce-back is not sufficient, and a detailed momentum balance must be made when copying to opposite locations. A review of the boundary conditions on straight walls can be found in [80].

The main limitations of the bounce-back method lie in (i) a stair-case representation of curved walls and (ii) achieving a second order accuracy only in the case of a straight wall which is actually midway between two lattice nodes (and achieving first order in all other cases).

High order models for wall treatment

The bounce-back model has been extended by Bouzidi et al. [14] for curved wall located arbitrarily between grid nodes. Their second-order method uses interpolation or extrapolation schemes to compute the new streaming step, which involves a parameter $0 \leq q \leq 1$ describing the location of the wall between two consecutive nodes. The method of Bouzidi et al. is valid for moving boundaries in addition, and an extended validation can be found in [79].

Guo et al. [49] proposed a similar boundary treatment for moving curved walls. The originality of the method lies in the decomposition of the distribution at a boundary node into an equilibrium part f^{eq} and a non equilibrium part f^{ne}. The velocity involved in the equilibrium is extrapolated from a neighbor in the bulk of the fluid and from the imposed velocity value on the boundary. On the contrary to the bounce-back type of condition, the boundary nodes execute here a normal collision step like all the fluid nodes. The model of Guo et al. [49] achieves also a second-order accuracy.

Immersed Boundary Methods (IBM)

Similarly to Navier-Stokes IBM, the presence of an immersed object in the fluid is accounted for with a forcing term included in the discrete Lattice-Boltzmann equation introduced before (Equation 2.6). Guo et al. [50] have worked on the treatment of this force in LBM. The authors have shown that in order for the LB equation with forcing term to match the correct NS equations, discrete lattice effect must be considered. Equation 2.6 becomes in this case:

$$f_\alpha(\mathbf{x} + \mathbf{c}_\alpha \Delta t, t + \Delta t) - f_\alpha(\mathbf{x}, t) = -\frac{1}{\tau}(f_\alpha(\mathbf{x}, t) - f_\alpha^{eq}(\mathbf{x}, t)) + \Delta t F_\alpha \tag{2.9}$$

where F_α is a component of the force $\mathbf{F}$. Guo et al. [50] have demonstrated firstly that in order to obtain the correct continuity equation, the fluid velocity must be defined such that the effect of the external force is included. Secondly, to obtain the correct momentum equation, the contributions of the force to the momentum flux must be canceled. Thereby the estimation of the velocity (Equation 2.5 in the case without forcing) and the force are derived in Equations 2.10 and 2.11 respectively.

$$\rho \mathbf{u} = \sum_\alpha f_\alpha \mathbf{c}_\alpha + \frac{\Delta t}{2}\mathbf{F} \tag{2.10}$$

$$F_\alpha = (1 - \frac{1}{2\tau})\omega_\alpha[\frac{\mathbf{c}_\alpha - \mathbf{u}}{c_s^2} + \frac{(\mathbf{c}_\alpha \cdot \mathbf{u})}{c_s^2}\mathbf{c}_\alpha] \cdot \mathbf{F} \tag{2.11}$$

Most of the recent IBMs use the approach of Guo et al. [50] for the introduction of a force in the Lattice-Boltzmann equation. A few other approaches have been developed previously for this specific issue, like in [77] and [16]. However Guo et al. [50] have highlighted the limitations of those other models and even pointed out the erroneous derivation of the momentum equation in [16].

2.2 IBM in the Navier-Stokes context (NS-IBM)

The original immersed boundary method was first introduced by Peskin [101] for the Navier-Stokes equations. Currently the immersed boundary methods refer to the methods dealing with immersed obstacles, which emerged from the original method of Peskin. Complete reviews have been written by Mittal and Iaccarino [91] and by Sotiropoulous and Yang [119].

2.2.1 Types of NS-IBM

IBM can be classified in different subgroups according to the formulation of the forcing term, which accounts for the presence of the immersed body.

Continuous forcing and discrete forcing approaches

All the Immersed Boundary (IB) techniques can be organized in two different families of methods, which leads to different types of implementation. In the first type of approach, namely the continuous forcing approach, a momentum source term is added to the governing equations to account for the desired boundary condition at the interface with the body. This modification is performed before the discretization, and therefore the momentum conservation equation becomes the following for incompressible flows :

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho\nabla\cdot(\mathbf{uu}) - \mu\nabla^2\mathbf{u} = -\nabla p + \mathbf{f} \tag{2.12}$$

where $\mathbf{f}$ is the forcing term which is required to impose in general a Dirichlet condition for the fluid velocity. The forcing term can depend on the spatial position, and on both fluid and solid velocities, $\mathbf{u}$ and $\mathbf{u_{ib}}$.

On the other hand, for the discrete forcing approaches, the forcing term is added after the discretization of the governing equations. Thereby the system of equations is modified. This family of methods is thus dependent on the type of discretization used, unlike continuous forcing approaches. A typical immersed boundary method using a discrete approach can be written as below:

$$\frac{\mathbf{u_i^{n+1}} - \mathbf{u_i^n}}{\Delta t} = \mathbf{RHS_i} + \mathbf{f_i} \tag{2.13}$$

where $\mathbf{RHS_i}$ includes all the discretized terms of the Navier-Stokes equation, except the transient term.

Eulerian and Lagrangian approaches

Another type of classification for the IB techniques relies on the framework in which the methods are based. In the Eulerian context, the cells inside the solid are identified by mask functions. Thereby, the forcing term introduced in 2.12 or in 2.13 is generally of the following form:

$$\mathbf{f}(\mathbf{x}, t) = \chi(\mathbf{x}, t)\mathbf{g}(\mathbf{x}, t) \tag{2.14}$$

where χ is a mask function, which is 1 inside the solid and 0 within the fluid for instance, and $\mathbf{g}$ is the forcing term for the boundary condition. The forcing term, through the mask function, is generally active only in the vicinity of the interface. However for some IBMs, the forcing term is present as well inside the solid (see below).

The mask function in Equation 2.14 can be obtained either geometrically, or from a surface mesh definition of the solid. In the latter case, the surface mesh can be used to identify Eulerian nodes inside the fluid domain from those inside the solid one.

When the solid is defined by a surface mesh or a simple set of Lagrangian points, it is then also possible to build an IBM in a Lagrangian framework. In this context, two grids are used in the model, i.e. an Eulerian grid for the fluid and a Lagrangian grid for the solid. The latter is defined with the Lagrangian coordinate of the solid $\mathbf{x_{ib}}$. Hence, the forcing term, which defines the effect of the solid on the surrounding fluid, is translated from one grid to another as described below:

$$\mathbf{f}(\mathbf{x}) = \int_\Gamma \mathbf{F}(\mathbf{x_{ib}})\delta(\mathbf{x} - \mathbf{x_{ib}})d\Gamma \tag{2.15}$$

Thanks to Equation 2.15, the force is distributed on the fluid points in the vicinity of the solid point $\mathbf{x_{ib}}$, according to the support of the smoothed delta functiion δ. Lagrangian IBMs are generally used when the IB velocity is not known *a priori*, but is a variable of the problem, like for the transport or the deformation of flexible solids. Those methods are very suitable for elastic boundaries for instance. In this case, the Lagrangian coordinates are governed by a motion equation for the structure. The Lagrangian force $\mathbf{F}$ in Equation 2.15 can be computed from the solid velocity, or from the difference between the solid position and an equilibrium position, as detailed by Lai and Peskin [78]. The suitability of the forcing term definition strongly depends on whether the immersed body is moving or not.

Sharp and diffuse approaches

Immersed boundary techniques can also be classified according to the treatment of the interface, i.e. in a sharp or in a diffuse way. Most of the IBMs are diffuse methods, where the effect of the IB solid on the fluid is distributed on the surrounding cells/nodes. This leads obviously to a less accurate definition of the interface, but on the other hand, the stiffness of the matrix system is reduced, and stability problem can be overcome. The diffusion introduced is also crucial for coupling problems. For sharp methods, this numerical diffusion is eliminated and the interface fluid-solid influences only one or two layer of Eulerian cells.

Lagrangian IBMs are diffuse methods by definition, and the level of diffusion is governed by the support of the delta function used in Equation 2.15. This function is a key parameter in the method, and different estimations can be found in literature as in Lai and Peskin [78] or in Beyer and Leveque [12]. The latter authors propose an analysis of the resulting accuracy for various choices of the discrete delta function.

Eulerian IBMs are implemented with both approaches for the treatment of the interface. In diffuse approaches, the mask function introduced in Equation 2.14 is smeared over several cells, as proposed by Gazzola et al [40], or it can be defined as a volume ratio function as in Introni et al. [60]. Sharp techniques, however more common, are employed with the unmodified heavyside function for the mask (see Bergmann and Iollo [10] , Kevlahan and Ghidaglia [70]).

Explicit and implicit approaches

Always regarding the forcing term, one can differentiate the implicit IBMs from the explicit methods. As in realistic applications, the background Eulerian nodes do not necessarily coincide with the Lagrangian solid points, the forcing term is not imposing exactly the solid velocity $\mathbf{u_{ib}}$ in an Eulerian node, but rather a reconstructed velocity which depends on both the neighboring fluid and solid velocities. These reconstructions are necessary to achieve high-order schemes and are discussed in Section 2.2.3.

Considering Eulerian IBMs, this dependence of the modified imposed velocity on the neighboring fluid velocities, can be formulated in an explicit or in an implicit way. With an explicit approach, the corrected velocity is obtained from fluid velocities of the current time step, while with an implicit approach, fluid velocities from the next time step are used [136].

For IBM of a Lagrangian diffuse type, the force distribution and the velocity interpolation are coupled within an implicit approach as detailed in [123], while they are decoupled within an explicit approach.

Not surprisingly, implicit IBMs are more arduous to implement and requires more computational resources. However, their advantage lies in a more accurate imposition of the boundary condition at the interface. This avoids for instance the penetration of any streamlines inside a fixed solid body, as can be the case for explicit IBMs.

On the solution of the NS equations inside the IB

Although the solution is not of interest in the solid domain, the momentum and mass conservation equations can be solved within this domain: this is the last distinction between IBMs.

Indeed, the IB forcing term can be either active only in the vicinity of the fluid-solid interface to ensure the desired boundary conditions, or either in the whole solid domain.

For the first type of approaches, the Eulerian nodes inside the solid where the forcing is not active, are usually removed from the solution system for the whole simulation, or at the considered time for moving boundaries. For the second type of approach, although the solution is not of interest inside the solid, the resolution of the fluid equations arises questions about the validity of the pressure field for instance. The distinction between both types of method is finally very similar to the distinction between continuous and discrete approaches, as discrete approaches generally belong to the first type, while continuous approaches do not belong to this group.

Summary

A summary diagram linking each method to references from literature is given in Figure 2.9. This diagram is a simplification and is not exhaustive. It should be noted in particular that some original IBMs cannot fit this classification. For instance, the Tensorial Penalty Method of Vincent et al [143] or the IBM of Borazjani et al. [13], do not use an explicit forcing term in the momentum equation. For Vincents' method, the boundary condition for velocity is imposed with a modification of the fluid viscosity, while for Borazjani's method, the Dirichlet condition is directly imposed in the desired Eulerian nodes.

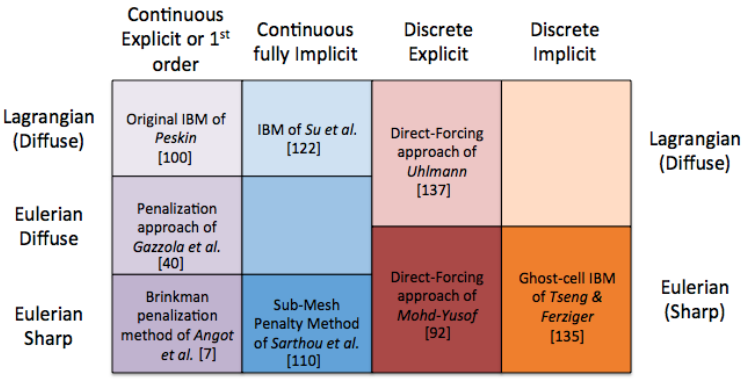

Continuous Explicit or 1st order
Continuous fully Implicit
Discrete Explicit
Discrete Implicit
Lagrangian (Diffuse)
Eulerian Diffuse
Eulerian Sharp
Lagrangian (Diffuse)
Eulerian (Sharp)
Original IBM of Peskin [100]
IBM of Su et al. [122]
Direct-Forcing approach of Uhlmann [137]
Penalization approach of Gazzola et al. [40]
Brinkman penalization method of Angot et al. [7]
Sub-Mesh Penalty Method of Sarthou et al. [110]
Direct-Forcing approach of Mohd-Yusof [92]
Ghost-cell IBM of Tseng & Ferziger [135]

Figure 2.9: Diagram summarizing the different types of IBM. The following papers are provided to illustrate the different types of IBM: Peskin [101], Gazzola et al. [40], Angot et al. [7], Su et al. [123], Sarthou et al. [111], Uhlmann [138], Mohd-Yusof [93], and finally Tseng & Ferziger [136].

It is important to distinguish as well the first-order IBM, from the IBM with high-order schemes for the representation of the interface (see Section 2.2.3). In this sense, the penalization term in the work of Gazzola et al. [40], for instance, is treated as implicit. However, as a first order approach, the forcing term once discretized only depends on the considered cell, and this definition only introduces an additional term in the matrix diagonal, which does not make the overall scheme very different from an explicit approach. On the contrary, in the Sub-Mesh Penalty Method of Sarthou et al. [111] the forcing term, treated implicitly as well, involves the neighboring fluid cells, which makes the matrix formulation more complicated but allows to achieve a second order accuracy.

One can remark that for discrete approaches, the distinction is not made between sharp and diffuse Eulerian methods. Indeed, the Eulerian methods are generally employing, for few background nodes in the vicinity of the solid surface, a forcing term which is based on a reconstructed velocity. Allowing in this case for a sharp and accurate definition of the solid, a smearing of the forcing term, like in a diffuse approach, would cancel this advantage.

2.2.2 Examples of NS-IBM

The original IBM

The original IBM by Peskin [101] was aimed at simulating the blood flow through an heart valve. The method was developed in a continuous context with a diffuse Lagrangian formulation. In this method, the IB is modeled as an elastic surface and hence the Lagrangian force $\mathbf{F}$ of Equation 2.15 defined for each solid point was related to the Hooke's law. In addition, the fluid and the solid motions are coupled with the following equation:

$$\frac{\partial \mathbf{x_{ib}}}{\partial t} = \mathbf{u_{ib}}(\mathbf{x_{ib}}, t) \tag{2.16}$$

where the IB velocity $\mathbf{u_{ib}}(\mathbf{x_{ib}}, t)$ is obtained from the surrounding fluid velocity, similarly to the convolution of the force from the Lagrangian to the Eulerian context in Equation

2.15.

The limitations of the latter method appear in the rigid limit. The high stiffness coefficient from Hooke's law may trigger instabilities. In this context Goldstein et al. [47] have developed a forcing term which is valid for rigid bodies:

$$\mathbf{F(t)} = \alpha \int_0^t \mathbf{u}(\tau)d\tau + \beta\mathbf{u(t)} \tag{2.17}$$

In the equation above, α and β are coefficients selected to best enforce the boundary condition at the immersed solid interface. The force field represents a feedback of the velocity field information. The latter method can be seen as well as a type of penalization (or penalty) method.

Another type of approach which has been originally proposed to get rid of the limiation of the IBM of Peskin for rigid solid, is namely the direct forcing methods, presented later.

The Penalty methods

For these methods, the forcing term is a penalty equation which is added to satisfy the desired boundary condition in the solid. The penalty method was first introduced by Arquis [8] in the following way:

$$\mathbf{f}(\mathbf{u}, \mathbf{u_{ib}}) = \lambda(\mathbf{u_{ib}} - \mathbf{u}) \tag{2.18}$$

where λ is the so-called penalty constant, which had originally a physical meaning.

In the Darcy penalty method for instance (see Khadra et al [71]), such constant can be written as $\lambda = \frac{\mu}{K}$, where K is the local permeability. For the fluid domain the permeability tends towards infinity, and the original Navier-Stokes equation is thus obtained. On the contrary, physical values are fixed for K in the porous solid in order to model the fluid-porous flow interaction. And finally, in the non-porous part $K \to 0$, which is ensuring the correct solid velocity $\mathbf{u} = 0$ (for $\mathbf{u_{ib}} = 0$).

Another example is the volume penalty method, firstly used by Angot [6] for heat transfers in electronic components. In this method, the penalization constant is derived from the Biot parameter, which is also used to impose a Dirichlet condition for the temperature.

From a general point of view, the penalization constant is commonly set to 0 in the fluid domain, and to a large value inside the immersed body in order to neglect all the terms in regards to the forcing term. In this sense, the penalty methods can be categorized in general in the Eulerian and sharp class of IBM. Angot et al have presented rigourous estimates of the error introduced by such estimation in [7]. However diffuse penalty methods have also been investigated as in [40] and [70] for instance.

Another type of penalty approach is the tensorial penalty method, where the stress tensor is penalized instead of the velocity. In this approach, the stress tensor is decomposed in several terms according to the physical contributions of compression, tearing, shearing and rotation, as proposed by Vincent et al [143]. The stresses are thus dissociated and different viscosities are imposed in the fluid and the immersed body. Thereby, if the latter domain is a solid, high viscosities will avoid shearing and tearing and ensure a constant rotation.

The penalty methods belong generally to the continuous methods, but not necessarily. However, the distinction between continuous and discrete methods is weak for these type of models. As the penalty forcing term is merely a linear velocity term, its introduction in the discretized equation is straightforward and does not depend on the spacial schemes. Thereby, in this document the penalty methods are always considered as continuous methods.

The Direct Forcing methods

The direct forcing type of method was introduced by Fadlun et al [34] in order to avoid numerical instabilities encountered with Peskin's type of method [78] for rigid bodies.

Unlike the penalty methods, the direct forcing methods are generally used in a discrete context, after the discretization of the equations. For this method, the forcing term introduced in Equation 2.13 is simply expressed as follows:

$$\mathbf{f}(\mathbf{u}, \mathbf{u_{ib}}) = -\mathbf{RHS} + \frac{\mathbf{u_{ib}} - \mathbf{u^n}}{\Delta t} \tag{2.19}$$

The forcing term cancels all the terms in the Navier-Stokes equation except the transient one, and thus ensures the Dirichlet condition for the velocity at the considered points: $\mathbf{u^{n+1}} = \mathbf{u_{ib}}$, which is generally a no-slip condition for a solid. For the direct forcing method, the forcing term does not have a physical meaning. Those methods can be used either in an Eulerian frame or a Lagrangian frame, and either with a diffuse or sharp definition of the interface.

Another way of defining the forcing term for direct forcing methods reads:

$$\mathbf{f}(\mathbf{u}, \mathbf{u_{ib}}) = \frac{\mathbf{u_{ib}} - \mathbf{u}^*}{\Delta t} \tag{2.20}$$

where $\mathbf{u}^*$ is a first estimation of the velocity without considering the effect of the immersed body. This velocity is thus solution of the equation $\frac{\mathbf{u}^* - \mathbf{u^n}}{\Delta t} = \mathbf{RHS}$. The above formulation is generally employed when this estimated velocity is computed in a preliminary step [138], [147], [85].

2.2.3 High-order NS-IBM

The methods described above are generally first-order in space, and significantly depend on the mesh definition. In each IBM, the purpose of the forcing term is to impose the desired boundary condition for velocity, pressure and any other quantities at the selected Eulerian nodes. Thereby, for complex geometries with high-curvature, where the secondary geometry does not coincide with the Eulerian mesh, these simple models are not sufficient because the desired condition is not satisfied at the correct location. In this context, a range of improvements have been proposed in literature to handle the weaknesses of these methods, whose implementations highly depend on the type of IBM considered. The improvement techniques are classified into two main groups [58]:

- schemes that produce a local reconstruction of the solution based on the target boundary value and surrounding fluid values. These schemes are used with sharp type of methods.
- schemes that spread the function in the vicinity of the immersed interface, for the diffuse (mostly Lagrangian) type of methods.

In sharp methods

For the sharp type of methods, the reconstruction procedure allows the boundary condition to be satisfied at the exact position of the interface. In this case, the physical quantity imposed by the forcing term is generally obtained from an interpolation between the desired value at the interface, and the values in the neighboring fluid nodes. Linear interpolation is generally used and can achieve second-order accuracy [43]. However, specific configurations where Eulerian nodes are very close to the solid interface leads to large weighting coefficients and can destabilize the computations. To avoid this issue, bilinear (and trilinear in 3D) interpolations are proposed [9], [73], as well as quadratic interpolation [45]. The latter was found to also improve the accuracy of problems with moving boundaries as compared to simple linear interpolation. Another approach for the problem of ill-conditioned weights was developed by Tyagi et al. [137] with a forcing on both sides of the immersed boundary. One can note that a more straightforward method than the interpolation consists in weighting the forcing terms in each cell with the volume fraction occupied by the body with respect to the cell volume [34]. All the IBMs listed above are explicit methods, as the physical values imposed by the forcing term are interpolated from surrounding values from the previous time step. Implicit approaches were also proposed, as it is the case in [136], which is based on a ghost-cell method and can treat both Dirichlet and Neumann conditions.

Considering direct forcing methods, the forcing term of this type of approach involves time-derivative of the velocity. This is problematic for moving immersed bodies, when a grid node previously in the solid domain is freshly exposed to the fluid. For that type of node, the time-derivative is not supposed to be physically correct. Gilmanov et al. [45] use a dual-time stepping to solve this issue, while the solution in [155] is based on a field extension and a similar algorithm is proposed in [160]. Other types of schemes can be used to improve the results for moving boundaries. Hence, Liao et al. [85] use a combination of the standard linear reconstruction in the fluid with a solid-body forcing in the solid which has similarities with penalty methods.

While the boundary condition for velocity is clear, a boundary condition for pressure at the immersed interface is not always imposed. A non-negligible amount of authors do not propose any specific correction for the pressure at the interface. In this context, Yang et al. [155] argues that the boundary condition for the pressure $\frac{\partial p}{\partial n} = -\rho \frac{D\mathbf{u}}{Dt} \cdot \mathbf{n}$ which is usually enforced directly in a body-conforming approach, is obtained implicitly with IBM from the projection of the inviscid momentum equation on the solid boundary. Indeed, the viscous term turns out to be zero when all velocity components are linearized in the vicinity of the interface thanks to the interpolation based correction of the velocity. However, numerous discrete IBMs make the choice of imposing also a Neumann condition for the pressure at the immersed interface [43], [136], [90], as a numerical condition for the Poisson equation. The issue with the pressure at the immersed boundary is well detailed in the work of Kang et al. [66]. The authors propose a versatile interpolation-based correction of the velocity at the interface, which enforces mass conservation for an approximated fluid domain strictly included in the physical fluid domain. This correction satisfies the pressure decoupling constraint between the fluid and the solid domains, and solves the incompatibility between the interpolated velocity boundary condition and mass conservation, which, from the author's knowledge, has seldom been investigated in IBM with velocity corrections at the interface. Regarding the mass conservation issue again, Kim et al. [73] developed a method introducing a mass source/sink in addition to the momentum forcing, in order to satisfy the continuity in the cells containing the immersed boundary. Most of the recent IBMs including treatment for mass conservation are either based on the works of Kang et al. [66] or Kim et al. [73]. One can note as well the original method of Kumar & Roy [75] using a Solution Algorithm type of pressure correction that strongly enforce mass conservation while assuring a second order accurate imposition of the boundary condition. With this algorithm the continuity equation is iteratively solved in the local cell only, instead of using the usual pressure Poisson equation to achieve incompressibility.

Finally for continuous approaches, whose most common representatives are the penalization methods, no corrections are usually employed at the interface in order to take into account the rasterization effect [71], [10], [2], [89]. The overall accuracy of these methods are thus close to first order in space. To the author's knowledge, the Sub-Mesh Penalty Method of Sarthou et al. [111], the second-order penalty model of Introini et al. [60] ,and the work of Chantalat et al. [20] are the only models based on penalization which offer improvements to tackle the rasterization effect on Cartesian grids. In both the Sub-Mesh Penalty Method and the approach of Introini et al., the velocity is corrected at the nodes close to the interface by means of interpolation, similarly to the discrete approaches discussed earlier. In addition, a Neumann boundary is satisfied by the pressure in the model of Introini et al., thanks to a consistent fractional step method. The forcing term is in this case also taken into account in the correction step. On the contrary, in the the Sub-Mesh Penalty Method no specific treatment is undertaken for the pressure. In the work of Chantalat et al. [20], the penalization algorithm is improved using the velocity gradient and level-set information.

In diffuse methods

Unlike the schemes used in sharp methods, the spreading of the forcing function in diffuse methods produces extra dissipation. On one hand, the spreading removes the strong oscillations in the hydrodynamic forces, usually obtained with sharp methods for moving immersed boundaries [138]. These oscillations can be problematic if two-way coupling is considered. On the other hand the dissipation in diffuse approaches leads to more inaccurate results at similar resolution to the sharp methods [160], and this may deteriorate the prediction of the boundary layer development. In addition, diffuse methods are limited to first order in space and can achieve only a formal second order at maximum [78].

2.2.4 Treatment of the coupling in NS-IBM for FSI problems

For fluid-structure interaction problems, when the velocity of the structure is not known *a priori*, an additional solver is required to describe the motion of the latter. The governing equation for the structure usually reads as follow:

$$\rho_{ib}\frac{\partial^2 \mathbf{x_{ib}}}{\partial t^2} = \nabla \cdot \sigma_{ib} + \mathbf{F_{ext}} \tag{2.21}$$

In Equation 2.21, $\mathbf{F_{ext}}$ represents the external force and σ_{ib} is the stress tensor given by the constitutive equation of the structure. The coupling is then achieved either thanks to the action of the fluid on the solid, included in the stresses, or with the boundary condition for velocity at the interface. The coupling process is discussed later. Furthermore, two kind of coupling can be distinguished [119]:

- The loose coupling, where the governing equation for the fluid and the solid are solved sequentially one after the other
- The strong coupling, where between each fluid iteration, several sub-iterations are computed for the structure until a convergence criterion is obtained.

Strong coupling schemes allow to reduce the eventual numerical instabilities, generated by a more inaccurate imposition of the boundary condition at the interface, potentially given by a loose coupling. This is especially true when the density ratio between the two continuum becomes close to 1, and has been described as an added-mass effect [13].

Explicit and implicit coupling can also be differentiated. In implicit coupling the IB forcing term for the next time step $n+1$ is computed using the structure velocity at the same time step $n+1$. In return, this structure velocity is computed also from the IB forcing at the next time step, leading to a linear system to invert. An implicit coupling scheme is for instance detailed in Lacis et al. [76].

Coupling with rigid solids and application to particle-laden flow

In the framework of rigid solids, Equation 2.21 becomes the standard Newton equation of motion for the center of mass of the structure:

$$m_{ib}\frac{\partial \mathbf{U_{ib}}}{\partial t} = \int_{\Gamma} \tau \cdot \mathbf{n} d\Gamma + \mathbf{F_{ext}} \tag{2.22}$$

What remains a difficulty and distinguishes the different IBMs is the formulation of the force that the fluid exerts on the body, and which is defined by the surface integral of the hydrodynamic stresses $\tau = -\mathbf{I}p + \mu\nabla^2\mathbf{U}$ in Equation 2.22.

The fluid stresses can be decomposed to a contribution from the pressure and a contribution from viscous stresses. In order to estimate this surface force, Calderer et al. [18] neglected the viscous term and implemented a pressure projection boundary condition method (PPBC), which approximates the surface of the solid body by the closest grid pattern, and estimates the pressure on the real solid surface according to a Neumann condition. Their IBM was successfully applied to arbitrary solid shapes in multiphase flow at high Reynolds number ($Re > 10^5$). A similar approach to Calderer et al. [18] was employed by Borazjani et al. [13] for flexible solids. For very light particles, Schwarz et al. [114] developed a method based on the injection of an added mass term on both sides of Equation 2.21, in order to remove the singularity when $\rho_{ib} \to 0$.

A more intuitive approach is to derive the formulation of the fluid stresses from a momentum balance of the fluid enclosed in the solid body. For IBMs solving the fluid equations inside the structure, the inertial term is non-zero and is given by:

$$\frac{d}{dt}\int_{\Omega_s} \rho_f \mathbf{U} d\Omega = \int_{\gamma} \tau \cdot \mathbf{n} d\Gamma + \int_{\Omega_s} \mathbf{f_{f\to s}} d\Omega \tag{2.23}$$

In Equation 2.23, the body force $\mathbf{f_{f\to s}}$ acting in the control volume is merely the opposite of the IB forcing $\mathbf{f}$ in the fluid momentum equation, which takes into account the presence of the solid. Under the widely used assumption of rigid motion for the fluid inside the solid, the inertial term can be simplified to $\rho_f \Omega_s \frac{d\mathbf{U_{ib}}}{dt}$ and can be injected in the left hand side of Equation 2.21. For cases where the rigid body assumption is not satisfied, Kempe & Frohlich [68] derived another approach to numerically evaluate the volume integrals in Equation 2.23. The authors also developed an IBM which imposes a slip condition for the velocity in order to handle bubbly flows [69], instead of the common no-slip condition for particles.

Finally, in the versatile method of Vincent et al. [143], the flow around the particles is sufficiently resolved to account for the forces acting on the fluid with a four-way coupling. In this context, it is no longer necessary to solve a motion equation for the rigid solids. This fictitious domain approach has been applied to dense particulate flows. More standard IBM ([138], [147], [157]) have been applied to multi-particle systems as well. In those methods, a collision force is developed to model the interaction particle-particle and is added to the motion equation.

Shwarz et al. [115] have extended a direct forcing IBM for bubbles of varying shapes. The shape of the bubble is computed by minimizing the local displacement energy of pressure and surface tension forces, but the authors have limited their investigation to axisymmetric shapes. On another note, Wang et al. [145] applied a vorticity-based IBM to the study of self-excited oscillations of a system of many linked plates.

As stated in previous sections, diffuse IBMs are widely used to handle FSI problems as they tend to smooth the estimation of the forcing term and thus stabilize the computations. One can note however the work of Kim et al. [73], and that of Yang & Stern ([156],[157]) who developed sharp direct forcing approaches to model flow interaction with a rigid body. In both their papers, Yang & Stern used a field-extension strategy to tackle the issue of spurious force oscillations. Furthermore, on their more recent work [157], the authors implemented a versatile non-iterative strong coupling which combines straightforward implementation and stability improvements for low density ratio problems.

Coupling with flexible solids

Similar to the work of Schwarz et al. [115] introduced above, several authors have developed IBMs able to handle flexible solids, with imposed deformation. The main application of this kind of IBM is self-propelled fish-like swimming. The deformation of the fish is controlled by a function of time, which dictates the action exerted by the fluid on the fish, governing at the end its trajectory using the motion equation. For the computation of the force exerted by the fluid, Ghaffari et al. [42] follows the same methodology based on the momentum balance introduced previously (Equation 2.23), while Gazzola et al. [40] omit the inertial term inside the solid domain. Bergmann & Iollo [10] used another strategy by converting the surface integrals of the stresses τ in volume integrals of the equivalent Navier-Stokes terms in an arbitrary domain surrounding the body. It is important to note that all the IBMs applied to fish-like swimming are sharp first-order penalization methods. Although first-order, the sharpness of these methods may still lead to stability issues. In this context, Ghaffari et al. [42] have developed a filter to denoise the force exerted by the fluid on the immersed body.

For arbitrary deformation of the solid, the most accurate approach is to couple the fluid solver with a structural dynamic solver, such as the Finite Element Method (FEM) for instance. These kind of approaches are however extremely computationally expensive. To tackle this cost issue, Gilmanov & Acharya [44] developed a coupled IBM - Material Point Method (MPM). MPM is a meshless solid solver, which has the advantage of being able to consider large deformation, including situation with material rupture. Their hybrid IBM was applied to a deformable sphere falling and a capsule in shear-flow.

For more specific deformation, IBMs are most generally used to model the structural dynamics of 1D filaments and 2D membranes. In these cases, as the solid has at least one dimension less than the fluid problem, the only possibility is obviously to use diffuse IBMs. If solid membranes of finite thickness are considered, sharp IBMs can still be carefully employed as it is the case in [82], provided that the Eulerian grid size is smaller than the thickness of the solid.

In general, one can distinguish two types of IBM to deal with flexible solids:

- The approaches where the forcing term in the fluid momentum equation has a physical meaning, and is calculated from the mechanical force obtained from the membrane deformation. For these approaches, the coupling is performed through the calculation of this force.

- The approaches where the forcing term in the fluid momentum equation has no physical meaning. The mechanical force is used to compute the new position and velocity of the solid, and the coupling is achieved by imposing a Dirichlet condition for the fluid velocity to be equal to the solid one.

For both types of methods however, the computation of the internal mechanical forces, included in σ_{ib} in Equation 2.21 involves usually the same quantities, namely the stretching, shearing, bending and twisting forces. These forces are estimated based on the spatial derivative of the solid position $\mathbf{x_{ib}}$.

Both the IBM of Siguenza et al. [118] and Le et al. [81] belong to the first group and simulate large deformation of membranes, as well as a heart valve and red blood cells respectively. Le et al. [81] have implemented an implicit time scheme to avoid the problem of having to use very small time steps. As for the IBM of Xu & Wang [154], belonging as well to the first type of approach, their model has been employed to study the flow around a flapping wing.

The second type of approach is more prevalent. The deformation of the membrane is governed by a motion equation involving the internal mechanical forces introduced just above and the hydrodynamic force from the fluid. Similar to rigid solids the latter force can be estimated from surface integration of the fluid stresses, as is the case in [28], from the IB forcing term ([56]), or from the volume integral of the equivalent Navier-Stokes terms ([82]). Both penalization ([33], [56]) and direct forcing ([162], [99], [82]) methods were used to simulate insect flight or filament and flag deformation. Huang et al. [54] developed a versatile penalty IBM involving two sets of solid points, one massive set prone to deformation, and one massless set interacting with the fluid. The two groups of points are linked by a stiff spring with damping. Regarding versatile methods, Tullio et al. [28] modeled the immersed solid as a spring-network system, which account for bending and stretching. Finally, one can note the work of Xu et al. [153], who combined a standard direct forcing IBM for specific Eulerian nodes that prescribes locally the velocity, with a single momentum equation for both fluid and solid, modeling the fluid-structure interaction and the solid deformation.

2.2.5 Treatment of turbulence in NS-IBM and realistic engineering applications in literature

The different reconstructions of the velocity presented in Section 2.2.3 are still valid at high Reynolds number if the boundary layer is sufficiently resolved ($y^+ < 5$), i.e. when the first layer of fluid cells is located in the viscous sub-layer. However in most engineering applications, and especially for IBMs where the mesh is refined in all three directions, achieving such a resolution is unfeasible. In this context, the velocity in the first layer of fluid cells does not exhibit a linear behavior and the usual reconstructions based on linear interpolation are no longer valid. This issue has not been treated until recently, for two reasons mainly. Firstly, the physical problems investigated with IBM were limited to moderate Reynolds number. Secondly, authors who have modelled high Reynolds flow, have been using massive adaptive refinement in order to obtain satisfactory y^+.

Alternative methods have been proposed since. In these methods, the reconstructed velocity complies with wall-layer models, which are logarithmic in order to satisfy the correct wall shear stress. Several IBMs have hence been combined with the Werner and Wengle law of the wall [25], [64], [19] or similar logarithmic models like in Roman et al. [109] or in Calderer et al. [18]. In [21] in particular, the tangential velocity near the surface is constructed as a power-law function of the local wall normal distance. The power is a user-defined parameter, whose appropriate choice enables the method to approximate the energizing effects of a turbulent boundary layer for higher Reynolds number flows. In other wall-layer approaches for LES modeling, the correction of the velocity is based on the Turbulent Boundary Layer (TBL) equation [146]. Under the local equilibrium assumption adopted in [131], [19] and [11] for instance, the equation for the velocity near the wall reduces to:

$$\frac{\partial}{\eta}[(\nu+\nu_t)\frac{\partial u_T}{\partial \eta}] = 0 \tag{2.24}$$

where η indicates the local wall-normal direction, u_T the velocity in the wall-parallel direction, and where ν_t is the eddy viscosity usually obtained from a simple mixing length eddy viscosity model with near wall damping [17]. It is interesting to note that Chang et al. [19] conclude that their predictions without wall models, i.e. assuming linear velocity profile at the wall, actually return better results both in mean and turbulence quantities. Such results were probably obtained because of the relatively low Reynolds number considered in their work.

As for the linear reconstruction detailed in Section 2.2.3, it is important to remember that velocity corrections satisfying wall models are inherent to sharp type of IBMs only. One can highlight the versatile method of Munjiza et al. [64] and Cristallo et al. [25] which uses a diffuse IBM inside the solid, and a sharp IBM in the fluid domain where the velocity reconstruction can then be applied.

Finally, all the wall-layer reconstructions have been developed in combination with LES modeling, except for the work of Kalitzin et al. [65], based on a RANS approach. Similarly all these models have been used only with the discrete direct forcing approach. To the author's knowledge, no wall-layer models have been implemented for continuous approaches like the penalty method.

Regarding the applications, the wall-layer models have been assessed for density-stratified flows [106] or flow over periodic hills [19] at Reynolds around 10^4, and for realistic engineering applications like the flow around a model road vehicle ($Re = 10^5$) [25] and other highly separated flows like the one around a rudimentary landing gear at

$Re = 10^6$[11]. Multiphase flows at high Reynolds ($Re > 10^5$) have been investigated as well [18].

However most research on IBMs has not investigated the consistency of the velocity at the fluid-solid interface with the law of the walls for turbulent flows. Neither do they seem to have used high enough mesh resolution at the immersed wall to ensure a first fluid cell in the viscous sub-layer. For instance, Balaras & Yang ([9], [155]) treated flows along wavy walls at Reynolds number around 10^4, and Zhang et al. [160] considered a moving cylinder and a flat plate at similar level of Reynolds number. Realistic engineering applications have also been considered, like stirred tank at $Re = 7280$ [137], mixed flow in pumps at $Re = 1.5 \cdot 10^5$ [105], and wind and hydro turbines at $Re = 4.2 \cdot 10^4$ and $1.7 \cdot 10^5$, respectively [5]. Finally penalization approaches have also been used in this context for highly turbulent flows without wall-layer models, such as fish-like swimming [10] or hydroplaning [144].

2.3 IBM in the Lattice-Boltzmann context (LB-IBM)

Although rapidly developing, Lattice-Boltzmann Methods (LBM) are still a rather recent method in the CFD community. That is why relatively few IBMs have been developed in a Lattice-Boltzmann context in comparison to the amount of existing IBMs developed in a Navier-Stokes context.

As in the Navier-Stokes context, the definition of the forcing term is what differentiates mainly one LB-IBM from another. The existing types of IBMs implemented with LBM are described in this section. With a simple classification, three types of forcing can be differentiated:

- based on penalty methods. These methods depends then on a user-defined parameter (penalization coefficient, stiffness parameter, etc).
- based on direct forcing methods. Interestingly in these methods, the forcing term is derived from the Navier-Stokes equations and introduced then in the discrete Lattice-Boltzmann equation.
- estimated through other methods, inherent to Lattice-Boltzmann modeling.

One can note also that nearly every IBM developed in the Lattice-Boltzmann framework are based on a spreading of the forcing term. To the author's knowledge, the only coupled LB-IBM using a sharp interface definition has been developed by Kang et al. [67] and is based on a direct-forcing scheme. Finally, in this section a distinction is made as well on the type of application. IBM involving coupling with rigid bodies are distinguished from methods taking into account solid mechanics for the solid deformation.

2.3.1 LB-IBM with forcing based on penalty type of methods

Feng & Michaelides [38] have considered an IB forcing of a penalty type for solving fluid-particles interaction problems like sedimentation. In [38] the forcing term is given by Hooke's law, similar to the original method of Peskin [101]. In this way, the IBM can model a near rigid behavior using a high stiffness, or elastic solid with small deformation.

The Newton equation solved for the particles motion takes into account gravity and collision forces, as well as a restoration force acting like a linear spring (in the case of a rigid body motion). This force, distributed from the Lagrangian points to the Eulerian lattice, is used as a forcing term in the discrete Lattice-Boltzmann equation.

Coupling with flexible solids

Tian et al. [133] have extended the penalty method of Kim & Peskin [74] to handle a moving deformable boundary. As in Huang [54], the flexible filament is modeled here by two sets of points, one massless following the motion equation, and another one with mass, is attached to the first filament through a virtual spring. This approach has shown to be more stable for the estimation of the inertial force of the flexible solid with mass, for any range of density ratio between the fluid and the solid. A motion equation is then solved for the ghost filament with mass, and the forcing term introduced to model the presence of the massless physical filament involves tension and bending forces as well as the spring force between both filaments. Finally, the position of the physical filament is explicitly updated from the surrounding fluid velocity, which is distributed from the Eulerian lattice to the Lagrangian points. Regarding the discrete Lattice-Boltzmann equation, Tian et al. followed Guo's treatment [50] for the introduction of the forcing term.

Zhu et al. in [161] used a similar type of coupling to model a flexible sheet, in the sense that the solid velocity is explicitly obtained from the fluid velocity. The forcing term of a penalty type involves however only the bending and tension components.

2.3.2 LB-IBM with forcing based on direct forcing methods

On the contrary to [38], in [39] Feng & Michaelides used a forcing term estimated from the direct-forcing methods of Mohd-Yusof [93] and Fadlun et al. [34]. This approach eliminates the need for the determination of the stiffness parameter involved in the penalty methods. It is supposed to be more stable and accurate at high Reynolds number, and for complex particle geometries. Furthermore, the methods of Mohd-Yusof and Fadlun et al. ([93] and [34]) are second-order accurate, as an interpolation procedure is employed to improve the imposition of the boundary condition for the velocity on the interface. Regarding the particle motion, Newton's equation is used involving gravity, collision and the hydrodynamic force from the fluid. The latter force can be estimated from integration of viscous and pressure stresses on the particle's surface, or by volume integration of the direct-forcing term. The authors stated that although both are first-order, both estimations lead to similar results. Finally, it is worth mentioning that Feng & Michaelides [39] based their work on Buick & Greated [16] for the introduction of a force in the discrete Lattice-Boltzmann equation. As explained in Section 2.1.2, the Navier-Stokes equations are not perfectly derived at a macroscopic scale in this case.

Dupuis et al. [31] developed an IBM based on the work of Guo et al. [50] for the introduction of an external force. The proposed method is applied to the flow past an impulsively started cylinder. However, the results indicate no significant differences from the same IBM without Guo's treatment for the external force. As for Feng & Michaelides in [39], the force in [31] is estimated from the direct-forcing methods of Mohd-Yusof and Fadlun et al. ([93] and [34]), with the exception that a preliminary momentum step is considered to compute the velocity in the absence of the solid (see Equation 2.20). As noted by Dupuis et al. [31], this preliminary step can be treated in a very efficient manner thanks to two properties of the LBM, which (i) simulates a compressible flow so that the preliminary velocity needs computation only in a neighborhood of the boundaries and (ii) does not change the velocity during the collision step so that only the streaming part is needed.

While no treatment is considered for the introduction of external forcing in the different works of Suzuki & Inamuro ([125],[126], [97]), the latter is estimated again from a direct forcing type of method. More precisely, their IBM is based on the multi-direct forcing approach of Wang [147], where the forcing term is computed iteratively. It is shown that such an approach obtains a better no-slip boundary condition at the immersed boundary, compared to the original direct-forcing methods of Mohd-Yusof and Fadlun et al ([93] and [34]). Suzuki & Inamuro have extended their IBM to avoid the discontinuity of the velocity gradient on the boundary [126]. For this purpose, the velocity field is expanded in the solid domain, as it is the case with ghost cell methods. The latter approach is shown to achieve better than first-order spacial accuracy. Regarding the estimation of the physical force acting on a closed body, it has interestingly been shown that the contribution from the internal fluid must be taken into account for a correct estimation [125]. In Newton's equation for instance, only the contribution of the external force has to be considered. Suzuki & Inamuro [125] have compared different ways of estimating the internal fluid contribution called the internal mass effect. Finally, they also applied their strictly first order IBM to the motion of a flapping wing [97].

Coupling with flexible solids

De Rosis et al. [27] employed the IBM of Suzuki & Inamuro [125] for the coupling of a fluid with flexible flapping wings. Regarding the solid, the structural dynamics is solved with a Finite Element Method.

Favier et al. [35] proposed a model based on the IBM of Pinelli [102], which uses a direct forcing approach. The forcing term in the discrete Lattice-Boltzmann equation is accounted for according to Guo's model [50]. Favier's model [35] is first applied to sedimenting particles, whose rigid motion is solved thanks to Newton's equation with the physical force acting on the solid being the opposite of the IB forcing. The IBM is applied secondly to fluid-flexible filament interaction. In this case, the filament deformations are governed by a motion equation including tension and bending terms. The latter equation is solved with a Finite Difference Method.

2.3.3 LB-IBM with forcing estimated through other methods

Wu & Shu. [150] developed an IBM based as well on Guo's model [50] for the external force. In their implicit scheme, the forcing term is an unknown which is determined in such a way that the velocity at the Lagrangian point satisfies the no-slip condition. The forcing can thus be interpreted as a velocity correction in order to obtain the correct boundary condition. The implicit character of the method improves the imposition of this condition and the authors have shown that no streamlines penetrate the solid body in their test cases (flow around a circular cylinder and an airfoil), which is not always the case for explicit IBM. The model is extended for 3D incompressible viscous flow in [151].

On a more original note, Niu et al. [96] adopted the Generalized Lattice Boltzmann (GLB) model of d'Humieres [30] instead of the common single relaxation time BGK model for the treatment of the collision step. This multi-relaxation time model has been shown to have superior stability over the BGK model. In this framework, the introduction of a forcing term is different and Guo's treatment [50] is not needed, as half of the force is added before the relaxation and the other half after. The forcing term in [96] is estimated by the momentum exchange of the boundary particle density distribution functions. Regarding the coupling for particle motion, it is achieved in Newton's equation, where the hydrodynamic force is merely the opposite of the forcing term. Hu et al.[53] developed a LB-IBM with the same principle of momentum exchange for the forcing estimation. In addition, they used an iterative method to enforce the boundary condition on the immersed solid.

Coupling with flexible solids

In Yuan et al. [159], the momentum exchange-based LB-IBM of Niu et al. [96] is used for the computation of the forcing term. The Lattice-Boltmann equation is however derived with a single relaxation time model instead of the multi-relaxation time model of Niu et al [96]. The method aims to simulate a flexible filament, whose motion is described by a similar equation to the one used in Favier et al. [35], including tension and bending terms.

2.4 Concluding Remarks

A review on the state-of-the-art for IBMs has been presented. Each type of IB techniques proved to be efficient at some point because of one or more specific property(ies). A final comparison of the main type of IBMs is proposed in Section 2.4.1 and the choice of the IBM used in this study is explained in Section 2.4.2.

2.4.1 Comparison of IBM

Table 2.1 provides a summary list of the advantages and disadvantages of the most common IBMs found in literature. It can be concluded that in certain cases there may be only one method suited to a specific application but generally several IBMs may be available. More detailed comparison can be found in literature, as in Piquet et al. [103], who made a comparative study of a penalty and a direct forcing approach for compressible flows.

2.4.2 Choice of IBM for the current problem

In this research, IBMs are used to model two types of immersed boundaries in fluids: (i) three-dimensional rigid components of complex geometries, fixed or moving in a fluid, and (ii) two-dimensional deformable components transported by the carrier phase.

Given the type of engineering applications of interest in this research, the Eulerian mesh used can be relatively coarse if the overall mesh cell count is to remain of the order of few million cells. For this reason, a sharp interface type of IBM is deemed preferable to keep an accurate representation of the rigid immersed components. A new sharp IBM has been developed and then implemented in the OpenFOAM® library using a Navier-Stokes framework. A continuous penalty approach has been used to formulate the forcing term. In order to improve the accuracy and the order of the method on relatively coarse grids, corrections schemes have been implemented for the physical variable to better satisfy their boundary conditions on the immersed interface. Finally, a diffuse interface type of IBM used for the modelling of three-dimensional rigid components has been

Model (IBMs)	**Advantages**	**Disadvantages**
penalty	· Straightforward implementation and mathematical formulation	· User-defined parameter · Matrix system with large stiffness
direct forcing	· Best evaluation of the near-wall variables	· Complex implementation and mathematical formulation · Computationally more expensive
physical	· Can model specific physical behavior	· Not general
sharp	· High-order accuracy achievable · Can be combined with specific boundary corrections (high-order, turbulence, etc)	· Suitable only for components of finite thickness · Stability issues might arise
diffuse	· Suitable for any kind of geometries · More stable	· Loss of accuracy near the interface · Low order only

Table 2.1: Advantages and disadvantages for the most common types of IBMs.

assessed and compared to our novel sharp method. This diffuse IBM is available in the Lattice-Boltzmann library Palabos®.

Regarding the second type of components modelled by IBM (thin flexible immersed structures), the diffuse type of approach is the only viable option given the very small thickness of the flexible membranes. This particular type of approach has been considered in two numerical frameworks, Navier-Stokes and Lattice-Boltzmann. In the latter, the diffuse IBM already used for three-dimensional rigid components has been modified and coupled to a solid solver to model the dynamics of slender deformable structures. This model is then compared to another diffuse IBM implemented in the former Navier-Stokes context. This comparison allows at the same time to assess in parallel two different types of coupling, as the IBM implemented in OpenFOAM® (Navier-Stokes) is based on a penalty approach, similarly to the method developed for the first type of component, and the IBM implemented in Palabos® (Lattice-Boltmzann) is based on a direct-forcing approach.

The different models developed or used in this research are summarized in Figure 2.10.

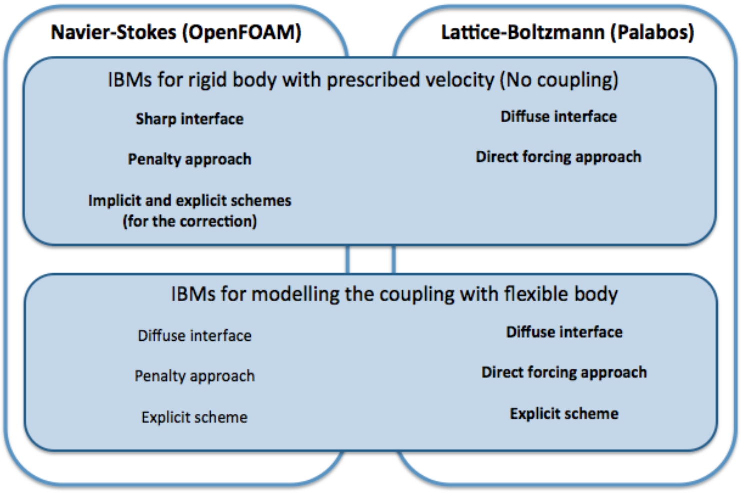


Figure 2.10: Summary of the different IBMs developed and used for this research.

Chapter 3

A versatile sharp NS-IBM for modeling the interaction between fluids and rigid boundaries in OpenFOAM: xIBM

3.1 Immersed Boundary formulation: a penalty approach

Consider the problem of an immersed wall presented in Section 2.1 with Equation 2.1 and Figure 2.1. A rigid solid is represented by Ω_s and is immersed in a fluid domain Ω_f. Γ is the interface between both domains. For the IBM developed in this study, the governing equations in the Navier-Stokes framework are solved in the entire domain $\Omega = \Omega_f \cup \Omega_s$ (see Section 2.2.1 about the types of IBMs).

3.1.1 Treatment of the solid geometry

The interface of the immersed solid Γ is represented by a discrete mesh which may be made of triangular or other polyhedral faces. The mesh is characterized by a set of Lagrangian points $\mathbf{X_{ib,n}}$, with n from 0 to N, N being the number of solid points. These points represent the face centers of the surface mesh. The latter is used to identify the cell centers inside the solid domain from those inside the fluid domain, thanks to an octree partition of space. An octree partition is a tree data structure, in which each domain is recursively subdivided into eight similar octants. This type of partitioning allows to speed-up searching algorithms.

3.1.2 Governing equations

An isothermal and incompressible flow is governed by the following equations of mass and momentum conservation:

$$\nabla \cdot \mathbf{u} = 0 \tag{3.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) - \nu \nabla^2 \mathbf{u} + \mathbf{f} = -\nabla p \tag{3.2}$$

where $\mathbf{u}$ is the velocity, p the pressure, ν the kinematic viscosity and t the time. In Equation 3.2, $\mathbf{f}$ is the forcing term, or immersed boundary force, which is required to impose the correct boundary condition for the velocity at the interface with the immersed solid. This force is here strictly only active inside the solid or in its direct vicinity (first layer of fluid cells), and its formulation is discussed below.

3.1.3 Penalization of the velocity

The immersed boundary method presented in this paper is based on the Penalty method, first introduced by Arquis [8]. In this approach, the immersed solid is considered as a porous media, with a very small permeability $0 < K \ll 1$. Hence the forcing term of Equation 3.2 can be written in the following form:

$$\mathbf{f} = \chi \frac{\nu}{K} (\mathbf{u} - \mathbf{u_{ib}}) \tag{3.3}$$

where $\mathbf{u_{ib}}$ is the velocity of the Immersed Boundary (IB) and χ is a characteristic function, which makes possible the localization of the solid. This function is defined in the equation below:

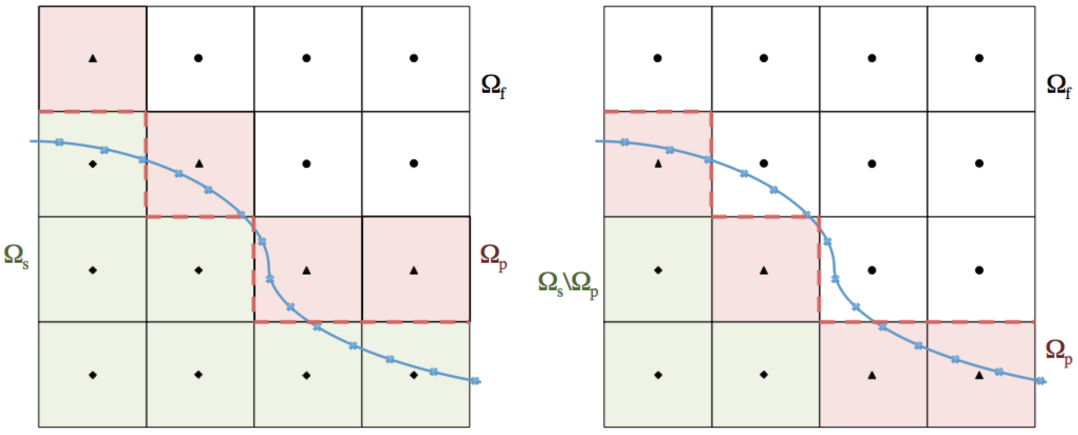
$$\chi(\mathbf{x}, t) = \begin{cases} 1 & \text{if} \quad \mathbf{x} \in \Omega_s(t) \\ 0 & \text{if} \quad \mathbf{x} \in \Omega_f(t) \end{cases} \tag{3.4}$$

In the solid domain, the forcing term reduces the momentum equation to $\mathbf{u} = \mathbf{u_{ib}}$, which satisfies the no-slip boundary condition. On the contrary in the fluid domain, the forcing term is not enabled and the original Navier-Stokes equation is thus obtained. Angot et al. have shown in [7] that this formulation is equivalent to the standard incompressible Navier-Stokes equations with a Dirichlet condition at the interface Γ for the velocity, and the authors have presented rigorous estimation of the errors introduced by this formulation as well as convergence theorems. In the cases considered in this document, the velocity of the immersed body $\mathbf{u_{ib}}$ in Equation 3.3 is always a geometrical velocity which is known and time-dependent.

3.1.4 Correction scheme for the velocity near the interface

For Cartesian grids, the forcing term presented in Equation 3.3 leads to a stair-case definition of the solid geometry as shown in Figure 3.1. Furthermore the cell centers, where the solid velocity is imposed, do not necessarily coincide with the solids points representing the immersed body. Two main errors are introduced with a simple penalty approach. Firstly, the interface orientation is not accounted for, and secondly, the Dirichlet condition is not necessarily satisfied at the exact position of the interface.

In order to address these issues, Chantalat et al. [20] have proposed for elliptic problems an iterative correction of the normal derivative of the solution at the interface. A more intuitive approach is to impose a corrected velocity in the cells near the interface, chosen to ensure that the Dirichlet condition $\mathbf{u} = \mathbf{u_{ib}}$ is satisfied at the exact interface position. Practically, this means that the corrected velocity is a function of both the solid velocity and the fluid velocity in the close neighboring area. In the Sub-Mesh Penalty Method of Sarthou et al. [111], this corrected velocity is obtained by linear interpolation between the solid and the fluid velocities, with an implicit formulation. In [60] the corrected velocity is evaluated from an averaged reconstruction of the velocity



(a) case A: Outer velocity correction

(b) case B: Inner velocity correction

Figure 3.1: Sketch of the discrete domains. The discrete solid domain is delimited by the dashed red line. ● are the cell centers in the fluid domain Ω_f. ♦ are the cell centers in the solid domain $\Omega_s \backslash \Omega_p$, except the penalized domain, in green, and ▲ are the cell centers in the penalized domain in red. x are the solid points (in blue). The grids considered in this study are made of cubic cells and uses the collocated storage of OpenFOAM®

gradient for the fluid contribution, and by means of a minimization problem for the solid contribution. The IB force is then scaled with the cell volume ratio, and the modified Navier-Stokes equations are solved with a fractional-step method, which takes into account the forcing term in the correction step as well, as suggested by [59] for a consistent imposition of the boundary condition at the interface.

In this study, the two main types of velocity correction scheme introduced above have been implemented in the OpenFOAM® framework. In addition to the imposition of the simple Dirichlet condition $\mathbf{u} = \mathbf{u_{ib}}$ inside the solid domain, a modified velocity $\mathbf{u_P^*}$ is also imposed near the interface to account for the exact position of the latter and achieve higher order of accuracy. In this context, the forcing term introduced in Equation 3.3 is re-defined as:

$$\mathbf{f}^* = \chi\frac{\nu}{K}(\mathbf{u} - \mathbf{u_{ib}}) + \chi^*\frac{\nu}{K}(\mathbf{u} - \mathbf{u_P^*}) \tag{3.5}$$

In Equation 3.5, χ^* stands for the characteristic function which represents the additional layer of cells, where the modified velocity is imposed. The new characteristic function defines a domain Ω_p, named here as the penalized domain:

$$\chi^*(\mathbf{x},t) = \begin{cases} 1 \quad \text{if} \quad \mathbf{x} \in \Omega_p(t) \\ 0 \quad \text{else} \end{cases} \tag{3.6}$$

The penalized domain can either be included in the fluid domain Ω_f or in the solid domain Ω_p. More precisely, each cell in Ω_f possessing at least one neighbor in Ω_s is added to Ω_p (case A for outer correction), and reciprocally in the other case (case B for inner correction). The two possible cases are illustrated in Figure 3.1. In the case of an internal penalized domain, the support of the characteristic function introduced in Equation 3.4 for the imposition of the non-modified solid velocity is changed from Ω_s to $\Omega_s \backslash \Omega_p$.

The estimation of the modified velocity $\mathbf{u}^*_\mathbf{P}$ is what differentiates the two types of corrections scheme. Similarly to [60], a correction based on the reconstruction of the velocity gradient is firstly detailed. As the velocity gradients are computed at the node where the correction is applied, the penalized domain needs to define an external layer of cells for the reconstructed gradients to have a physical meaning. Secondly, a correction scheme based on interpolation is introduced. This type of correction has been widely used with the direct forcing type of IBM ([9], [34], [136], ...) and penalization approach [111], with both explicit and implicit formulation. Regarding the location of the correction, both internal ([136], [112]) and external ([9], [45]) schemes have been considered previously in literature. Finally it is important to note that in all cases the momentum forcing is enabled everywhere inside the solid domain, either to impose the interpolated velocity or the solid velocity. Thereby, the implementation of the correction scheme for the velocity does not introduce a reversed fictitious velocity inside the IB, as it is the case for IBMs in which the unmodified Navier-Stokes equations are solved inside the IB [87].

Correction based on the reconstruction of the velocity gradient

The desired velocity $\mathbf{u_P^*}$ in a cell P of the penalized domain Ω_p can be derived from the velocity at the interface and the velocity gradient in the considered cell:

$$\mathbf{u_P^*} = \mathbf{u_{ib,n}} + \nabla \tilde{\mathbf{u_P}} \cdot \overrightarrow{\mathbf{X_{ib,n}P}} \tag{3.7}$$

Equation 3.7 defines a first order reconstruction, similarly to a linear interpolation. $\mathbf{X_{ib,n}}$ is the closest Lagrangian solid point to the cell center P. For the velocity $\mathbf{u_P^*}$ to account for the solid presence, the velocity gradient at the cell P must also take into account its presence. The procedure is illustrated in Figure 3.2. In OpenFOAM ® the gradient operator is computed according to the Gauss theorem. Assuming a uniform Cartesian 2D grid of perfect orthogonal quality with uniform spacing Δ, the gradient of velocity in cell P is defined as:

$$\begin{cases} \frac{d\mathbf{u_P}}{dx} = \frac{\mathbf{u_Q} - \mathbf{u_S}}{2\Delta} \\ \frac{d\mathbf{u_P}}{dy} = \frac{\mathbf{u_R} - \mathbf{u_T}}{2\Delta} \end{cases} \tag{3.8}$$

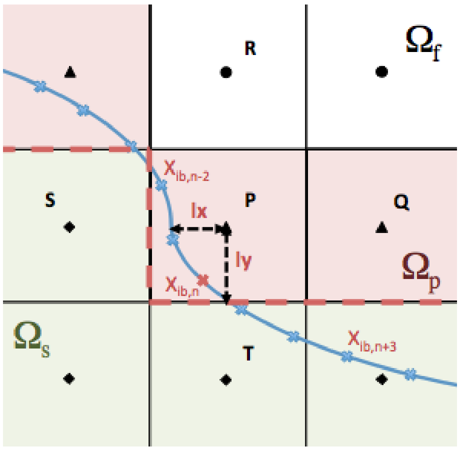


Figure 3.2: Sketch of the velocity correction scheme based on the reconstruction of the velocity gradient on a 2D Cartesian mesh. The same nomenclature is used for the domains definition.

Finally, the gradient computed without the presence of the solid as in Equation 3.8 is modified with the exact position of the interface. The new velocity gradient $\tilde{\mathbf{u_p}}$ reads then:

$$\begin{cases} \frac{d\tilde{\mathbf{u_P}}}{dx} = \frac{\mathbf{u_Q}-\mathbf{u_S}}{\Delta+l_x} = \frac{d\mathbf{u_P}}{dx}\frac{2\Delta}{\Delta+l_x} \\ \frac{d\tilde{\mathbf{u_P}}}{dy} = \frac{\mathbf{u_R}-\mathbf{u_T}}{\Delta+l_y} = \frac{d\mathbf{u_P}}{dx}\frac{2\Delta}{\Delta+l_y} \end{cases} \tag{3.9}$$

In Equation 3.9, l_x and l_y stands for the distance from the cell center P to interface in the x and y direction respectively. These parameters are found with a dichotomy algorithm and the same functions used to identify the cell centers inside or outside the solid domain. The main limitation of this correction scheme is due to the assumption that the grid is perfectly orthogonal and regular.

Correction based on direct interpolation

This correction scheme considers interpolation/extrapolation algorithms to estimate the modified velocity $\mathbf{u_P^*}$. In case A, for an outer correction, the modified velocity is calculated from a linear interpolation along the interface normal direction between the point on the interface and a virtual point in the fluid domain. The interpolation is defined in Equation 3.10. A similar procedure is employed in case B for an inner correction, which is based on a linear extrapolation as defined in Equation 3.11.

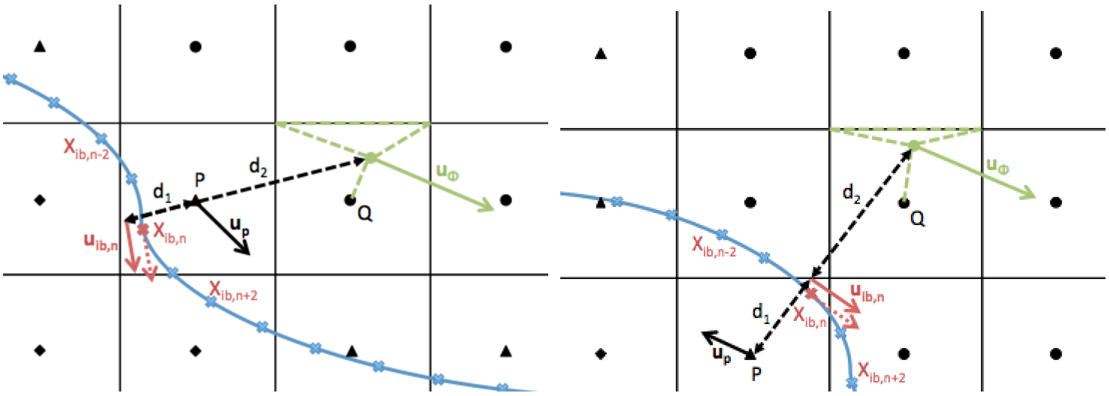
$$\mathbf{u_P^*} = \frac{d_2 - d_1}{d_2}\mathbf{u_{ib,n}} + \frac{d_1}{d_2}\mathbf{u}_\phi = \frac{d_2 - d_1}{d_2}\mathbf{u_{ib,n}} + \frac{d_1}{d_2}\sum_k \alpha_k \mathbf{u_{i,k}} \tag{3.10}$$

$$\mathbf{u_P^*} = \frac{d_1 + d_2}{d_2}\mathbf{u_{ib,n}} - \frac{d_1}{d_2}\mathbf{u}_\phi = \frac{d_1 + d_2}{d_2}\mathbf{u_{ib,n}} - \frac{d_1}{d_2}\sum_k \alpha_k \mathbf{u_{i,k}} \tag{3.11}$$

In the equations above, d_1 and d_2 represent respectively the distance from the cell center P to the interface, and the distance from the interface to a virtual point ϕ, along the interface normal. The procedure is illustrated in Figure 3.3. Similar correction schemes have been employed for instance in the IBM of Tseng & Ferziger [136] and in the Penalty method of Sarthou et al. [111]. Two types of case studies will be discussed in the present manuscript. The first considers academic problems involving immersed geometries which can be defined analytically. In this case the distance to the interface can be determined exactly and the Dirichlet condition $\mathbf{u} = \mathbf{u_{ib,n}}$ is thus satisfied at the exact interface position. The second considers more realistic engineering systems with relatively complex immersed boundaries. In which case determining the exact interface position is not possible. Two options have been considered: estimate the position (i) with a dichotomy algorithm or (ii) simply using the closest solid point $\mathbf{x_{ib,n}}$. The former iterative approach can converge reasonably quickly but the increase in the computational time is not negligible with high surface refinement. The error introduced in this case can be significantly reduced with a sufficient refinement of the solid surface mesh, and was found negligible in the test cases considered in this work. This result is discussed in Section 5.3.2. It is important to note however that this discrepancy may prevent the model achieving a formal second order accuracy in realistic applications. Regarding the accuracy of the solution as a function of the parameters d_1 and d_2, it was verified for the Wannier case (presented in Section 4.1) that $d_2 = d_1$ gives the best results. Finally, the velocity at the virtual point $\mathbf{u}_\phi$ is also obtained thanks to a linear interpolation of the velocity at the center of the cell Q, holding the virtual point, and velocities at different points k on the closest face of the considered cell. Thereby, α and β in Equation 3.11 are the coefficients of the linear combination. The velocities at the face corners $\mathbf{u_{i,k}}$ are again estimated through interpolation from the neighboring cells. For highly concave solid geometries, the virtual point ϕ may be located inside the solid. In this case, the normal to the immersed interface is rotated for the virtual point to be in the fluid domain. Although this correction is usually necessary only for a few cells, it was found to significantly denoise the results for the static impeller case (described in Section 4.4) or the pump case (described in Section 5.1).

Implicit correction based on interpolation

The Sub-Mesh Penalty Method (SMPM) proposed by Sarthou et al. [111] was also implemented for two-dimensional cases. This implicit correction scheme is based on a bilinear interpolation, which uses and combines linear 1-D interpolation in both directions.



(a) case A: Outer velocity correction

(b) case B: Inner velocity correction

Figure 3.3: Sketch of the velocity correction schemes based on interpolation/extrapolation on a 2D Cartesian mesh. The same nomenclature is used for the domains definition.

3.1.5 Estimation of the immersed boundary force

The force applied by the fluid on the immersed solid is of clear interest from an engineering point of view. The integrated values derived from this force are often used for validation purposes. This physical force **F** can be decomposed to a pressure and a viscous term, and reads:

$$\mathbf{F} = -\iint_{\Gamma} p\mathbf{dS} + \mu \iint_{\Gamma} (\mathbf{T} \cdot \mathbf{n})\mathbf{dS} \tag{3.12}$$

where **dS** is a small element of surface on the immersed boundary interface Γ, and **n** and **T** represent respectively the normal at the solid interface and the strain rate tensor. The integral terms in Equation 3.12 are estimated in two different ways in this work. Their estimation requires additional and delicate computations. Indeed, as the surface of the immersed body does not necessarily coincide with the faces of the Eulerian grid, a reconstruction of the variables at the Lagrangian solid points is usually implemented [13], [18]. This is the first method for estimating the force. The second method is based on the Gausss theorem.

Method 1: Direct integration on the solid interface

The first approach for estimating the force applied by the fluid is simply based on the discretization of Equation 3.12. The latter can be approximated in the following way:

$$\mathbf{F} = -\sum_n p_n \mathbf{S_n} + \mu \sum_n (\mathbf{T_n} \cdot \mathbf{n_n}) \mathbf{S_n} \tag{3.13}$$

The difficulties here lie in the definition of the surface elements $\mathbf{S_n}$, which are not necessarily (but usually) coincident with the spatial discretization of the fluid domain.

One way to obtain an estimate of the surface is to calculate first the volume fraction of the solid in each cell. This is done thanks to the distance function introduced in Section 3.1.1. Indeed, the volume fraction of a cell i is related to the distance function Ψ, as shown in Equation 3.14:

$$\alpha_i = \frac{1}{\Omega_i} \int_{\Omega_i} H(\Psi) d\Omega \tag{3.14}$$

In Equation 3.14, H stands for the Heaviside function. With a first order linearization of the distance function, the volume fraction in cell α_i reads for a cartesian mesh [140]:

$$\alpha_i = \begin{cases} 0 & \text{if} \quad \Psi_i < -\Psi_{max,i} \\ \frac{1}{2}\frac{(\Psi_{max,i}+\Psi_i)^2}{\Psi^2_{max,i}-\Psi^2_{mid,i}} & \text{if} \quad -\Psi_{max,i} \leq \Psi_i < -\Psi_{mid,i} \\ \frac{1}{2} + \frac{\Psi_i}{\Psi_{max,i}+\Psi_{mid,i}} & \text{if} \quad -\Psi_{mid,i} \leq \Psi_i < \Psi_{mid,i} \\ 1 - \frac{1}{2}\frac{(\Psi_{max,i}-\Psi_i)^2}{\Psi^2_{max,i}-\Psi^2_{mid,i}} & \text{if} \quad \Psi_{mid,i} \leq \Psi_i < \Psi_{max,i} \\ 1 & \text{if} \quad \Psi_i \geq \Psi_{max,i} \end{cases} \tag{3.15}$$

In Equation 3.15, the two parameters $\Psi_{max,i}$ and $\Psi_{mid,i}$ are respectively defined by Equations 3.16 and 3.17.

$$\Psi_{max,i} = \frac{1}{2} max(\Delta_x|\frac{\partial \Psi_i}{\partial x}|, \Delta_y|\frac{\partial \Psi_i}{\partial y}|, \Delta_z|\frac{\partial \Psi_i}{\partial z}|) \tag{3.16}$$

$$\Psi_{mid,i} = \frac{1}{2} min(|\Delta_x|\frac{\partial \Psi_i}{\partial x}|-\Delta_y|\frac{\partial \Psi_i}{\partial y}||, |\Delta_y|\frac{\partial \Psi_i}{\partial y}|-\Delta_z|\frac{\partial \Psi_i}{\partial z}||, |\Delta_z|\frac{\partial \Psi_i}{\partial z}|-\Delta_x|\frac{\partial \Psi_i}{\partial x}||) \tag{3.17}$$

where Δ_x, Δ_y and Δ_z are the dimensions of Cartesian grid cells.

Once the volume fraction is known, the area of the interface in each cell is given by Equation 3.18 for a finite volume approach:

$$\mathbf{S_i} = \Omega_i \nabla \alpha_i \tag{3.18}$$

Instead of being computed over the surface of the immersed solid, the sums of Equation 3.13 are then computed over Eulerian volumes in the vicinity of the immersed surface, for which each $\mathbf{S_i}$ is known. This practice of representing the discretized surfaces in each volume involves cells on both sides of the interface. By excluding the internal stress in the integration, the total area of the immersed surface is not recovered. For this reason the estimation of the forces acting on the solid is modified to take into account only the fluid cells. It is then scaled to obtain the correct surface area. The discrete force finally reads:

$$\mathbf{F} = (-\sum_{i \subset \Omega_f} p_i \mathbf{S_i} + \mu \sum_{i \subset \Omega_f} (\mathbf{T_i} \cdot \mathbf{n_i})\mathbf{S_i}) \frac{\sum_i \|\mathbf{S_i}\|}{\sum_{i \subset \Omega_f} \|\mathbf{S_i}\|} \tag{3.19}$$

With this method, the reconstructed surface is discontinuous but satisfies the constraint $\sum_i \mathbf{S_i} = 0$. The constraint $\sum_{i \subset \Omega_f} \mathbf{S_i} = 0$ is also verified with a relative error within 2% in the test cases considered in this research. As for each cell element involved in Equation 3.19, the pressure and viscous stress are evaluated at the cell center, the computation of the force is only first-order accurate. For this reason, the current approach for estimating the IB force is not suitable to analyze the order of accuracy of the overall IBM. Finally the good computational efficiency of this method should be noted, regarding the computation of the volume fraction. Indeed, in standard techniques such as described in [86], the fluid cells near the interface are decomposed in a large amount of sub-parcels. For each parcel, a searching algorithm is called to identify and distinguish those inside the solid domain from those outside. Given the complexity of such an algorithm, the computational time increases drastically for accurate estimations of the volume fraction.

Method 2: Navier-Stokes equivalence

The force $\mathbf{F}$ can also be derived from Equation 3.12 by applying the Gausss theorem to the fluid domain Ω_f surrounding the immersed body:

$$\mathbf{F} = -\iiint_{\Omega_f} \nabla p d\Omega + \iint_{\Pi} p\mathbf{dS} + \mu \iiint_{\Omega_f} \nabla^2 \mathbf{u} d\Omega - \mu \iint_{\Pi} (\mathbf{T} \cdot \mathbf{n}) \mathbf{dS} \tag{3.20}$$

where Π represents the boundaries of the domain Ω_f, excluding the immersed interface Γ (as illustrated in Figure 2.1). The volume integral terms can also be replaced by the transient and convective terms derived from the Navier-Stokes equations, as described in Equation 3.21:

$$\mathbf{F} = \rho \iiint_{\Omega_f} \frac{\partial \mathbf{u}}{\partial t} d\Omega + \iint_{\Pi} p\mathbf{dS} + \rho \iiint_{\Omega_f} \nabla \cdot (\mathbf{uu}) d\Omega - \mu \iint_{\Pi} (\mathbf{T} \cdot \mathbf{n}) \mathbf{dS} \tag{3.21}$$

Similarly, the Gauss theorem can be applied internally in the solid domain Ω_s. In this case, additional boundaries are not intervening, but the forcing term from the momentum equation (see Equation 3.2) is contributing to the estimation of the fluid force.

The order of accuracy of this method is directly governed by the orders of the schemes used to discretized the velocity and the pressure. Second order schemes in space are used to compute each term of Equation 3.21 in each cell of Ω_f. Volume integral of gradients are written as a discrete summation over the cell faces (see [61]), and the face information is obtained using second order interpolation with the neighbouring cell values. For the cells of Ω_f directly adjacent to a cell in the solid domain, this estimation of the velocity gradient does not take into account the exact position of the interface. In this case, the overall order of accuracy of this method can then be lower than the order imposed by the schemes.

3.2 Discretization techniques and solution solver

The discretization procedure for the Navier-Stokes equations are based on the work of Jasak [61], following a Finite Volume formulation. All the variables are stored at the cell centers, where a non-staggered grid arrangement is used. The procedure of Jasak uses a predictor-corrector algorithm for the pressure-velocity coupling (PISO scheme), whose formulation is detailed below with combination of a pressure correction accounting for the immersed boundary presence. The equations are discretized in time with an Euler implicit scheme, using second order schemes in space.

3.2.1 Momentum predictor

The first step of the method consists in estimating an intermediate velocity field, thanks to a semi-discretized momentum equation:

$$a_i \mathbf{\tilde{u}_i} = \mathbf{H}(\mathbf{\tilde{u}}) - \nabla p^n \tag{3.22}$$

In order to avoid a checker-boarding effect, the pressure gradient is not discretized at this stage [107]. In the matrix system 3.22, $\mathbf{\tilde{u}_i}$ represents the numerical estimation of the intermediate velocity at the center of the i-th cell, and a_i is the diagonal matrix coefficient for the cell i. The term $\mathbf{H}(\mathbf{\tilde{u}})$, includes the contribution from neighbouring cells, the explicit part of the transient term, and all other source terms. The forcing term f introduced in Equation 3.2 is divided in an implicit contribution, involving the fluid velocity, and an explicit contribution, involving the solid or corrected velocity. The implicit part thus directly modifies the diagonal matrix coefficients a_i, while the explicit part is included in the term $\mathbf{H}(\mathbf{\tilde{u}})$. The term $\mathbf{H}(\mathbf{\tilde{u}})$ finally reads for a cell inside the solid domain:

$$\mathbf{H}(\mathbf{\tilde{u}}) = -\sum_j a_j \mathbf{\tilde{u}_j} + \frac{\mathbf{u_i^n}}{\Delta t} + \frac{\mathbf{u_{ib}}}{K} \tag{3.23}$$

If one considers a cell inside the domain Ω_p, the solid velocity $\mathbf{u_{ib}}$ is replaced by the corrected velocity $\mathbf{u_p^*}$ in Equation 3.23. In the latter, j stands for the neighbouring cells,

n represents the previous iteration time step and Δt is the value of the time step. The contribution of the neighbouring cells for the term $\mathbf{H}(\tilde{\mathbf{u}})$ is obtained by a linearization of the quadratic convection term, which implies that a flux satisfying the mass conservation equation at time n is used to calculate the matrix coefficients a_i and a_j.

3.2.2 Pressure solution and PISO loop

The second step of the discretization procedure is to solve the pressure equation. This equation is derived from the momentum Equation 3.22, by imposing mass conservation ($\nabla \cdot \mathbf{u} = 0$). Gradient and divergence terms are discretised using the Gauss theorem, providing an equation for pressure:

$$\sum_f \mathbf{S_f}.[(\frac{1}{a_i})_f(\nabla p)_f] = \sum_f \mathbf{S_f}.(\frac{\mathbf{H}(\mathbf{u}^m)}{a_i})_f \tag{3.24}$$

where $\mathbf{S_f}$ is the outward normal surface vector for the face f of the ith cell. The resulting matrix for pressure is solved with an iterative method. One can note that the pressure does not depend directly on the intermediate velocity field, but only on its contribution through the term $\mathbf{H}(\mathbf{u}^m)$. Here, m stands for the increment of the PISO loop, whose initial conditions are given by $\mathbf{u}^m = \tilde{\mathbf{u}}$ and $p^m = p^n$ at the initial iteration $m = 0$.

The new face flux and cell center velocity are calculated from the new pressure field. Equation 3.25 gives the new conservative fluxes, which will be used to build the matrix system for the momentum predictor at the next time step. The velocity is corrected explicitly with Equation 3.26. The value at the faces are obtained using interpolation techniques.

$$F^{m+1} = \mathbf{S_f}.\mathbf{u_f}^{m+1} = \mathbf{S_f}.[(\frac{\mathbf{H}(\mathbf{u}^m)}{a_i})_f - (\frac{1}{a_i})_f(\nabla p^m)_f] \tag{3.25}$$

$$\mathbf{u_i}^{m+1} = \begin{cases} \frac{\mathbf{H}(\mathbf{u}^m)}{a_i} & \text{if} \quad i \in \Omega_s \cup \Omega_p \\ \frac{\mathbf{H}(\mathbf{u}^m)}{a_i} - \frac{1}{a_i}\nabla p^m & \text{else} \end{cases} \tag{3.26}$$

The velocity correction step, defined by Equation 3.26 to account for the corrected pressure field, is modified in the solid and the penalized domains. As the intermediate velocity estimated from the momentum predictor step does satisfy the boundary condition determined by the momentum forcing, this velocity correction step is not necessary in the latter domains. Because the velocity correction step is explicit, several iterations of pressure solutions and explicit velocity corrections are necessary (usually two or three) to take into account the transported influence of the corrections of neighbouring velocities. After solving the velocity correction step for $\mathbf{u^{m+1}}$, the term $\mathbf{H}(\mathbf{u^m})$ is recalculated and a new increment is performed as $m \rightarrow m+1$. Once a predetermined convergence criterion is achieved, velocity and pressure are updated for the next time step with $\mathbf{u^{n+1}} = \mathbf{u^{m+1}}$ and $p^{n+1} = p^m$.

Finally, it is worth noting that the Rhie-Chow discretization procedure is adopted to avoid checker-boarding on the collocated grid [107]. The cell face fluxes are interpolated from the momentum equation (Equation 3.25) and then used to derive the pressure correction equation (Equation 3.24) from the continuity equation.

3.2.3 Boundary condition for pressure at the immersed interface

For continuous IBM like the penalization approach presented here, no pressure boundary condition is usually needed on the immersed boundary [92]. However, there is no guarantee that the pressure inside the body has a physical meaning. Indeed, the velocities obtained from interpolation (see Section 3.1.4) generate a reversed flow [87], if the solid velocity is not imposed everywhere inside the solid domain, and results in an unphysical flux over IB in any case. Problems with the mass conservation arise then in the interior and boundary cells. This issue has been considered in detail by Kang et al. [66] for a discrete IBM, where a decoupling constraint for the pressure is proposed, and combined with a compatible interpolated velocity boundary condition related to mass conservation. Kim et al. [73] developed a method introducing a mass source/sink in addition to the momentum forcing, in order to satisfy the continuity in the cells containing the immersed boundary.

In this study, no artifice is introduced to satisfy mass conservation. Instead, two types of methods have been investigated to avoid the effect of spurious pressure inside the domain. The two types of approaches are detailed below.

Pressure Decoupling

The decoupling is carried out during the construction of the pressure matrix, where the contribution of the faces between the fluid and the solid domains are removed from Equation 3.24. Hence, the pressure equation becomes:

$$\sum_f \chi_f \mathbf{S_f}.[(\frac{1}{a_i})_f(\nabla p^m)_f] = \sum_f \chi_f \mathbf{S_f}.(\frac{\mathbf{H}(\mathbf{u}^m)}{a_i})_f + \sum_f \bar{\chi_f} \mathbf{S_f}.\mathbf{u_{ib,f}} \tag{3.27}$$

In Equation 3.27, χ_f stands for the face characteristic function, and $\bar{\chi_f}$ its opposite, which identifies the faces between the fluid domain Ω_f and the solid domain Ω_s:

$$\chi_f = \begin{cases} 0 \quad \text{if} \quad f \in \Omega_f, f \in \Omega_s \\ 1 \quad \text{else} \end{cases} \tag{3.28}$$

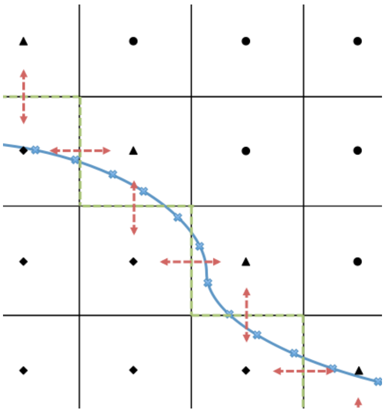


Figure 3.4: Sketch of the pressure decoupling method used for the pressure equation step. The faces where the correction is applied are represented in green. The red arrows symbolize the cancellation of the pressure gradient a those faces.

As a result, in a given domain, pressure from the other domain does not contribute to the resolution of the pressure field. Furthermore, the flux at those specific faces is approximated with the immersed boundary velocity $\mathbf{u_{ib,f}}$ estimated at the face. The flux correction is finally modified in a similar way, again for the faces between the solid and fluid domains:

$$F_f^{m+1} = \begin{cases} \mathbf{S_f}.\mathbf{u_{ib,f}} \quad \text{if} \quad f \in \Omega_f, f \in \Omega_s \\ \mathbf{S_f}.\mathbf{u_f^{m+1}} = \mathbf{S_f}.[(\frac{\mathbf{H}(\mathbf{u}^m)}{a_i})_f - (\frac{1}{a_i})_f(\nabla p^m)_f] \quad \text{else} \end{cases} \tag{3.29}$$

The principle of the method is illustrated in Figure 3.4 for an outer velocity correction. The main limitation of this approach lies in the fact that the information about the interface orientation is lost for the calculation of pressure. Furthermore, the faces where the correction of the flux is applied are not necessarily inside the solid. This technique leads then to a first order approximation of the interface.

Pressure Penalization

Another way to correct the pressure in the vicinity of the interface, is to penalize the latter, similarly to the velocity penalization method, in order to force the desired Neumann condition at the boundary ($\frac{\partial p}{\partial \mathbf{n}}|_\Gamma = 0$).

The most intuitive approach is then to impose the pressure in a solid cell close to the interface to be equal to the pressure at a virtual point along the normal. A sketch of this method is shown in Figure 3.5(a). The pressure at the virtual point p_ϕ^n is obtained by linear combination, similarly to the velocity corrections introduced earlier:

$$p_P^m = p_\phi^n = \sum_k \alpha_k p_k^n \tag{3.30}$$

This method uses then an explicit Dirichlet constraint to satisfy the Neumann condition. As the method is explicit, the estimation of the corrected pressure p_P^m for the PISO iteration m relies on the surrounding fluid pressure obtained at the completed time step n. The location of the virtual point ϕ is obtained using a scalar Δ (see Figure 3.5(a)) which is adjusted to the cells dimension. It is important to note that the estimation of the pressure at the points k, relying on interpolation with neighbouring cells, can sometimes involve the pressure in cell P among others. This means that the corrected pressure in cell P uses the value of the pressure in the same cell. In order to obtain a consistent correction, several PISO iterations are thus necessary. Similar to for the velocity reconstruction scheme described in Section 3.1.4, the virtual point ϕ is moved when it lies in the solid domain for highly concave solid geometries.

Equation 3.30 can be integrated in the pressure matrix system given by Equation 3.24. The matrix line referring to cell P reads then:

$$[a_p + \frac{1}{\epsilon}]p_P^m + \sum_N \alpha_N p_N^m = S_P + \frac{1}{\epsilon}p_\phi^n \tag{3.31}$$

where a represent the matrix coefficients, N stand for all the neighbours of cell P, and S is the source term, including the fluxes, of the pressure equation. Equation 3.30 is satisfied if the penalization coefficient ϵ is small enough. A relevant choice for this coefficient is related to the magnitude of the source term and the matrix coefficients.

A more advanced approach is to obtain the Neumann condition in an implicit way. The Neumann condition for the cell p can be decomposed in the two directions in 2D, as written in Equation 3.32.

$$\frac{\partial p}{\partial \mathbf{n}}|_p = n_x \frac{\partial p}{\partial x}|_p + n_y \frac{\partial p}{\partial y|_p} = 0 \tag{3.32}$$

where $\mathbf{n} = (n_x, n_y)$ is the interface normal for the cell P. For a Cartesian mesh, Equation 3.32 can be discretized in the following way:

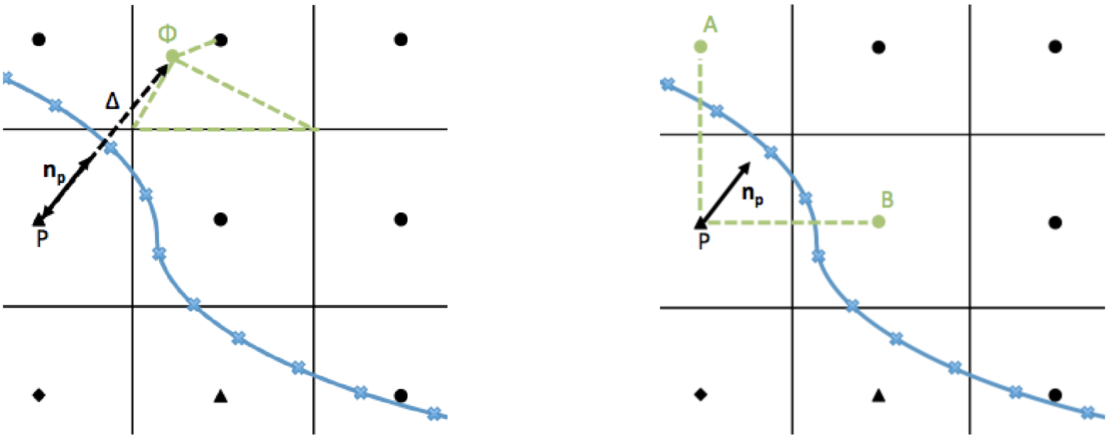
$$\frac{n_x}{\Delta_x}(p_B - p_p) + \frac{n_y}{\Delta_y}(p_A - p_p) = 0 \tag{3.33}$$

In Equation 3.33, A and B stand for the two neighbouring cells of P, as detailed in Figure 3.5(b), while Δ_x and Δ_y are the cell dimensions. The relation between the pressure in cell P and the pressure in the neighbouring cells, given by Equation 3.33, can be incorporated in the pressure matrix system with a penalization constraint. The modified pressure equation finally becomes for cell P:

$$[a_p + \frac{1}{\epsilon}(\frac{n_x}{\Delta_x} + \frac{n_y}{\Delta_y})]p_P^m + \sum_N \alpha_N p_N^m + \frac{1}{\epsilon}\frac{n_x}{\Delta_x}p_B^m + \frac{1}{\epsilon}\frac{n_y}{\Delta_y}p_A^m = S_P \tag{3.34}$$

where it should be noted that A and B are included in the neighbours N of cell P. One can note however that for a cell P having only one neighbour in the fluid domain, Equation 3.33 reduces to $p_P = p_A$ or $p_P = p_B$. In a non-negligible number of cells, the information about the interface orientation is then lost, in comparison to the explicit approach.

The extension of both explicit and implicit schemes is straightforward for three-dimensional cases.



(a) Explicit correction

(b) Implicit correction

Figure 3.5: Sketch of the pressure penalization methods used to satisfy the Neumann condition for the pressure equation step.

3.3 Turbulence modelling

From an engineering point of view, a RANS model for turbulence can be sufficient to give a good estimation of a general integrated quantity such as the head in a centrifugal pump. However, such RANS turbulence modelling like the $k - \omega$ SST model or the Spalart-Allmaras (SA) closure model, developed to improve modelling accuracy in bounded flow with adverse pressure gradients, are known to under-predict Reynolds stresses in the shear layer after boundary layer separation and as a result to delay flow recovery downstream of obstacles (e.g. a periodic hill [88] or a backward facing step [117]). Scale resolving simulations such as for example Large Eddy Simulation (LES) models have been shown to produce more realistic and accurate predictions for this type of separated turbulent flows. Furthermore, scale resolving methods are able to capture significant turbulent features of the flow, which are of prime importance for the transport of a secondary phase for instance.

The idea behind the LES is to separate the large scales of turbulence, where large eddies tend to be case and geometry-dependent, from the small scales, where small eddies tend to be isotropic and universal. The separation is achieved by applying a spatial filtering operator. Thereby, the larger scales are fully resolved, while the effect of the smaller ones is taken into account as a dissipative process, thanks to a "sub-grid" turbulent viscosity. For an accurate LES simulation, the filter, which is in general simply the space discretization, must be thus situated in the universal equilibrium range, where the eddies are mainly isotropic and the turbulence is mainly dissipative. From this, one can deduce that the major problem of LES is related to the treatment of the flow near walls. Indeed, as the scale of the turbulence tends to 0 in the immediate neighborhood of a wall, the refinements necessary for a correct separation between grid and sub-grid scales would soon become prohibitive. In this context, Detached-Eddy Simulation (DES) [98] types of model have proved to be good alternatives for such problems of computational cost. These hybrid RANS/LES models take the advantage of the RANS modeling for the wall layer, while outside this region the LES is still resolving all the scales of turbulence up to the filter size.

The IBM presented here has been coupled to a DES turbulence model. The relative simplicity and robustness of the SA model makes it a good candidate for the RANS approach. In addition, the similarities of modelling between the S-A model and a standard LES formulation for the turbulent viscosity simplifies the coupling of the RANS and LES models. The hybrid models are becoming more and more attractive due to their simplicity and their good prediction of high Reynold flows with large-geometry-induced separation regions [72], [113]. However, their performance on internal wall-bounded flows is not yet clear in general, although much research is going on [152]. The combinations of our IBM with the SA and the DES models are respectively discussed in sections 3.3.1 and 3.3.2.

3.3.1 IBM formulation of the Spalart-Allmaras model: pSA

The standard Spalart-Allmaras model assumes a zero Dirichlet condition on a wall surface for the turbulent viscosity ν_t [4]. In order to satisfy this boundary condition on the immersed boundary, the standard SA model has been modified to take into account the presence of the immersed body, using a a penalization method similar to the momentum equation. The Dirichlet condition is thus imposed for the turbulent viscosity inside the whole solid. Furthermore, for a better representation of the immersed interface, a corrected viscosity is imposed in its vicinity, similarly again to the correction scheme of the velocity.

In OpenFOAM®, the SA model requires the solution of the transport equation for a modified kinetic viscosity $\tilde{\nu}$. The turbulent viscosity is obtained from the latter according to:

$$\nu_t = \tilde{\nu} f_{v1} \tag{3.35}$$

where:

$$\begin{cases} f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \\ \chi = \tilde{\nu}/\nu \end{cases} \tag{3.36}$$

The penalized version of the transport equation for the modified turbulent viscosity finally reads:

$$\frac{D\tilde{\nu}}{Dt} = P - D + \frac{1}{\sigma}(\nabla \cdot (\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}(\nabla\tilde{\nu})^2) + \frac{\chi}{K_{\tilde{\nu}}}(\tilde{\nu} - \nu_{ib}) + \frac{\chi^*}{K_{\tilde{\nu}}}(\tilde{\nu} - \nu_p^*) \qquad (3.37)$$

In Equation 3.36 and Equation 3.37, P and D stands for the production and destruction terms which are implemented as defined by Spalart & Allmaras in [120], while c_{v1} and c_{b2} are constants of this model, and finally σ is the turbulent Prandtl number. The production term P involves the distance to the wall d_w, which is corrected to also take into account the distance to an immersed wall. Regarding the imposition of the boundary condition, ν_{ib} stands for the zero immersed body viscosity, while ν_p^* is the corrected viscosity taking into account the position and the orientation of the interface. As for the penalization of the velocity, χ, χ^* and K_{ν_t} denote here the characteristic functions and the penalization coefficient associated to the turbulent viscosity. The corrected viscosity ν_p^* is estimated by an interpolation between ν_{ib} at the position of the interface and the value of the turbulent viscosity in the neighboring fluid area, similar to the algorithm used for the penalized velocity in Equation 3.10. A similar type of correction has been used by Balaras in [9].

As for any RANS model, relevant wall functions should be used at walls when the grid refinement is not sufficient to resolve the viscous sub-layer. In our treatment of the immersed wall, no wall function is taken into account to mimic the effect of the boundary layer. This limitation of the model could be an issue for flow where the boundary layer is playing a major role, or when the location of the transition from laminar to turbulence is of interest.

3.3.2 IBM - DES

In the DES model, the transition from RANS to LES is triggered by the following parameter:

$$\tilde{d} = min(d_w, C_{DES}\Delta) \qquad (3.38)$$

In Equation 3.38, d_w stands for the distance to the wall, C_{DES} is a constant usually set to 1 and Δ defines the grid spacing as $\Delta = (\Delta x \Delta y \Delta z)^{1/3}$. When $d_w \leq \Delta$ the models acts as the SA model, while for $\Delta \leq d_w$ it acts as a LES model.

The IBM of this work is then combined with the DES model by merely changing d_w to d_w^* as detailed in Equation 3.39.

$$d_w^* = min(d_w, \Psi) \tag{3.39}$$

where Ψ is the distance function to the immersed boundary. Therefore, the model is acting as a SA model in the regions near the walls of the immersed solid.

3.3.3 Turbulence modelling in OpenFOAM

OpenFOAM® relies on the Boussinesq approximation and the definition of the turbulent viscosity to estimate the Reynolds stresses, whether it is with a RANS approach or a LES approach considering the sub-grid scale stresses. With the Boussinesq approximation, the momentum conservation equation becomes:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) - \nabla(\nu_{eff} \nabla \cdot \mathbf{u}) - \nabla(\nu_{eff} dev[(\nabla \mathbf{u})^T]) + \mathbf{f} = -\nabla p \tag{3.40}$$

In Equation 3.40, $\nu_{eff} = \nu + \nu_t$ is the effective kinematic viscosity, and dev is the deviatoric operator, which transforms any tensor $\mathbf{T}$ in $dev(\mathbf{T}) = \mathbf{T} - \frac{1}{3}tr(\mathbf{T})\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. An increased apparent viscosity can reproduce the effect of turbulence such as energy dissipation and transport. The equations used to compute the turbulent viscosity are what differentiates the RANS models.

For near-wall modelling, wall functions can be used to correctly estimate the wall shear stress if the grid resolution is not sufficient to reach the viscous part of a boundary layer. The wall functions provide a boundary condition for the turbulent viscosity equations. In the body-fitted simulations presented in this research, Spalding's law [122] is used. It is an all-y^+ wall function, which was developed for attached boundary layers.

Local friction velocity and y^+ are obtained iteratively using a Newton-Raphson method, and are then used to compute the local turbulent viscosity. For IBM simulations, the boundary condition for the turbulent viscosity is provided by the penalization of the Spalar-Allmaras equation, as described in Section 3.3.1.

3.3.4 Velocity correction scheme in a turbulent context

Given the nature of IBMs, which do not rely on inflation layers at walls, it can be assumed that cells adjacent to immersed boundaries will typically be placed in the logarithmic layer for highly computationally demanding 3D turbulent cases. This does not comply with a linear reconstruction of the velocity as introduced in Section 3.1.4. On the contrary, power-law based types of reconstruction can approximate logarithmic profiles. However, such profiles are only present provided that the flow remains attached. The consistence of power-law formulation for the turbulent rotating flows of interest in this research, involving high mixing and separations and no clear attached boundary layers, is thus questionable. It was thus decided that the sensitivity of flow simulation on the type of velocity reconstruction scheme should be investigated.

A 1/7 power law was firstly introduced by Werner and Wengle [149] and was reported to perform best among several wall models for the simulation of separated flow in a channel in [130]. Power-law types of corrections have been proposed also by Choi et al. [21] and Chang et al. [19] for turbulent flows. In [19], Chang et al. have considered and compared both linear and power-law based reconstruction of the velocity. Their results suggest that the former method gives slightly better results, and the authors argue that this is most likely due to the high level of grid refinement used, which is fine enough to reach the viscous sub-layer in some areas. In this research, the reconstruction is formulated similarly to the *tangency correction* of Choi et al. [21], by decomposing the velocity into its tangential and normal components relatively to the immersed surface: $\mathbf{u_P^*} = \mathbf{u_T} + \mathbf{u_N} + \mathbf{u_{ib,n}}$. In this formulation, $\mathbf{u_T}$ and $\mathbf{u_N}$ are the components of the fluid velocity relative to the wall velocity. The dependence of the tangential component $\mathbf{u_T}$ to the normal distance to the wall is assumed to follow a power-law function. In this case, the tangential velocity at the cell center P can be derived from the tangential velocity at a virtual point ϕ as:

$$\mathbf{u_T} = (\frac{d_1}{d_2})^k \mathbf{u_{\phi,T}} \tag{3.41}$$

where small values of k (usually 1/7) can approximate the logarithmic distribution expected in the near wall region for an attached turbulent flow. The definition of d_1, d_2 and $\mathbf{u}_\phi$ are identical to Section 3.1.4. The normal component of the velocity $\mathbf{u_N}$ is assumed to follow a linear distribution, i.e. using $k = 1$. The reconstruction of the tangential velocity component is illustrated in Figure 3.6.

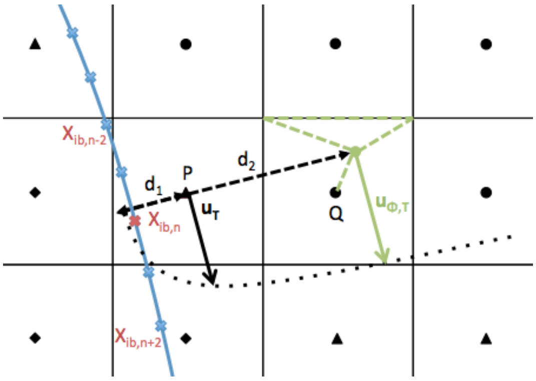


Figure 3.6: Sketch of the velocity correction scheme based on a power-law used for the turbulent case. The same nomenclature as previous is used for the cells definition.

The sensitivity to this type of velocity reconstruction in comparison to the linear scheme is discussed in Section 5.3.5 for the mixer case.

Chapter 4

Results and discussion: Validation of xIBM for laminar flows

The current IBM has been validated against several test cases, covering different types of solid geometries and flow regimes. The improved penalization approach with corrections schemes is compared to a simple first order penalization approach, where no treatments are applied for velocity and pressure boundary conditions on the immersed surface. Numerical solutions were also computed on equivalent body-fitted grids for references. For all validation cases, the IBM solutions are obtained on Cartesian grids of perfect orthogonal quality. In addition, the penalization coefficients are the subject of a convergence study to confirm independence of the results with regards to the penalization coefficients. In the standard approach, the body-fitted grid is mainly Cartesian as well, and based on the same mesh size as the IBM grid, as illustrated in Figure 4.1 for the Wannier case studied in Section 4.1. Then, in order to fit the Cartesian cells to the solid geometry, a so-called "Cut-Cell" method is used. Finally, an additional layer of identically sized boundary conforming cells is added on the surface of the solid, without any inflation layers. This was implemented in an attempt to minimize differences between the IBM and the body-fitted approaches. The computations were performed up to a residuals of 10^{-5} for both velocity and pressure. In OpenFOAM®, the residuals are normalized with the volume-weighted central coefficients from the discretized transport equation of the considered variable (where Upwind Differencing is used on the convection term) [61]. If not stated otherwise, second order schemes are used for the space discretization in all the test cases presented below, for both body-fitted methods and IBMs. The second order Gaussian integration is used for gradient and laplacian operators and a bounded Self-Filtered Central Differencing (SFCD) scheme is used for the discretization of the divergence. Regarding the time discretization, the first order Euler scheme is considered, except for the Wannier case which uses a second-order Crank-Nicholson scheme.

All types of reconstruction schemes for velocity and pressure introduced in the previous chapter are assessed and compared in terms of accuracy for different test cases. The following correction schemes are considered in this research to improve the imposition of the Dirichlet condition for the velocity at the immersed interface:

- the external (outer) velocity correction based on linear interpolation of the velocity in the outer layer of fluid cells adjacent to the interface, with an explicit formulation

- the external (outer) velocity correction based on bilinear interpolation of the velocity in the outer layer of fluid cells adjacent to the interface, with an implicit formulation (similar to the SMPM developed by Sarthou et al. [111])

- the external (outer) velocity correction based on the reconstruction of the velocity gradient in the outer layer of fluid cells adjacent to the interface, with an explicit formulation

- the internal (inner) velocity correction based on linear extrapolation of the velocity in the inner layer of solid cells adjacent to the interface, with an explicit formulation

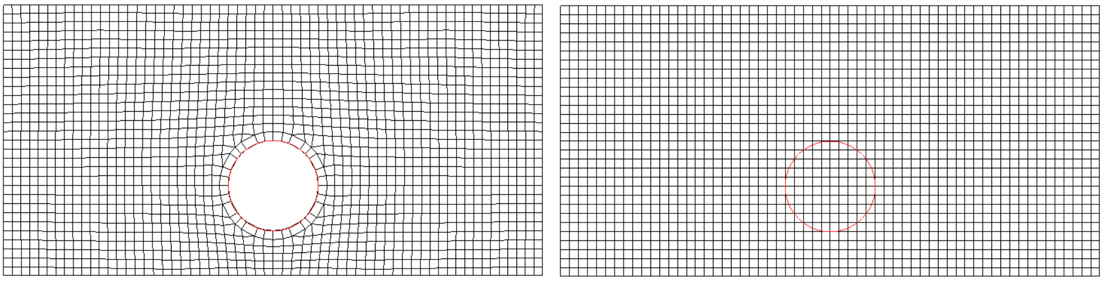
As confirmed in this chapter, very few differences are identified between the solutions generated by the three types of outer velocity reconstruction. For this reason, most of the research presented here focused on the first external correction scheme and its internal equivalent, both based on direct linear interpolation/extrapolation of the velocity. Regarding the pressure correction at the immersed interface, the schemes developed in this research are the following:

- the pressure decoupling scheme in order to exclude the contribution of the pressure inside the solid domain to the solution inside the fluid domain
- the explicit pressure penalization in order to satisfy a Neumann boundary condition
- the implicit pressure penalization in order to satisfy a Neumann boundary condition

For the two penalization approaches, the pressure is corrected internally (inner layer of solid cells adjacent to the interface).

4.1 Accuracy study in a Wannier flow

The order of accuracy of our penalty based IBM is first verified with a simple 2D flow in the Stokes regime. The analytical solution of a flow around a circular cylinder, in the vicinity of a moving wall, has been derived by Wannier [148], for very low Reynolds number. The existence of an analytical solution allows a precise comparison with the numerical estimations. The case considers a cylinder of diameter D, centred in a domain of length $6D$ and height $3D$. The dimensions of the problems are presented at the correct scale in Figure 4.1. The bottom wall, located at a distance of $0.5D$ to the bottom of the cylinder, is a moving wall with a constant velocity $U = 1m.s^{-1}$. On all other boundaries the velocity also satisfies a Dirichlet condition with the analytical solution. A zero gradient condition is imposed for the pressure at every boundary, except at the top wall for the sake of symmetry, where a Dirichlet condition is used.



(a) Body-fitted

(b) IBM

Figure 4.1: Sketch of the grids used for the 2D Wannier test case, with 10 cells in the cylinder diameter for the IBM, and the equivalent body-conforming mesh. The surface mesh of the cylinder is represented in red.

4.1.1 General results

As an analytical solution is known, the error between the numerical and analytical solutions can provide a good estimation of the accuracy of the method. The latter is computed through the norms $\mathcal{L}_1$ and $\mathcal{L}_2$ as defined in Equations 4.1 and 4.2. An averaging of the relative error $\bar{E}_{r,k}$ is also estimated with Equation 4.3.

$$E_{\mathcal{L}_1,k} = \frac{1}{N_c}\sum_i |u_{i,k} - v_{i,k}| \tag{4.1}$$

$$E_{\mathcal{L}_2,k} = (\frac{1}{N_c}\sum_i |u_{i,k} - v_{i,k}|^2)^{0.5} \tag{4.2}$$

$$\bar{E}_{r,k} = \sum_{i\subset\tilde{\Omega}}(\frac{|u_{i,k} - v_{i,k}|}{|v_{i,k}|})s_i / \sum_{i\subset\tilde{\Omega}} s_i \tag{4.3}$$

where $\mathbf{u_i}$ and $\mathbf{v_i}$ stands respectively for the numerical and analytical solutions computed at the center of the ith cell, k is the velocity component, and N_c is the total number of grid cells. The norms $\mathcal{L}_1$ and $\mathcal{L}_2$ represent global values computed on the whole domain, while the relative error $\bar{E}_{r,k}$ is averaged on a square domain $\tilde{\Omega}$ of size $0.6D$ around the immersed object, and provides an estimate of the local error. In Equation 4.3, s_i stands for the surface elements inside $\tilde{\Omega}$. The norms $\mathcal{L}_1$ and $\mathcal{L}_2$ of the errors, obtained with different models, are listed in Table 4.1 and Table 4.2 respectively, for both velocity components. The orders of the different models are estimated using the Richardson method [108]. This method is based on the ratios of the considered values obtained for three different grid refinements. Grid 1, Grid 2 and Grid 3 considered here, include respectively 10, 20 and 40 cells in the cylinder diameter. The local errors $\bar{E}_{r,k}$ are listed in Table 4.3 for the finest grid. All correction schemes for the velocity and the pressure at the immersed interface are considered in this test case. In the three tables, DI stands for the scheme based on Direct Interpolation as described in Section 3.1.4 and VG stands for the scheme based on the reconstruction of the Velocity Gradient as described in Section 3.1.4.

From the three tables, it appears that the simple penalization approach increases the global and local errors by comparison with the body-fitted solution by a factor of 2 and 10 respectively. Also, the order of the simple penalty model is only about 0.5, while the body-fitted approach leads to an order close to 1. The relatively low order obtained with the body-fitted solution may result from the skewness of the few cells making the transition between the inflation layer and the hexahedral cells. Both internal and external type of velocity penalization bring the global errors to a similar level than the body-fitted case, as shown in Tables 4.1 and 4.2. In addition, the orders of accuracy are increased for the models with corrections, and reach 2 for the norm $\mathcal{L}_2$. It is yet interesting to note that those orders are more disparate (between 1.5 and 2) than the orders of body-fitted and simple penalty models. Finally, the fact that the corrected IBMs achieve a larger order than the ones obtained with the body-fitted approach for a fairly similar error on the fine mesh, can be explained by the lower accuracy achieved by all IBMs on the coarsest mesh, but also by the higher quality of the mesh used with IBMs as noted above.

Regarding the local errors, the results listed in Table 4.3 show that the errors for the simple penalty method and the penalty method with inner velocity correction are almost one-order of magnitude higher than the others. This can be due to the fact that, in both cases, the velocity in the fluid domain near the interface is obtained from the standard Navier-Stokes equation, while for the outer velocity correction, the velocity in the first layer of fluid cells is obtained from the momentum equation modified by the IBM forcing which results directly in $\mathbf{u} = \mathbf{u}^*_\mathbf{P}$, $\mathbf{u}^*_\mathbf{P}$ being close to the analytical solution. It should also be noted that the local error estimates the error relative to an exact analytical solution which is very close to zero as the cell centre approaches the immersed surface.

The use of a pressure decoupling scheme does not seem to have a significant impact on the results. Similar results have been found with the use of a pressure penalization scheme combined with the outer velocity correction. The only case where the pressure correction is meaningful is when an inner penalization of the pressure is combined with an inner penalization of the velocity, reducing the local error to a similar order of magnitude to the other corrections (see Table 4.3). This improvement shows the importance of having a correct pressure gradient at the interface for the Wannier case, when the velocity is not directly imposed.

Model	$E_{\mathcal{L}_1,x}$	$E_{\mathcal{L}_1,y}$	**order in x**	**order in y**
Body-fitted	0.00104	0.00057	0.9	1.1
simple Penalty	0.00251	0.00121	0.4	0.6
Penalty with outer velocity correction DI	0.00087	0.00071	1.5	1.4
... and pressure decoupling	0.00086	0.00063	1.2	1.6
Penalty with outer velocity correction VG	0.00087	0.00076	1.5	1.3
Implicit Penalty with outer v. correction (SMPM)	0.00086	0.00061	1.0	1.5
Penalty with inner velocity correction	0.00086	0.00058	0.6	1.1
... and inner explicit pressure penalization	0.00090	0.00051	1.4	1.4

Table 4.1: Norm $\mathcal{L}_1$ of the errors obtained with the fine grid (Grid 3), and the associated order of accuracy, for both velocity components.

Model	$E_{\mathcal{L}_2,x}$	$E_{\mathcal{L}_2,y}$	**order in x**	**order in y**
Body-fitted	0.00148	0.00078	0.9	1.2
simple Penalty	0.00379	0.00166	0.4	0.6
Penalty with outer velocity correction DI	0.00118	0.00097	1.8	1.5
... and pressure decoupling	0.00122	0.00092	2.1	1.7
Penalty with outer velocity correction VG	0.00117	0.00101	1.7	1.4
Implicit Penalty with outer v. correction (SMPM)	0.00125	0.00088	2.0	1.6
Penalty with inner velocity correction	0.00132	0.00115	1.7	1.3
... and inner explicit pressure penalization	0.00142	0.00079	1.5	1.4

Table 4.2: Norm $\mathcal{L}_2$ of the errors obtained with the fine grid (Grid 3), and the associated order of accuracy, for both velocity components.

Model	$\bar{E}_{r,x}$	$\bar{E}_{r,y}$
Body-fitted	0.041	0.037
simple Penalty	0.403	0.293
Penalty with outer velocity correction DI	0.101	0.030
... and pressure decoupling	0.098	0.030
Penalty with outer velocity correction VG	0.096	0.035
Implicit Penalty with outer v. correction (SMPM)	0.087	0.044
Penalty with inner velocity correction	0.840	0.764
... and inner explicit pressure penalization	0.134	0.086

Table 4.3: Local averaged relative error $\bar{E}_{r,k}$ obtained with the fine grid (Grid 3), for both velocity components.

The results presented here show that the different types of outer correction schemes for the velocity lead to similar solutions. This is visible on the three tables for both the magnitude of the errors and the orders of accuracy. The same conclusion was drawn for cases involving moving boundaries. For this reason, the remainder of the research is focusing mainly on the inner and outer correction schemes based on direct extrapolation and interpolation respectively. Regarding the pressure correction, results from the decoupling scheme are also not shown in the next sections, given the low influence of the scheme on the solutions of all the test cases considered in this research.

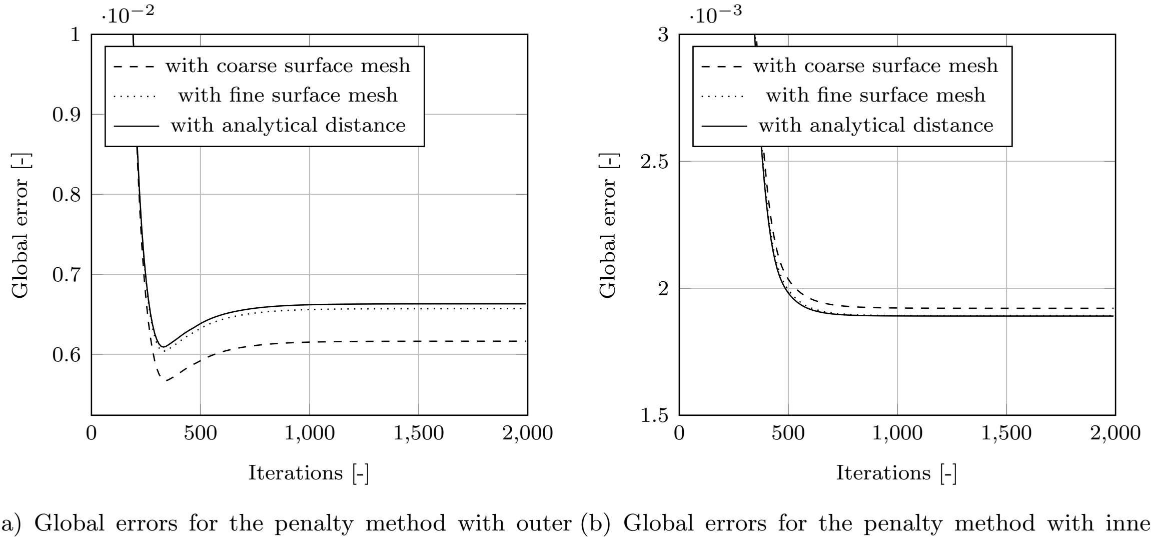
4.1.2 Sensitivity to the boundary conditions

The results of the Wannier case have been found to be highly sensitive to the boundary conditions. Imposing a Dirichlet velocity condition on all boundaries over-constrains the solution and may explain this sensitivity. It has been verified for example that both pressure corrections (decoupling and inner penalization) have a stronger influence, decreasing both the global and local errors in a significant manner, when no Dirichlet condition is imposed for the pressure on the symmetry boundary (zero-gradient everywhere). Other cases have been considered for a quantitative validation of xIBM.

4.1.3 Sensitivity to the IB surface mesh

All velocity correction considered require that the distance to the immersed interface is estimated. For the sake of simplicity and computational time when complex geometries are involved, this distance is approximated as the distance to the closest vertex of the surface mesh. Thereby, too coarse a definition of the immersed surface mesh can generate additional errors in the corrections of the penalized velocity. In order to assess the sensitivity of the correction method to this surface mesh, global errors for the Wannier case are computed with a coarse surface mesh (with element's size comparable to the cell size), a fine surface mesh (with element's size four times smaller than the fluid mesh) and finally with the analytical distance to the interface used instead of the distance to the closest solid point.

The global errors are shown for the penalty methods with outer and inner velocity correction, respectively in Figure 4.2(b) and Figure 4.2(a). The correction methods are consistent in the way that the errors are shown to converge towards the one obtained with the analytical distance. However, the data from Figure 4.2(a) shows that the more accurate the estimation of the distance, the higher the global error for the penalty method with outer velocity correction. This result can be explained by a virtual thickening of the immersed solid. The penalization of the corrected velocity in the outer cell layer in the fluid domain is indeed thickening the real dimension of the IB, and this was shown in several benchmarks (see next test cases). In this case, the use of a coarse surface mesh, which overestimates the distance to the interface, compensates the artificial thickening and leads to a smaller global error. As is visible in Figure 4.2(b), the correction of the penalized velocity inside the immersed body reduces the error by comparison with Figure 4.2(a) and probably removes this thickening. The error obtained with the analytical distance is indeed the lowest in this case. Finally, one can see that a surface mesh with element's size of the order of the cell size is giving only an increase of 1.6% of the global error.



(a) Global errors for the penalty method with outer velocity correction.

(b) Global errors for the penalty method with inner velocity correction.

Figure 4.2: Influence of the IB surface mesh on the global error (Grid 1).

4.2 Fixed cylinder in a cross flow

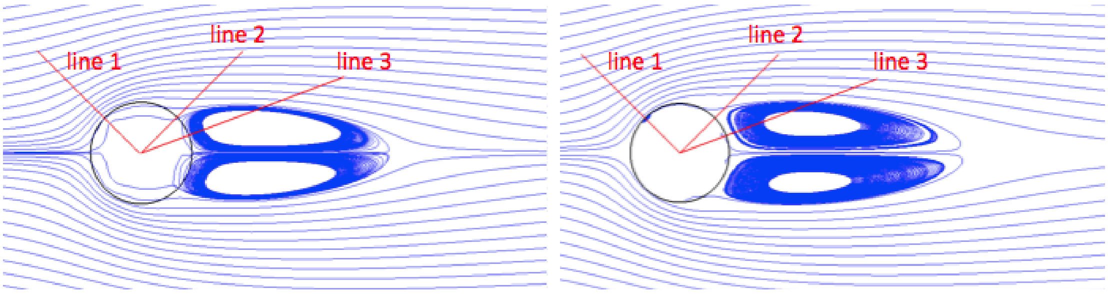
In this test case, a fixed cylinder in a 2D cross flow is considered. The domain is rectangular, with a length of $40D$ and a height of $30D$, where D is the diameter of the cylinder. The center of the cylinder is placed at $(10D,\ 15D)$. On the top and bottom walls a slip-condition is imposed, while inflow and outflow boundary conditions are fixed at the left and right boundaries respectively. A Reynolds number of 40 is considered. The same three grid refinements relatively to the cylinder, as for the Wannier flow, are used for this case (Grid 1, Grid 2 and Grid 3). This case has been widely studied and used for validation in literature.

The comparison focuses firstly on the drag coefficient C_D and the re-circulation length L_S behind the cylinder evaluated from the steady-state solutions. The drag coefficient is computed from the estimation of the force exerted by the fluid with the Method 2 described in Section 3.1.5. Both physical parameters obtained in previous published experimental and numerical work focusing on IBM are compared with results from the present penalization method and the equivalent body-fitted simulations (see Table 4.4). For a similar resolution near the interface ($\Delta x = 0.025D$ for grid 3, $\Delta x = 0.05D$ for Xu and Wang [154], and $\Delta x = 0.02D$ for Taira and Colonius [128]), the two physical parameters obtained with the current penalization approaches are in good agreement with what can be found in literature. More precisely, one can note the improvements made by the corrections for the penalty method, which increase both the drag and the re-circulation length, in accordance with the body-fitted results. The order of accuracy obtained for the recirculation length using Richardson method are also listed in Table 4.4. One can firstly note a very high order obtained with the body-fitted approach. Two possible causes were identified to explain this result: (i) a converged value of L_S could have already been reached with the medium grid, and (ii) the presence of skewed cells at the transition between the core of the computational domain and the boundary conforming layer may deteriorate the order of accuracy. The orders obtained with the different corrected IBMs show (i) a clear improvement in comparison to the simple first-order IBM, and (ii) a good agreement with the orders of accuracy found in literature for already published IBM, except for the inner velocity correction scheme. In this case, the order has been reduced from 1 to 0.5 in comparison to the simple penalty approach. It is possible that an additional level of refinement is needed for this method in order to get a correct estimation of the order. However another possibility is that the mirrored velocities resulting from the inner extrapolation scheme deteriorate the pressure field near the immersed wall and thus the separation point and the recirculation zone as well. The results from other velocity correction schemes and from the pressure decoupling are not listed in the table here. As it was shown for the Wannier case in the previous section, these correction methods are leading to results, which are very similar to the penalty approach with outer velocity correction.

Model	C_D	L_S	**order for L_S**
Coutanceau and Bouard (*Exp.*) [24]	–	2.13	
Tritton (*Exp.*) [135]	1.59	–	
Taira and Colonius [128]	1.54	2.30	$1\ to\ 1.5$
Xu and Wang [154]	1.66	2.21	$1.6\ to\ 2.3$
Parussini and Pediroda [100]	1.55	2.27	/
Body-fitted	1.60	2.31	5.1
Simple Penalty	1.58	2.21	1.0
Penalty with outer velocity correction	1.60	2.29	1.4
Penalty with inner velocity correction	1.59	2.27	0.5
... and inner pressure penalization	1.60	2.30	1.5

Table 4.4: Comparison of the drag coefficient and re-circulation length obtained with the fine grid (Grid 3) in a steady-state with literature data.

The streamlines of the flow obtained with the present sharp penalization method are shown in Figure 4.3. As in reality the penalization coefficient (in Equations 3.3 and 3.5) is not exactly equal to 0, the zero Dirichlet condition on the immersed interface in this case is not satisfied up to the machine accuracy. As a result, the streamlines may penetrate the immersed solid as shown for the simple Penalty approach. The improvements on the interface boundary conditions obtained with the corrected Penalty formulation are visible on the same figure, as they reduce significantly the penetration of the streamlines inside the body. In addition, velocity profiles are extracted along the lines marked in red in Figure 4.3. These lines make respectively an angle with the vertical axis of $-45°$, $45°$ and $70°$, which correspond to a location where the boundary layer is developed (line 1), a location after the separation point (line 2), and finally a location crossing the re-circulation bubble (line 3). The corresponding velocity profiles are shown in Figures 4.4, 4.5 and 4.6, for both streamwise and transverse components. Moreover, for the sake of visibility in the comparison, some zooming was performed on the region close to the interface. On these three figures, results are shown for the fine grid and are non-dimensionalized with the inflow velocity U_0.

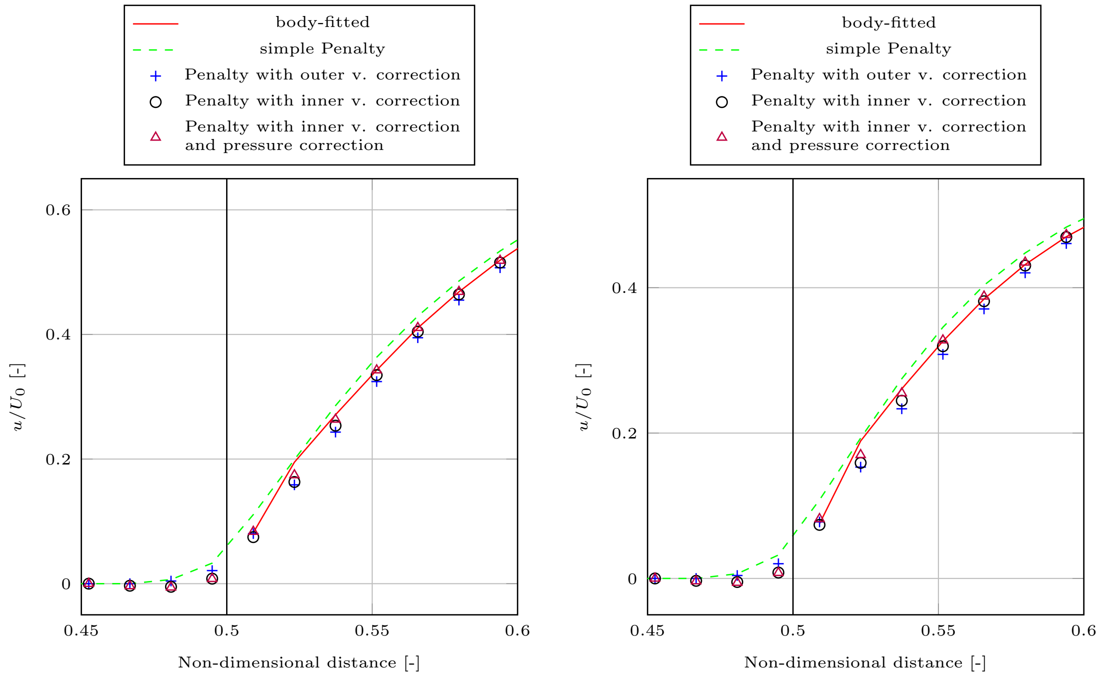


(a) Simple Penalty

(b) Penalty with corrections (inner velocity and pressure penalization)

Figure 4.3: Streamlines of the flow around a cylinder obtained with the coarse grid (Grid 1). The red lines show the position of the extracted velocity profiles.

Figure 4.4 shows the velocity profile before the separation point, where the boundary layer is still developed. As the main issue in this problem is actually for a numerical model to determine the right location of this separation point, all the models lead as expected to similar profiles at this location. One can still note the improvements brought by the corrected penalty approaches in comparison to the simple penalty method. The model with inner velocity and pressure corrections (purple triangle in the figure) give the best results after $0.55D$. However between $0.5D$ and $0.55D$, which corresponds to the first cell, it appears that the simple penalty approach is the closest to the body-fitted one. This underlines again the thickening effect of the velocity penalization methods. Similarly to the previous cases, we can see that the inner penalization of velocity, and then the additional correction of pressure, are both reducing the effect of this thickening.



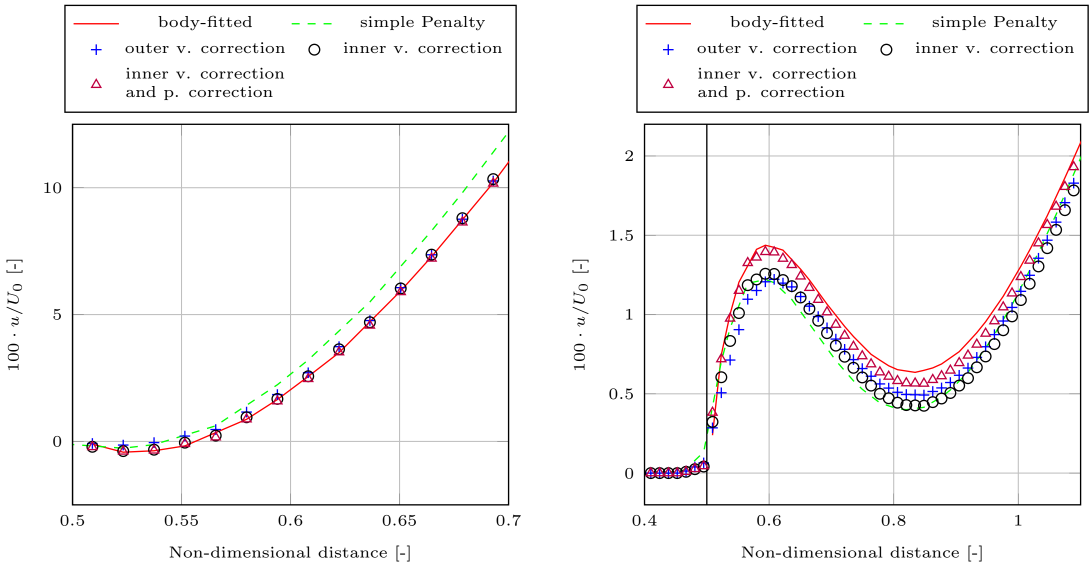
(a) Steamwise velocity profile

(b) Transverse velocity profile

Figure 4.4: Velocity profiles along the line 1 for different IBM. Results obtained with the fine grid (Grid 3). The vertical black line depicts the position of the cylinder interface.

After the separation points, the profiles from Figures 4.5 and 4.6 show the significant improvements with the different corrected penalization approaches compared to the first order model. Again, the penalty approach with both inner velocity and pressure corrections gives the best results, both for the separation region and the re-circulation bubble. These conclusions are consistent with the numerical estimation of the re-circulation lengths, for which the simple penalty method gives the biggest discrepancy with the body-fitted estimation (around 5%), while the penalty approach, with velocity and pressure corrections, has shown to be the most accurate model.

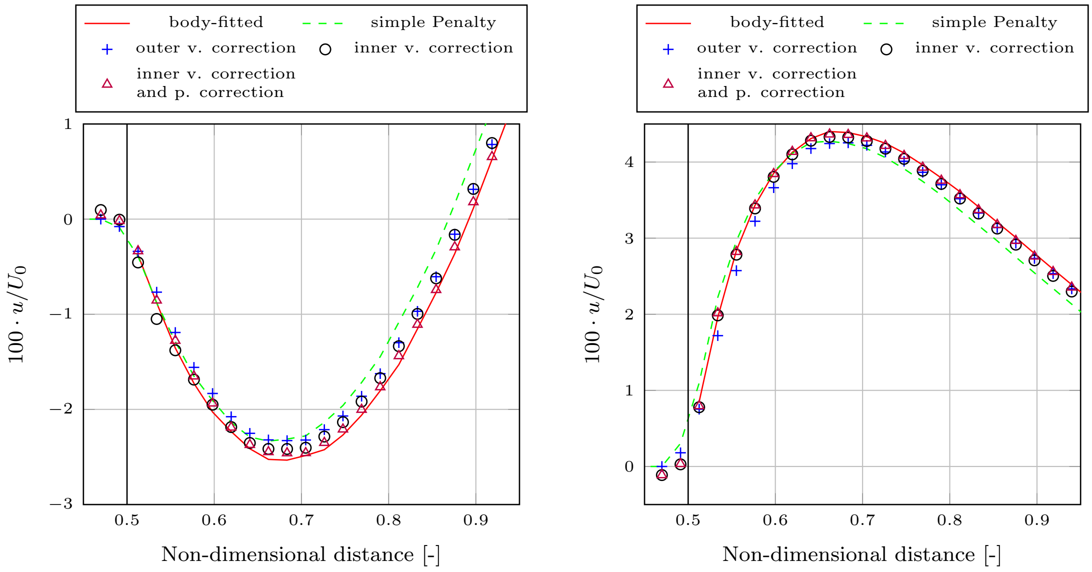
Figure 4.7 shows locally the order of accuracy for the two velocity components computed with both simple and improved IBM. The latter refers to the penalty approach with inner correction of the velocity and with pressure penalization in addition to satisfy



(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 4.5: Velocity profiles along the line 2 for different IBM. Results obtained with the fine grid (Grid 3). The vertical black line depicts the position of the cylinder interface.

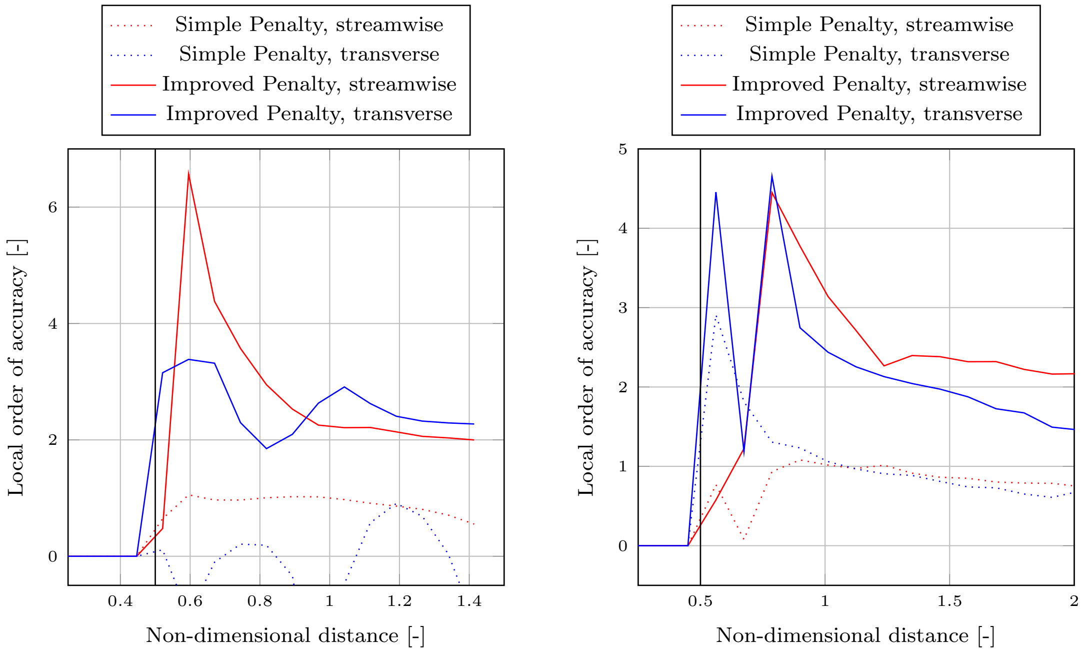


(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 4.6: Velocity profiles along the line 3 for different IBM. Results obtained with the fine grid (Grid 3). The vertical black line depicts the position of the cylinder interface.

a Neumann boundary condition at the interface. The orders are estimated along the lines 2 and 3 (see Figure 4.3) with the Richardson method [108], using an additional level of refinement for the finest grid. The different curves exhibit a high variability of the order near the interface. Although the first sampling point shows a low order for the two penalty approaches, the average order of accuracy is above 2 for the penalty with corrections at a distance of one radius from the interface. Considering the streamwise component, away from the interface, the values seem to converge asymptotically towards 2 for the improved IBM, and towards 1 for the standard penalty approach. Similarly for the transverse component of the velocity, the order stabilizes between 1.5 and 2 for the improved IBM depending on the sampling line. One can note that the orders outline a higher variability for the transverse component. In addition, the local orders obtained for the simple penalty approach along line 2, which is in the vicinity to the separation point and on the edge of the recirculation bubble, are negatives in some parts. It could be the case that between two grid levels, the sampling line is passing inside or outside the recirculation zone, which would affect significantly the extracted velocity magnitude. An extra level of refinement might be then necessary to obtain consistent orders of accuracy. The results highlighted here emphasize the improvements brought by the corrections. Furthermore, they are consistent with the orders of accuracy found for the recirculation length for instance (1.0 for the standard IBM and 1.5 for the improved IBM).



(a) Local orders of accuracy along line 2

(b) Local orders of accuracy along line 3

Figure 4.7: Local order of accuracy for the velocity along the lines 2 and 3. 20 sampling points were considered for the estimation of the orders. The vertical black line depicts the position of the cylinder interface.

4.3 In-line oscillating cylinder in a fluid at rest

A cylinder oscillating in a fluid at rest is considered in this case. The dimensions of the problem are similar to the previous test case, as well as the IBM and body-conforming grids used for the simulations. In the standard body-fitted approach, the cells are slightly and smoothly stretched in time, in order to follow the motion of the cylinder. The equation of motion reads: $x(t) = -Asin(\omega t)$, where the frequency $f = \frac{\omega}{2\pi}$ is set to $0.2Hz$ and the amplitude A is fixed to $\frac{5}{2\pi}$. This motion corresponds to a Reynolds number equal to 100 and a Keulegan-Carpenter number equal to 5.

4.3.1 General results

The oscillating cylinder case is a validation case also frequently used in literature ([85], [116], [156], [83]) for numerical model handling moving geometries. Laboratory experiments have been conducted as well by Dütsch et al. [32]. The oscillations of the cylinder lead to the development of lower and upper boundary layers, which separate and generate two counter-rotating vortices in the lee. This generation process stops when the cylinder starts to move backward and finally splits the vortex pair. The pressure contours obtained with our simple penalty approach are plotted in Figure 4.8 for a phase position $\omega t = 180°$, when the cylinder is in its central position and is moving to the right. The pressure contour shows the symmetry obtained with our model, and is in good agreement with the results of Dütsch et al. [32]. On this figure, the white lines represent the position of the extracted velocity and pressure profiles at the same phase angle, respectively at a distance $x = -0.6D$, $x = 0$, and $x = 0.6D$ from the center of the cylinder. At this time, the cylinder is back to its initial central position, for which the body-fitted mesh is not stretched, allowing for a more relevant comparison with the IBM results.

The pressure is normalized in terms of the cylinder maximum speed U_m and the average pressure at the inlet p_I as $p^* = (p - p_I)/(\rho U_m^2)$. The velocity is normalized as $\mathbf{u}^* = \mathbf{u}/U_m$. Figure 4.9 details the velocity profiles across the center of the cylinder for the different IBMs. Here also, the penalty approach with both inner velocity and pressure

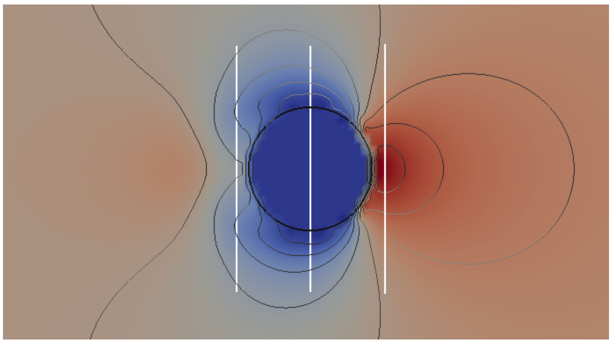
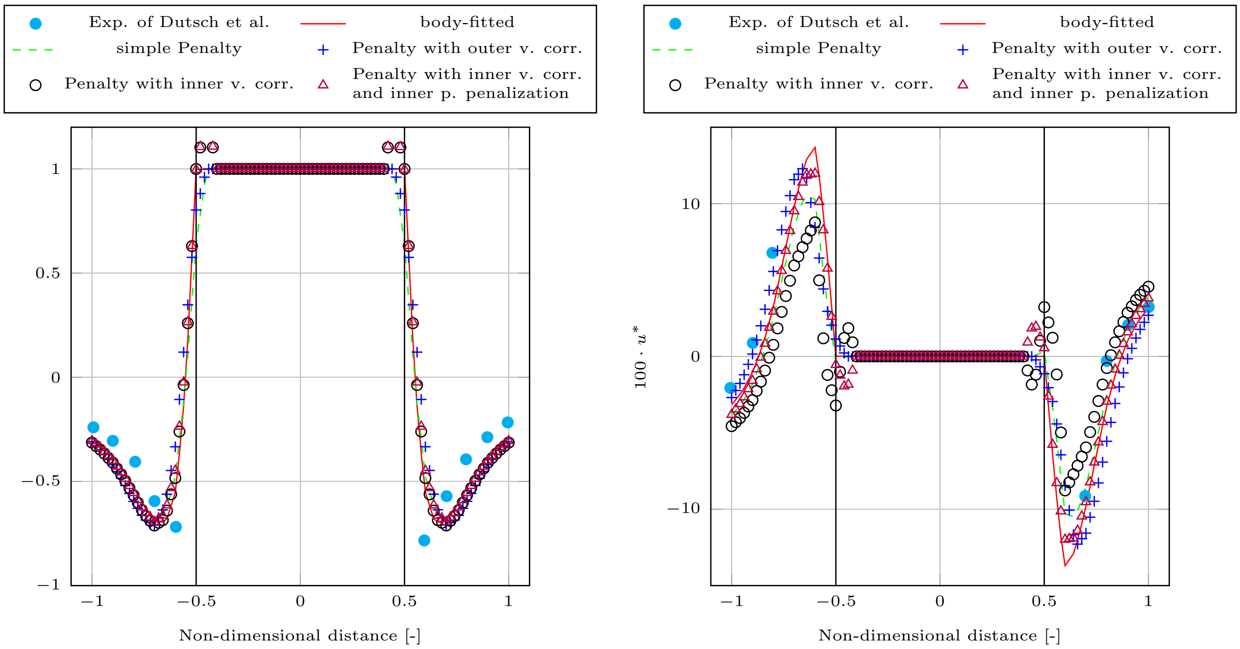


Figure 4.8: Pressure contours around the oscillating cylinder for a phase angle $\omega t = 180°$, obtained with the simple Penalty approach on the medium grid (Grid 2). The white lines show the position of the extracted velocity and pressure profiles.

correction gives the best results in comparison to the numerical results of the body-fitted simulations. The improvements of this penalty approach are especially visible near the interface for both streamwise and transverse velocities (between $-0.6D$ and $-0.5D$, and between $0.5D$ and $0.6D$), where a good agreement with the body-fitted case can be found. One can note in particular the good estimation of the transverse velocity peaks. As for the previous cases, the thickening effect of the outer velocity correction is visible in Figure 4.9 for both components. Similarly, for the inner velocity correction, one can see a clear improvement if the pressure is corrected. For the two other profiles on both sides of the cylinder in Figure 4.10, the penalization approach leads to satisfactory results as well. The experimental results of Dütsch et al. are not shown here for a quantitative validation, but for comparison purpose. The profiles obtained with IBM are consistent with the experimental velocities and show thus that the qualitative features of the flow are well captured. Furthermore, similar level of discrepancies with these experimental data are obtained with other numerical models in literature [85], [83].

Figure 4.11 shows instantaneous pressure profiles along the three sampling lines considered in this test case. When comparing the inner velocity correction with and without pressure treatment, one can observe that the imposition of the Neumann condition improves the results near the interface on the profile crossing the cylinder ($x = 0$) in Figure 4.11(a). For the two other profiles, although the magnitude is slightly under-



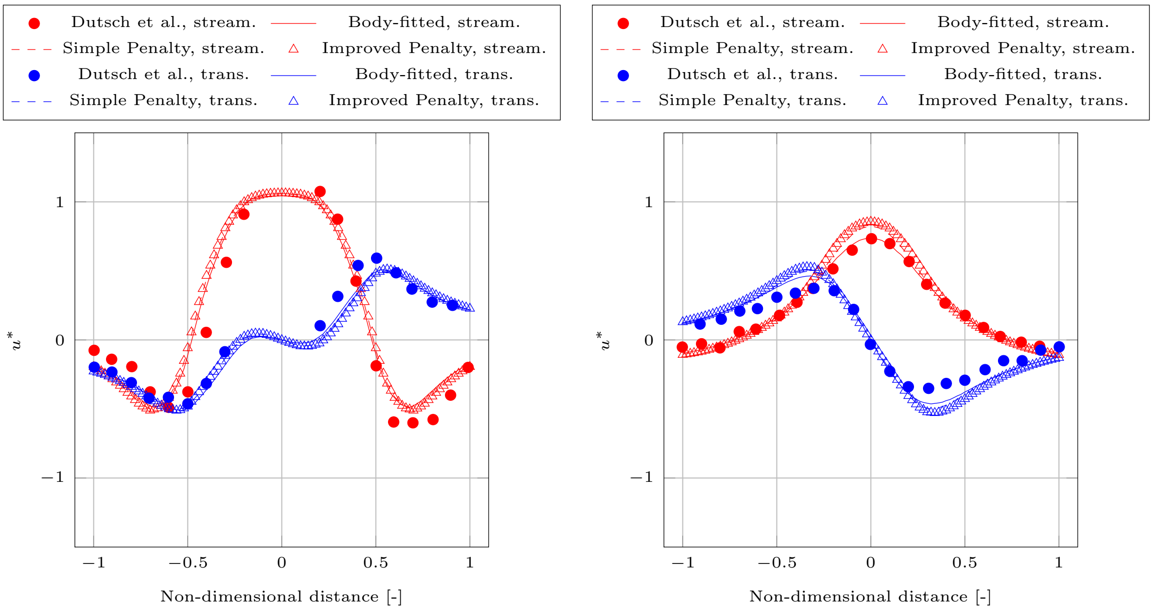
(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 4.9: Instantaneous velocity profiles near the oscillating cylinder, at $x = 0$ and for a phase angle $\omega t = 180°$. The results are obtained with Grid 2. The vertical black lines depict the position of the cylinder interface.

estimated, Figure 4.11(b) and 4.11(c) show a better distribution of the pressure around the cylinder for this particular penalization approach. This result supports the fact that the drag is correctly estimated in this case, and implies as well that the pressure gradient is very similar between the body-fitted and this corrected penalty method. Similarly, the large peak obtained with the simple penalty approach and the outer velocity corrections scheme are reduced in accordance with the body-fitted case.

The drag coefficients listed in Table 4.5 for each penalty approach and the body-fitted case, are time-averaged over half a period. The drag coefficient is computed from the estimation of the force exerted by the fluid with the Method 1 described in Section 3.1.5. One can see here the evidence of a better pressure distribution around the cylinder when the Neumann boundary condition is satisfied within the penalty approach. In this case, we can retrieve an averaged drag coefficient identical to the one obtained with the



(a) Profiles at $x = -0.6D$

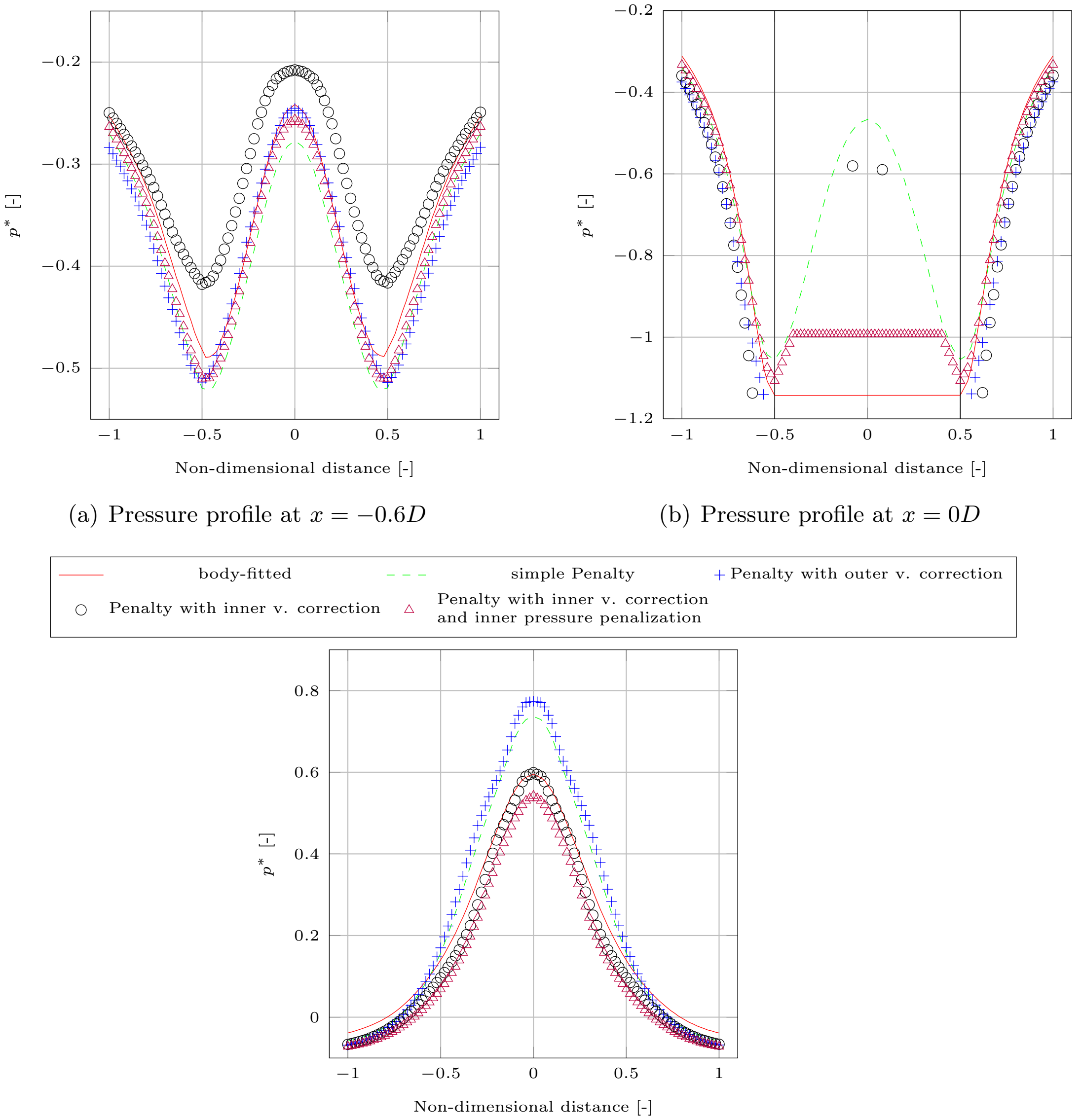
(b) Profiles at $x = 0.6D$

Figure 4.10: Instantaneous velocity profiles near the oscillating cylinder, at $x = 0.6D$ and $x = -0.6D$ and for a phase angle $\omega t = 180°$. The results are obtained with Grid 2 for the streamwise (in red) and the transverse (in blue) components of the velocity. The improved penalty refers to the penalty with inner velocity and pressure corrections.

body-conforming simulation.

Model	C_D
Body-fitted	2.05
simple Penalty	1.93
Penalty with outer velocity correction	1.94
Penalty with inner velocity correction	1.99
... and inner pressure penalization	2.05

Table 4.5: Values of the time averaged drag coefficients over half a period (Grid 3).



(a) Pressure profile at $x = -0.6D$

(b) Pressure profile at $x = 0D$

(c) Pressure profile at $x = 0.6D$

Figure 4.11: Instantaneous pressure profiles along the three sampling lines near the oscillating cylinder for a phase angle $\omega t = 180°$. The results are obtained with the medium grid (Grid 2).

4.3.2 Sensitivity to the penalization coefficient

The influences of the penalization coefficient on the flow are described here. The time evolution of the drag coefficient is shown in Figure 4.12 for different values of the penalization coefficient K. The results are obtained with the simple penalty method on the medium grid (Grid 2). The mean value of the drag coefficient is converging towards a single value by increasing K, implying that the penalty model proposed here is consistent. However, one can note the appearance of oscillations for high values of K. For a low value of the penalization coefficient, the evolution of the drag is delayed, showing that K is playing the role of a relaxation time. Furthermore in this case, the IBM force is not sufficient to ensure the correct no-slip condition for the cylinder, leading to a lower drag. These results show that a compromise can be made in order to choose a relevant order of magnitude for the penalization coefficient. The estimation of K should generally take into account the values of the time step and of the kinematic viscosity.

4.3.3 Sensitivity to the coincidence with the mesh

In order to limit the errors introduced by the use of an IBM to represent curved geometry on Cartesian grids, it is usual to place the solid symmetrically in regards to the grid. This positioning generally helps to preserve the symmetry of the flow for simple solid geometries. It is however interesting to see the influence of a non-symmetric positioning on the results. Simulation presented here were presented with Grid 2. For this purpose, the cylinder has been shifted by $0.24h$ (h being the cell size) in the upward y direction. Figure 4.13 shows the pressure profiles at $x = 0.0D$ and $x = 0.6D$ for the body-fitted approach as reference, and compares the profiles for IBMs with a symmetric or coincident positioning, with those obtained with a shifted position. Not surprisingly, shifting the position breaks the symmetry for IBMs. As visible in Figure 4.13(a), this loss is far more important for the simple penalty model. As for the improved IBM, which refers to the penalty approach involving both inner velocity and pressure corrections, the non-symmetric positioning is balanced by the latter interpolations. In Figure 4.13(b), one can see that the influence of the breaking symmetry becomes more important as the

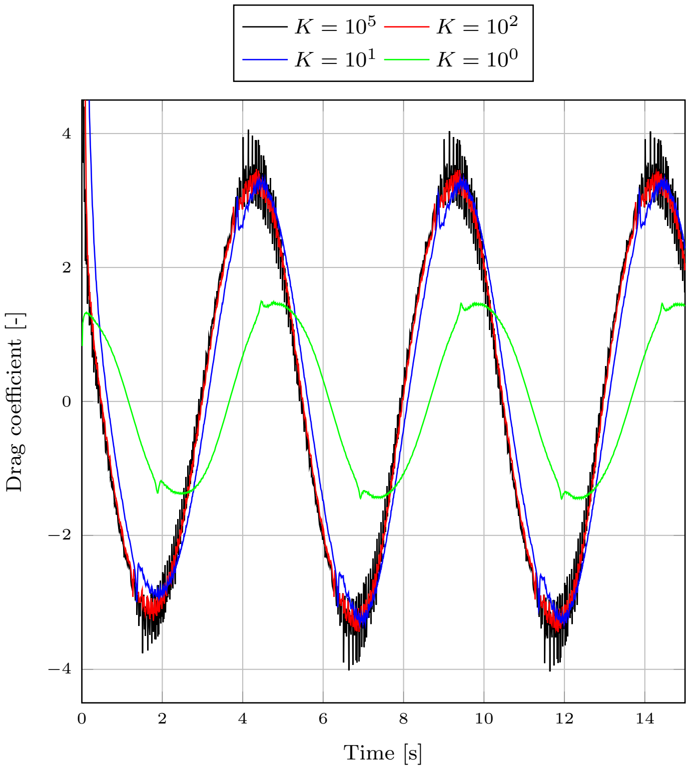
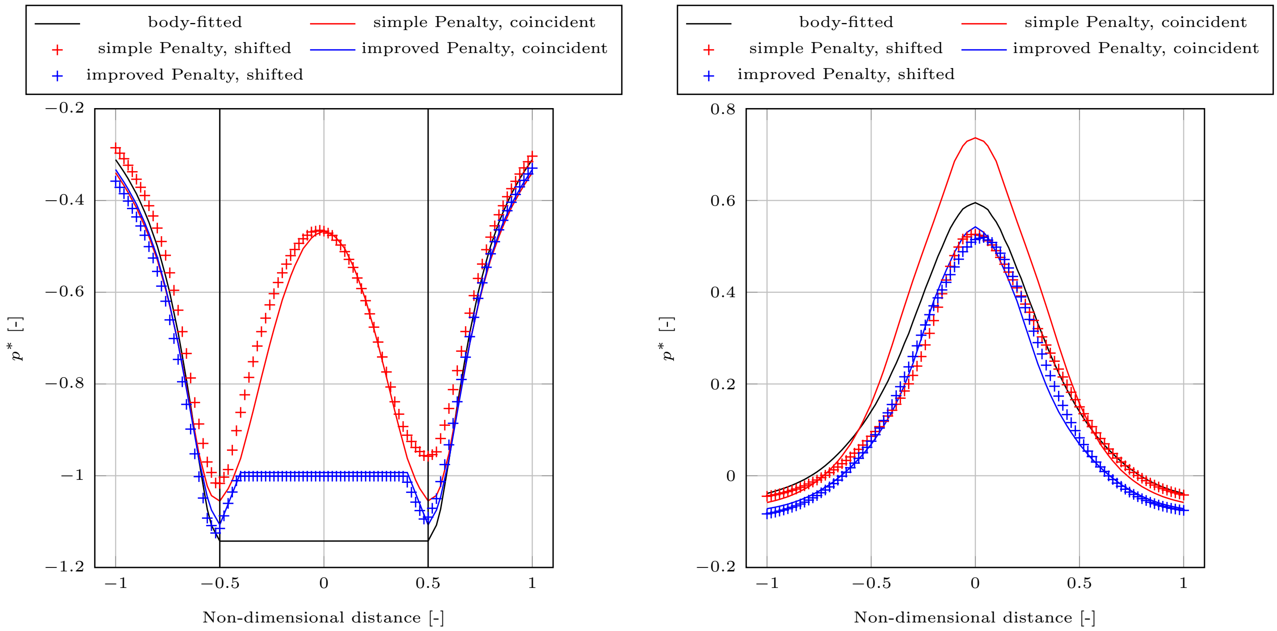


Figure 4.12: Evolution of the drag coefficient for different penalization coefficient. Results obtained with the simple penalty method on the medium grid (Grid 2).

distance from the interface increases when the simple penalty approach is used. This sensitivity is reduced with the corrected penalty model. The same conclusions can be drawn for the velocity profiles, but the differences are less visible. In this type of case, the benefits of the corrections for the penalty method are finally more visible. As most of the engineering applications do not involve symmetric solids, these results really support the necessity of velocity and pressure interpolations near the interface.



(a) Profiles at $x = 0.0D$

(b) Profiles at $x = 0.6D$

Figure 4.13: Instantaneous pressure profiles for a phase angle $\omega t = 180°$, for symmetric and non-symmetric positioning of the cylindrical obstacle. The results are obtained with the medium grid (Grid 2). The vertical black lines depict the position of the cylinder interface.

4.4 Simplified static impeller in a channel

4.4.1 General results

One considers here a four blade static impeller immersed in a 2D channel, of dimensions $60m$ in length and $10m$ width. The fixed impeller is modeled as a cross of length $4m$ with a $1m$ blade thickness. The latter is centred at $20m$ from the inlet, with an angle of 20 degrees with the bottom wall. A flow of $1m/s$ is imposed at the inlet, which corresponds to a Reynolds number around 400. Slip walls are considered at the top and bottom walls. A zero pressure gradient condition is fixed at all boundaries, except at the outlet, where a Dirichlet condition is used. In the present case, the flow separation is directly governed by the solid geometry, and leads to vortex shedding in this flow regime. In order to compare our IBM to the standard Navier-Stokes model with body-conforming grids, the time evolution of the surface averaged pressure at the inlet is extracted, as well as the drag and lift coefficients for the impeller. The surface averaged pressure is equivalent to the head provided by a pump, as analyzed in the full engineering cases in Chapter 5. The dimension of the problem as well as the general aspect of the flow are visible in Figure 4.14. The IBM grid is here also fully Cartesian with perfect orthogonal quality, while the body-fitted grid has been generated similarly to the Wannier case, with a cut-cell approach and a layer of conforming cells attached to the cross surface.

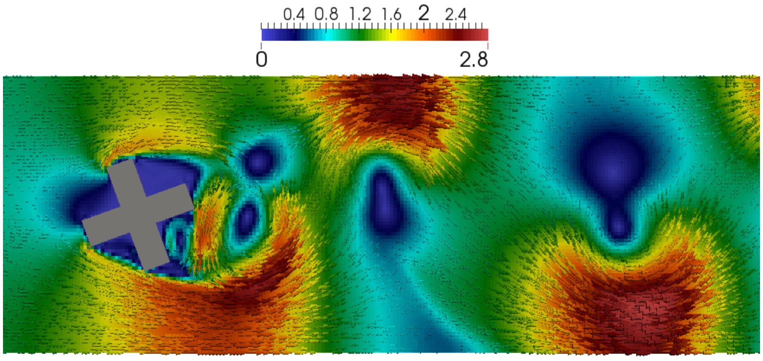


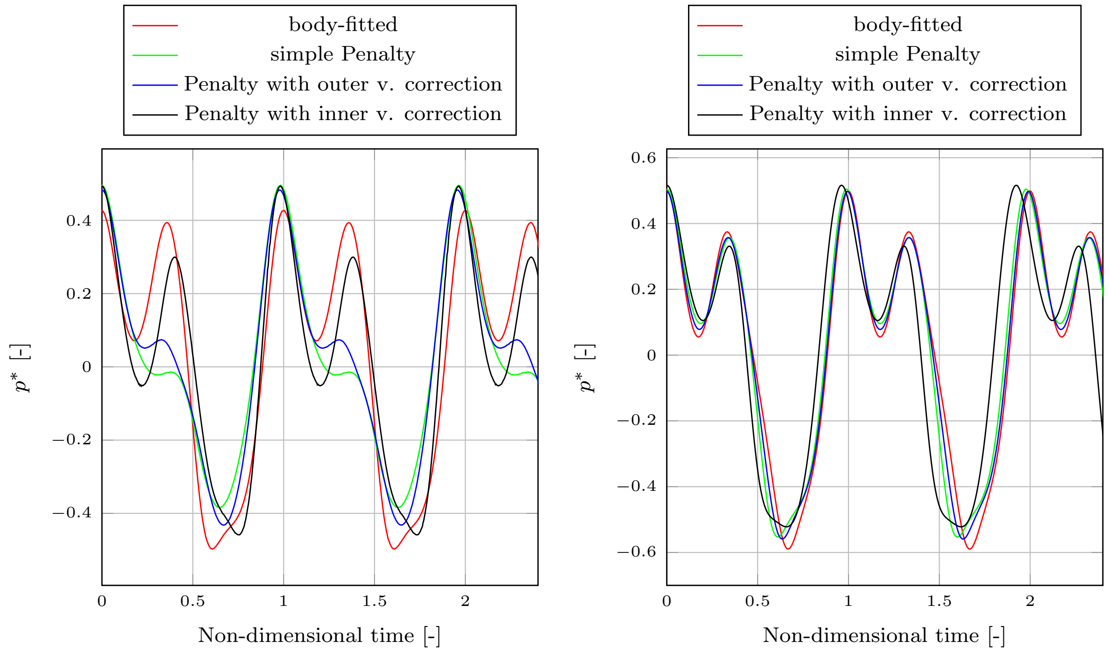
Figure 4.14: Velocity contours (m/s) and vectors of the flow through a static impeller in a channel.

Three levels of mesh refinement are considered in this problem, with (480x80), (960x1600) and (1920x320) cells, which corresponds to 32, 64 and 128 cells in the length of the cross respectively, or 8, 16 and 32 cells in its thickness. The evolution of the surface averaged pressure at the inlet is shown in Figure 4.15 for the body-fitted and the different penalty approaches, both on the coarse and the fine grids. The inlet pressure averaged over one period of oscillation $\bar{p_I}$ is used to non-dimensionalize the pressure as $p^* = (p - \bar{p_I})/(\rho U_m^2)$. For the sake of visibility, the time has been non-dimensionalized with the oscillation period obtained for the body-fitted method, and the data have been translated to start at one of the primary pressure peaks. It is important to note then that the oscillations are shifted not because of a delay on the onset of the vortex shedding, but because of small under-estimation of the period of oscillations. The average pressure and the amplitude of the oscillations are well estimated with relative errors on the fine grid below 3% and 5% respectively, for every penalty approach, in comparison to the the body-conforming simulations. On the coarse mesh, it appears that the resolution is not sufficient to reproduce the secondary peak of pressure with the IBMs, except for the penalty method with inner correction of the velocity. In this case, the difference between the two peaks is however over-estimated, in comparison to the results of the body-fitted model.

Regarding the pressure correction schemes, it was shown that the decoupling method has a negligible influence on the solution, as for the previous test cases. The penalization of the pressure however was found to be very sensitive to the proximity of the wall boundaries. The results from this scheme are discussed in Section 4.4.3.

4.4.2 Sensitivity to the discretization schemes

The same case is studied with a first order upwind scheme used for the discretization of the divergence terms. The results of the simple penalty method and of the penalty method with velocity correction are presented in Figure 4.16, and compared as usual to the body-fitted data. It appears from these results that the corrections have more of an impact when first order schemes are used. Indeed, the discrepancy between the simple



(a) Results obtained with the coarse grid (480x80)

(b) Results obtained with the fine grid (1920x320)

Figure 4.15: Evolution of the surface averaged pressure at the inlet

penalty approach and the model with corrections is smaller when second order schemes are used, as detailed previously (see Figure 4.15 in Section 4.4.1). One can thus conclude that the velocity correction is reducing the numerical diffusion happening close to the interface when first order spatial schemes are used, which is driving the vortex shedding and a fortiori the pressure evolution at the inlet.

4.4.3 Sensitivity to the proximity of the boundaries

In this section, the influence of the proximity of the wall boundaries is investigated for two parameters, the drag coefficient of the cross and again the surface averaged pressure at the inlet. The simulations were performed for cross width to channel width ratios of 2/5 and 2/30. The first ratio corresponds to the previous configuration, while the second corresponds to side walls away from the cross. The results are presented for the coarse grid merely to emphasize the differences obtained with the two ratios.

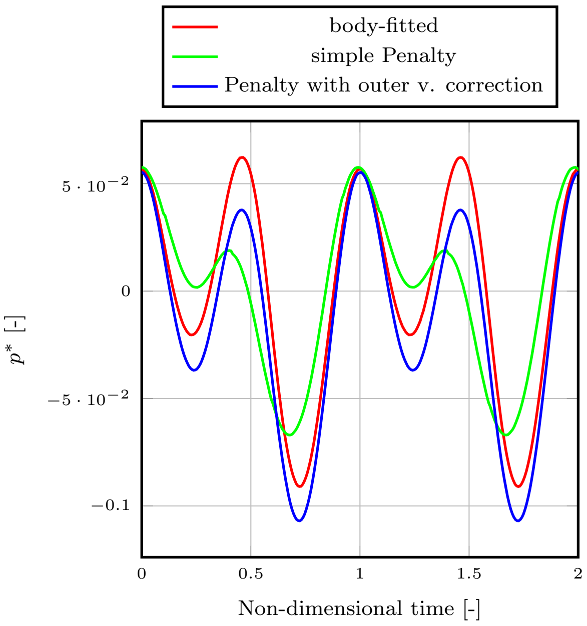
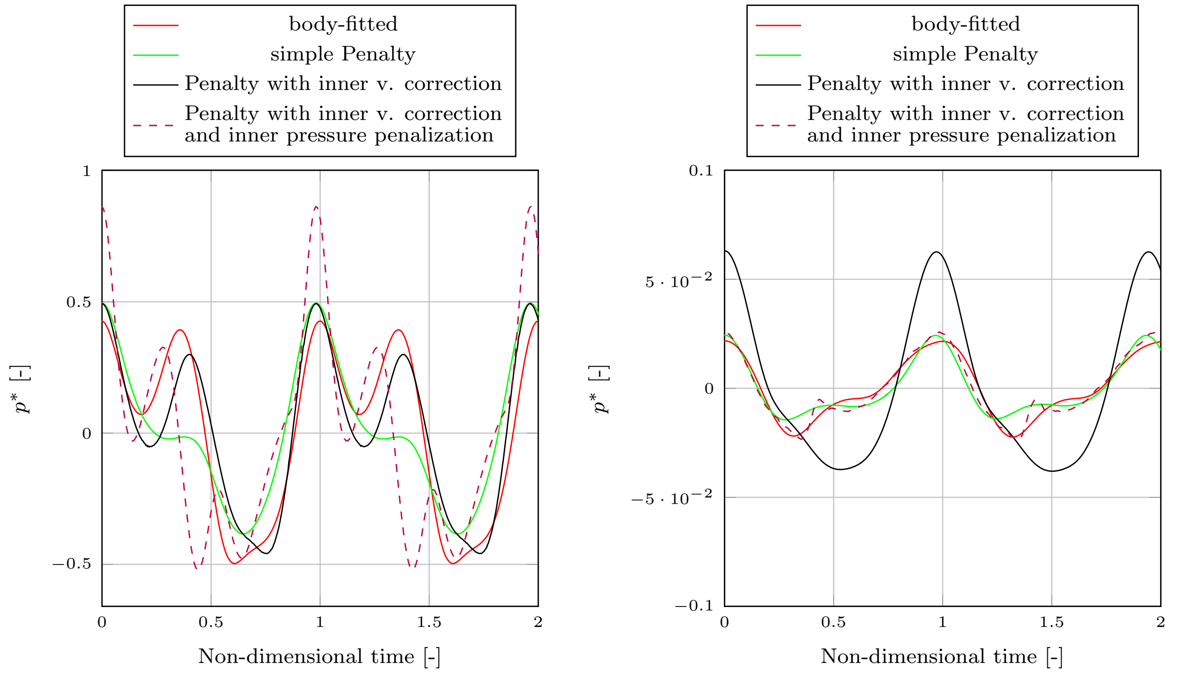


Figure 4.16: Evolution of the surface averaged pressure at the inlet for the coarse grid (480x80), obtained with first order spatial discretization schemes

As stated previously, the results for the small ratio plotted in Figure 4.17(a) shows that the penalty method with pressure penalization as a correction (in addition to the velocity correction) is unable to predict the correct pressure evolution at the inlet of the channel. Additional pressure peaks are predicted with this method, as well as an increase of the amplitude of the first harmonic. On the contrary, for the large cross to channel ratio, the penalty method with pressure penalization gives the best results upstream of the cross as illustrated in Figure 4.17(b). This difference might be due to a sensitivity to an over-constraint on the boundary conditions for the pressure. Indeed, the Neumann boundary condition at the immersed interface is imposed by a penalization of the pressure, which is of the Dirichlet type. The pressure could thus be mathematically over-constrained and be deteriorated by the close presence of an additional zero-gradient condition at the side walls. Again on Figure 4.17(b), one can see that the results obtained for the penalty method with inner velocity correction and no treatment for pressure are quite different from the body-fitted results, although the difference in the mean value is only around 0.4 %. This still suggests that the non-physical velocity gradients generated by the specific inner velocity correction slightly deteriorate the pressure field, showing that a correction of the pressure is necessary in this case. However, this difference is vanishing with finer grids.



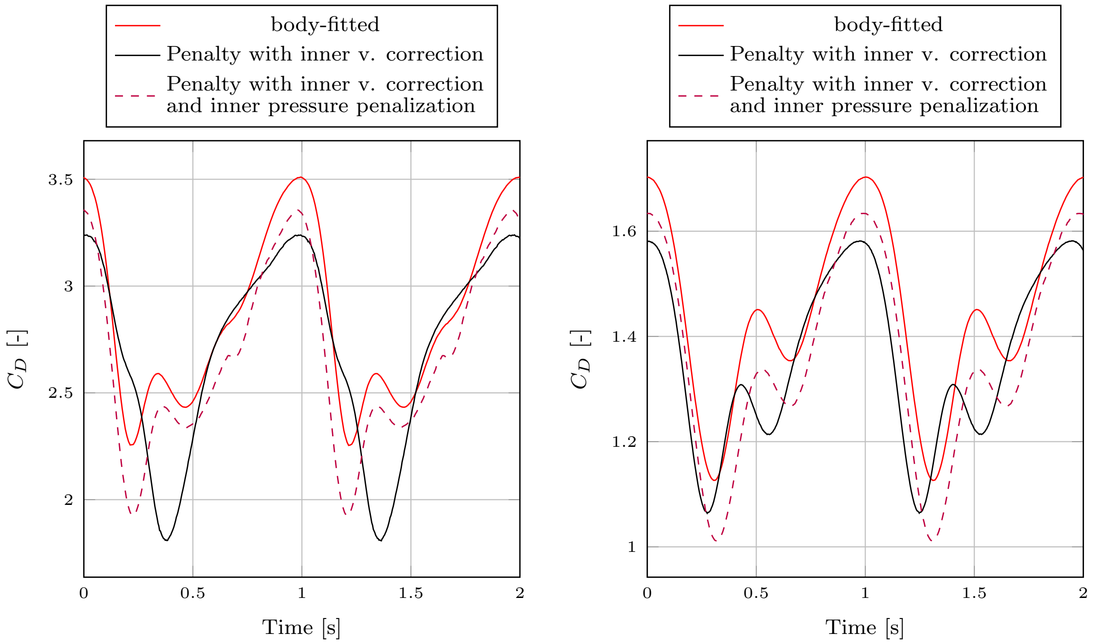
(a) Evolution of the surface averaged pressure for a cross to channel ratio of 2/5

(b) Evolution of the surface averaged pressure for a cross to channel ratio of 2/30

Figure 4.17: Influence of the proximity of the wall boundaries on the surface averaged pressure at the inlet. Results obtained with the coarse grid (480x80)

Interestingly, the pressure fluctuations highlighted from the use of the pressure correction scheme are only present upstream of the immersed obstacle. This results indicates that the errors introduced by such modification of the pressure equation are transported in the opposite direction of the flow during the PISO iterations. Finally, it was found that using different types of inflow and outflow boundary conditions for the simulation domain does not solve this issue concerning the proximity of wall boundaries. Although lower, the spurious pressure fluctuations are still present upstream of the immersed boundaries (and absent downstream) when a pressure gradient is for instance imposed through the channel.

Regarding the drag coefficients, the influence of the proximity of the wall boundary conditions on the latter is not as significant as for the pressure, as shown in Figure 4.18. For both cross to channel ratios, the difference between the results obtained with a Penalty model and the standard Navier-Stokes model are equivalent. It is interesting to note again that the best results are obtained for the penalty method with both velocity and pressure correction. This is even the case for the small ratio in Figure 4.18(a), when the pressure field exhibits fluctuations, showing that locally the gradient of pressure is likely to be correct and improved in comparison to a penalty approach without pressure correction.



(a) Evolution of the drag coefficient for a cross to channel ratio of 2/5

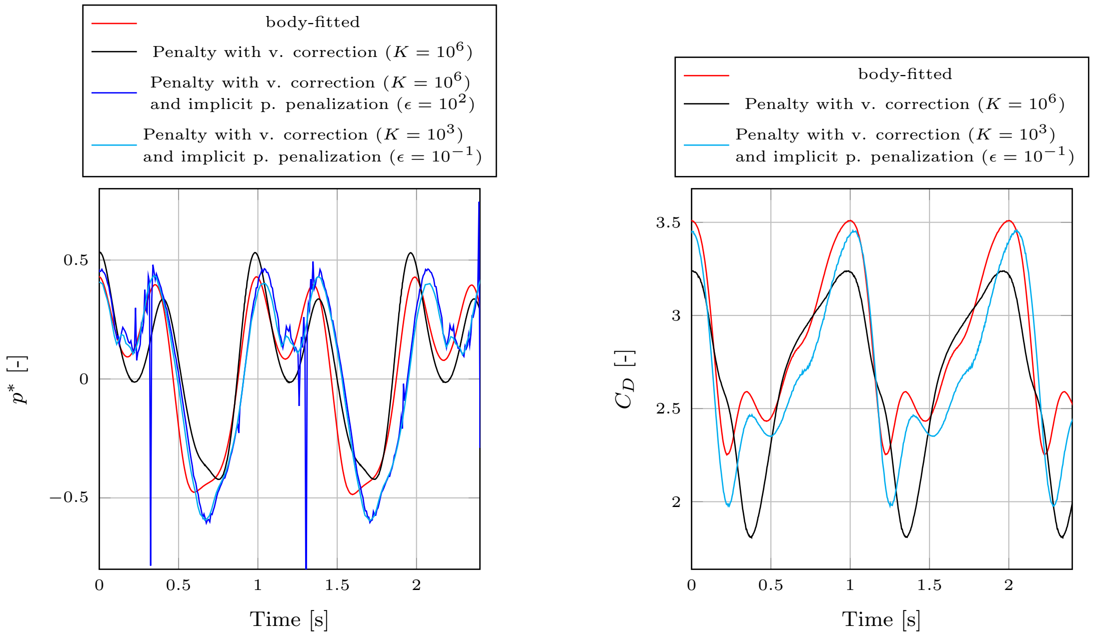
(b) Evolution of the drag coefficient for a cross to channel ratio of 2/30

Figure 4.18: Influence of the proximity of the wall boundaries on the drag coefficient. Results obtained with the coarse grid (480x80)

4.5 On the use of an implicit penalization correction for pressure

The implicit penalization has been implemented with the possibility of adapting the penalization coefficient ϵ for the pressure equation. The results of this model are presented in this section for both the simplified impeller case, and the oscillating cylinder case. Figure 4.19 shows the surface averaged pressure at the inlet and the drag coefficient of the cross for two values of penalization coefficients with an implicit pressure correction. As it was already shown in section 4.3.2 for K, reducing both velocity and pressure penalization coefficients removes the spurious pressure fluctuations visible in Figure 4.19(a). With a relevant choice of coefficient, the evolution of the equivalent head can be reproduced with a very good agreement, and gives clearly better results than the simple penalty method or the penalty method with only velocity correction. The conclusion is the same for the evolution of the drag coefficient, as it is visible in Figure 4.19(b). Furthermore, the detrimental influence of the proximity of other boundaries is not appearing with an implicit pressure penalization. This might be due to the fact that the Neumann condition is implicitly satisfied, by comparison to the explicit penalization of pressure for which an equivalent Dirichlet condition is used. There is however a limitation for this type of pressure correction. Indeed, the modified matrix becomes asymmetric, as the penalization for the Neumann condition is added only in one direction of the interface normal. This characteristic makes the matrix system harder to resolve, and it has been found unstable for cases with more complex solid geometries.

The implicit pressure penalization has been tested as well with the oscillating cylinder case, with a standard penalization coefficient for velocity and $\epsilon = 10^2$ for the penalization of pressure. Figure 4.20 shows the velocity profiles at $x = 0$, while Figure 4.21 shows the pressure profiles at $x = -0.6D$ and $x = 0.6D$ for the same phase angle. The streamwise and transverse velocities are reproduced with this implicit pressure penalization method as well as with the explicit method. Regarding the pressure, although fairly close to the standard explicit penalization, the implicit penalization under-estimates the central

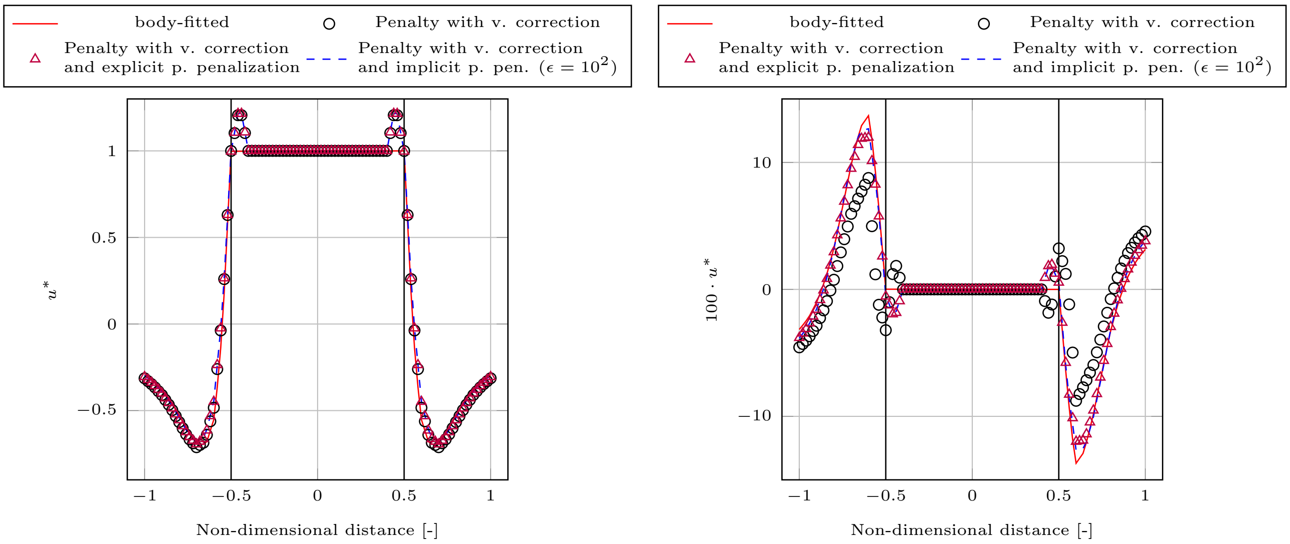


(a) Evolution of the surface averaged pressure

(b) Evolution of the drag coefficient

Figure 4.19: Influence of the penalization coefficient for the implicit pressure correction. Results obtained with the coarse grid (480x80)

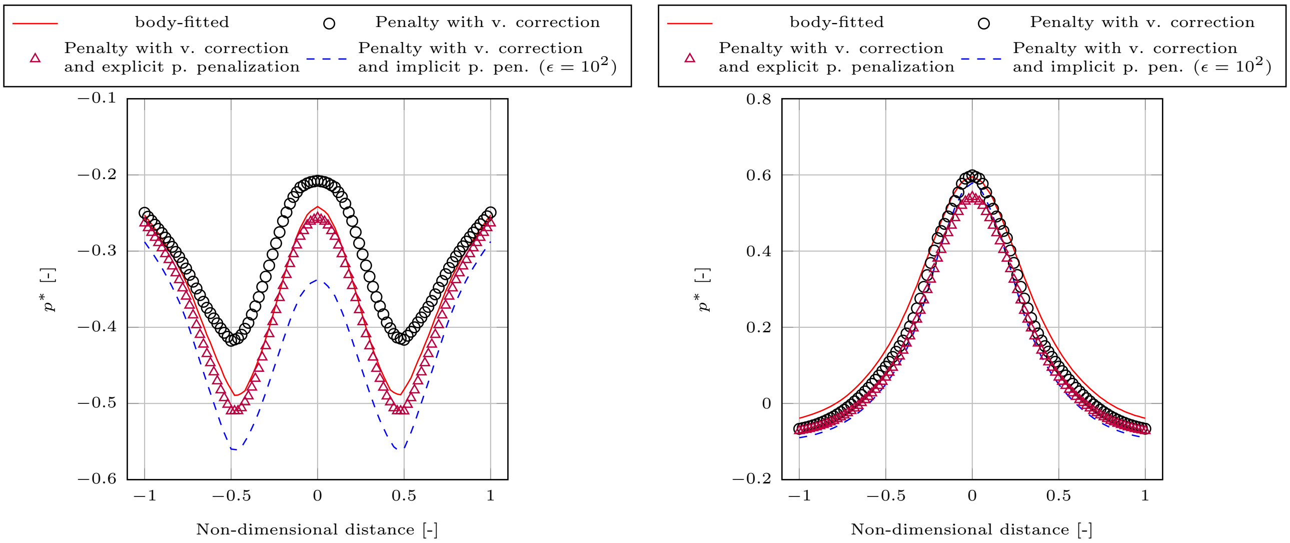
peak for the profiles $x = -0.6D$, relative to the magnitude of the pressure at the extremities. This result shows that the pressure gradient has been slightly disimproved. The coefficient ϵ is certainly not large enough in this case to satisfy the correct Neumann condition. However, a larger penalization coefficient seems to destabilize the simulation. Despite this discrepancy, the use of an implicit pressure correction scheme still leads to results in better agreement with the body-fitted simulation in comparison to a penalization approach without pressure correction.



(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 4.20: Comparison of velocity profiles at $x = 0$ and with phase angle $\omega t = 180°$ for explicit and implicit pressure penalization (on Grid 2). The vertical black lines depict the position of the cylinder interface.



(a) Pressure profile at $x = -0.6D$

(b) Pressure profile at $x = 0.6D$

Figure 4.21: Comparison of pressure profiles at $x = -0.6D$ and $x = 0.6D$, with a phase angle $\omega t = 180°$, for explicit and implicit pressure penalization. Results obtained with Grid 2

4.6 Conclusion

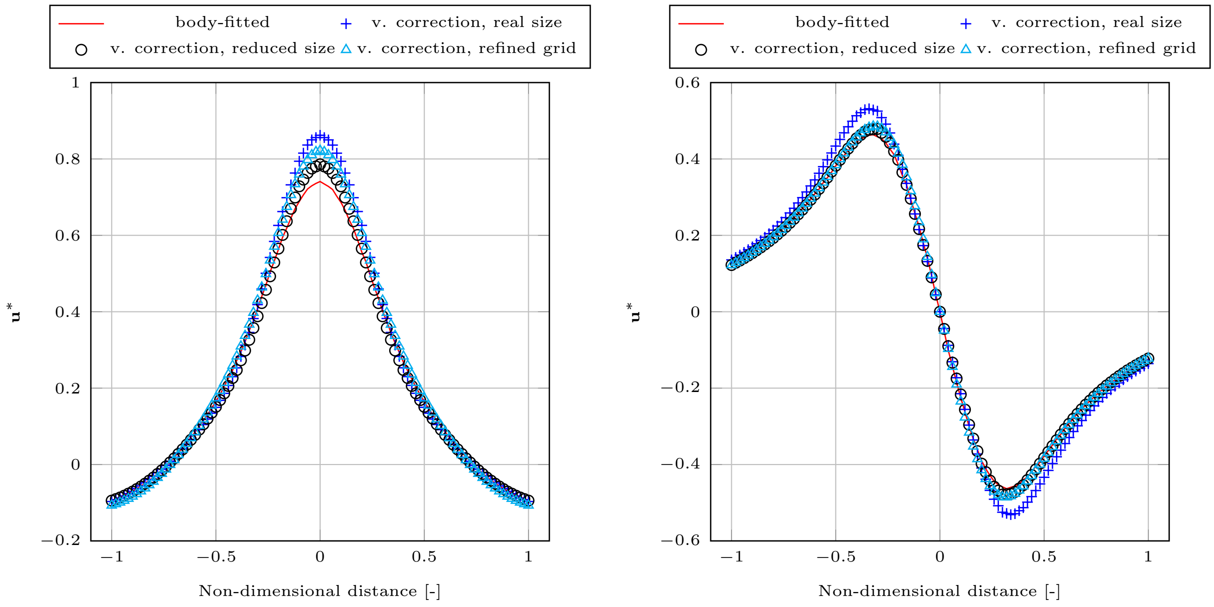
The validation of the penalty based IBM developed during this research has been performed with several test cases, involving fixed and moving geometries, with continuous and discontinuous curvature. The formulation preserves the sharpness of the immersed solid. A simple first order approach is able to estimate correctly the integrated variables of interest in each case. However, local profiles show discrepancies with the equivalent body-conforming simulations, especially for pressure, due to the rasterization effect, i.e. a loss of information regarding the exact interface location and orientation. In order to tackle these weaknesses and improve the orders of accuracy, correction schemes for both velocity and pressure were implemented to impose consistent boundary conditions on the immersed interface. Although a second order accuracy is not achieved locally at all boundary points, the corrections have been shown to increase both the accuracy of the results and the order of convergence in comparison to a simple Penalty approach, and to minimize the rasterisation effect inherent to Cartesian grids. The computational results presented here confirm that the IBM can produce integrated as well as local velocity and pressure values that are in close agreement with experimental data and standard body-fitted CFD simulations. The orders obtained with the current penalty approach with boundary treatments are comparable to previously published sharp high-order IBM. For example, the penalization method of Introïni et al. [60] leads to a global order of accuracy around 1.88 for the Taylor-Couette problem for both L_∞ and L_2 norms, while global orders between 1.27 and 2.31 are obtained for the flow around a static cylinder, or between 1.51 and 1.91 for the flow around a rotating cylinder. Similarly, the sharp IBM of Gilmanov et al. [43] reaches 1.48 and 1.74 as a global order for the norms L_∞ and L_2 respectively when studying the steady flow in a cubic lid-driven cavity. One can also mention that already existing high-order IBMs can achieve perfect second-order or above for specific test cases ([9], [111], [85]).

Shortcomings have also found to be present with specific correction schemes. Solutions were investigated in order to overcome these weaknesses. A detailed conclusion is proposed below:

1. The three types of velocity correction scheme developed or implemented during this research have been found to give very similar results both in terms of accuracy and order of convergence. Similarly, there was no difference found regarding the stability of these methods considering their use for geometries with discontinuous curvature like the simplified impeller case.

2. Although improving the estimation of integrated quantities, all three correction methods lead to a virtual thickening of the immersed solid, when applied in the outer layer of cells. This issue has been also encountered by Vincent et al. [143], when specific accurate accounting of the fluid-solid interface is involved. However the latter method does not rely on a linear scheme for the reconstruction of the velocity near the interface. In [143] the viscosity is penalized instead of the velocity. A solution to get rid of the thickening effect would be merely to decrease the size of the solid obstacle. This solution has been tested for the oscillating cylinder case, with a reduced diameter $D^* = D - h$ (h being the cell size). The velocity profiles at $x = 0.6D$ are presented in Figure 4.22. Despite a relative difference of 5% for the Reynolds number, the discrepancies in the velocity profiles appear to be significant between the two diameters. The size reduction improves significantly the results, in accordance with the body-fitted solution. Although not as accurate, a local refinement around the cylinder with cells halved in size leads consistently also to better profiles estimation and supports the argument regarding the efficiency of the anti-thickening solution. Similar improvements have been seen on the different profiles for both velocity and pressure. This type of size correction has been employed by Vincent et al. [143] as well.

   The origin of the thickening effect lies certainly in local inconsistencies between the linear reconstruction of the velocity and the real velocity profile of an attached laminar boundary layer. In the case of a fully-developed laminar boundary layer over a flat plate, the velocity follows a parabolic law as a function of the distance

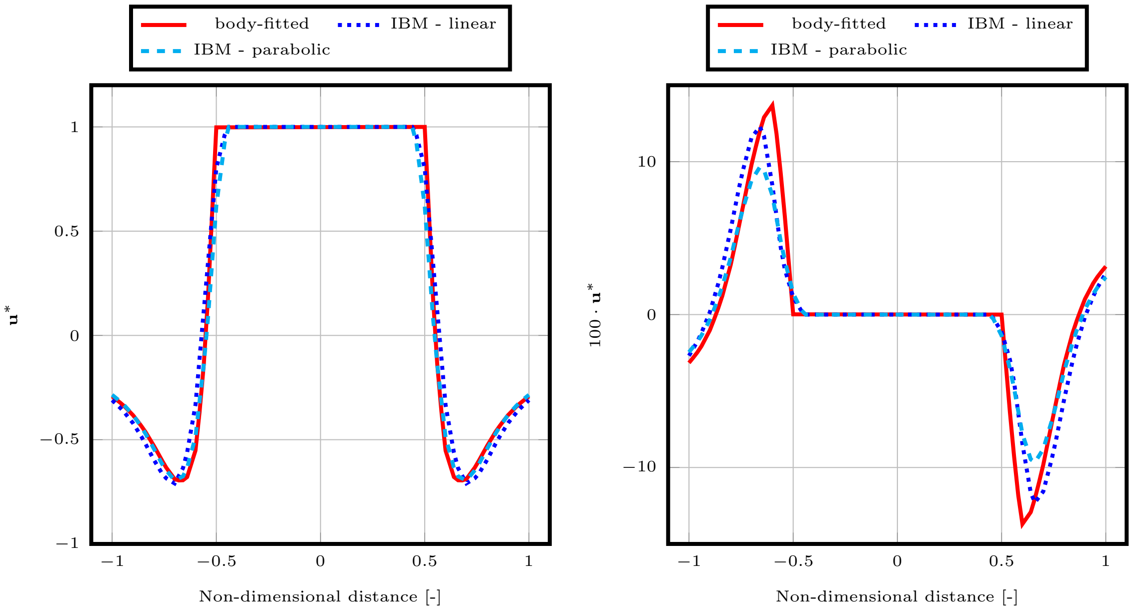


(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 4.22: Velocity profiles at $x = 0.6D$ with a phase angle $\omega t = 180°$ for a real size and a scaled cylinder. Results obtained with Grid 2

to the wall. For this reason a velocity correction scheme based on a parabolic law was implemented and compared to the standard linear correction for the oscillating cylinder case. The parabolic scheme is based on the power-law scheme introduced in Section 3.3.4 for turbulent cases, using instead a coefficient $k = 1/2$. Obviously, the relevance of the parabolic profile is decreasing when we deal with curved surfaces and non-fully developed boundary layers. The use of such a profile is even more questionable after the separation, in the recirculation zone behind obstacles. Furthermore the consistency of one or another reconstruction scheme is expected to depend also on the grid resolution. Figure 4.23 shows the velocity profiles for both type of correction schemes, along a sampling line crossing the cylinder in its center, which is located before the separation. The profiles of the streamwise component clearly indicate an improvement of the results when using the parabolic reconstruction. In this case, the minima of velocity are shifted towards the center, in accordance with the body-fitted results. This highlights a net reduction of the thickening effect. Although the magnitude is one order smaller for the transverse component, the results of Figure 4.23(b) show instead a deterioration of the profile obtained with the parabolic law. The same conclusion can be drawn for the profiles in front of the cylinder. At the back of the immersed obstacle however, that is namely in the recirculation zone, the results of the parabolic reconstruction are showing a deterioration of both velocity components. To conclude, these results support the argument regarding the source of the virtual thickening on a relatively coarse grid. There is however no strong argument to support the use of a specific correction scheme for the velocity. It is yet surprising that this issue is not mentioned in literature, while the majority of the correction schemes are based on linear interpolation and to the author's knowledge parabolic schemes have never been used in such context.



(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 4.23: Comparison of the velocity profiles at $x = 0D$ with a phase angle $\omega t = 180^\circ$ for the two velocity reconstruction schemes. Results obtained with Grid 2

3. The linear correction of the velocity does not induce important thickening when it is applied in the first layer of cells internal to the solid domain. While better velocity profiles were obtained in comparison to a simple penalty approach, a local deterioration of the pressure was identified near the immersed interface for most of the test cases, as well as a reduction of the order of accuracy in some cases. For this reason, the correction of the pressure at the immersed surface appears even more important when an inner velocity correction is considered.

4. The first boundary treatment investigated for the pressure was a decoupling scheme, which aims at removing the contribution of spurious pressure from the solid domain to the fluid domain. The decoupling scheme shows unstable behaviour when applied with an inner velocity penalization, most probably because of an increase in the numerical stiffness of the resulting pressure matrix. In the case of an outer velocity correction, the decoupling scheme does not influence significantly the solution. Two possible reasons are identified to explain this: (i) the outer velocity penalization does not deteriorate the pressure as strongly as an inner correction scheme (as it was seen for instance on Figures 4.11(a) and 4.11(b)), and (ii) the potential improvements of the decoupling scheme could be balanced by its inherent rasterization of the interface (checker-boarding representation).

5. The explicit penalization of pressure in order to satisfy a Neumann condition was found to improve both integrated quantities and local velocity and pressure profiles in the vicinity of the immersed boundaries as well. These improvements are more significant when an inner velocity correction scheme is used for the reason mentioned in (3). For this approach, an important shortcoming was however identified: the explicit correction leads to pressure fluctuations upstream of the immersed solid, which increase with the proximity of other boundaries. When the domain boundaries are far from the immersed boundaries, as it is the case for most academic cases, such spurious fluctuations are not visible. It is also important to note that despite the presence of spurious pressure fluctuations in the conditions

identified before, the pressure distribution on the immersed surface appear to be correctly predicted (and with a better accuracy compared to the simple Penalty method), as was seen from the evolution of the drag coefficient in Section 4.4.

6. The use of an implicit penalization method for the pressure boundary condition at the immersed interface can remove the spurious pressure fluctuations with adequate tuning of the implicit penalization coefficient. However, the implicit correction is generating a non-symmetric matrix with a strong diagonal, which has been found to be unstable for realistic applications involving relatively complex geometries as described for instance in the next chapter. A possible solution for this stability issue could be to implement a virtual pressure boundary condition in the internal solid domain. This condition will not have a physical meaning but could solve the stability issue by generating a symmetric matrix at the end.

Chapter 5

Results and discussion: Assessment of xIBM for engineering and multi-physics problems

Similarly to the previous chapter, second order schemes are used for the space discretization of the cases considered here. The second order Gaussian integration is used for gradient and laplacian operators. For the convection term in the momentum equation, the Linear-Upwind Stabilized Transport (LUST) scheme is used, in which linear-upwind is blended with linear interpolation to stabilize the solution while maintaining a second-order behaviour [1]. A non-orthogonal correction is added to the surface normal gradients evaluated at the cell faces in order to account for the potential non-orthogonality of the grid in some body-fitted configurations. Regarding the time discretization, the second order backward implicit scheme is used. For this type of transient problem involving high speed flows, the PIMPLE scheme is used to solve the pressure-velocity coupling, instead of the PISO scheme introduced before. The PIMPLE scheme is a combination of both PISO and semi-implicit method for pressure-linked equations (SIMPLE) schemes.

5.1 Single-blade impeller centrifugal pump

5.1.1 Presentation of the case

In order to compare pumps with geometrically dissimilar impellers, the specific speed is introduced. This non-dimensional parameter reads:

$$n_s = \frac{N\sqrt{Q}}{(H_p)^{0.75}} \tag{5.1}$$

where N is the rotational speed (in rpm) and Q and H_p are respectively the flow rate and the hydraulic head delivered by the pump. The flow rate and the head are usually taken at the Best Efficiency Point (BEP). The specific speed for this single-blade pump is $n_s = 56.4$.

The pump considered here is submersible and single-stage, with volute casing. Figure 5.1 shows a cross-section of this mechanical system. The feature of this pump is the ability to drive flow with a single blade impeller. Such an impeller is able to handle liquids with stringy materials and large solids, without screening, while maintaining good hydraulic performance.

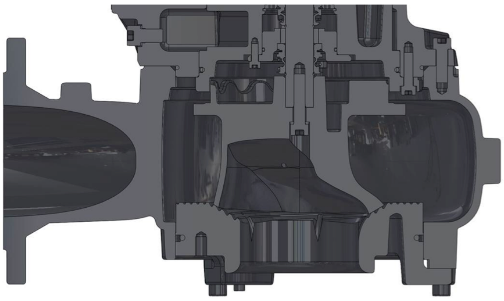


Figure 5.1: Cross-section view of the single-blade pump.

In order to reproduce the testing conditions for pump performance, which are taking place in a water tank and performed following the ISO 9906 Gr2 Annex A1/A2 Testnorm, a specific computational domain was chosen as shown in Figure 5.2. This allows a more relevant comparison with experimental data. In this framework, the entry of the pump is linked to a cylindrical sump of diameter $D_s/D = 8$, where D is the pump diameter.

The inlet of the computational domain is then defined by the top surface of this tank, where the flow enters vertically downward. The outlet of the domain remains the pipe outlet. The distance between the pump inlet pipe wall and the bottom surface modelled as a wall is $H/D = 1.17$. An experimental investigation was conducted at Sulzer Pump Solutions Irelands test facilities to confirm the lack of sensitivity of the pump head to the sump to the open tank depth/radius ratios. Finally, in order to drive the flow, the desired flow rate is simply imposed at the inlet. No-slip conditions for velocity and zero pressure gradients are fixed for all volute and tank walls.

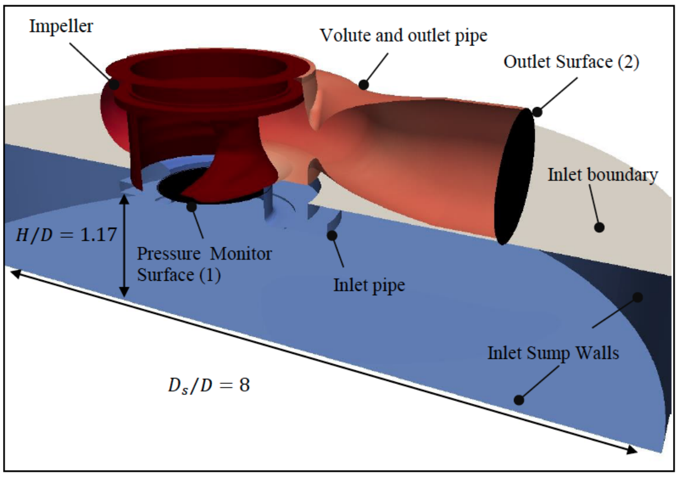


Figure 5.2: Computational domain and boundaries for the pump model.

The flow characteristics for the current case are listed in Table 5.1. The main data extracted from the simulations is the pump head, which is governing the hydraulic performance of the pumps. The pump head H_p can be obtained from a balance of energy, which reads for isothermal steady incompressible flow as:

$$\frac{p_2}{\rho g} + z_2 + \frac{V_2^2}{2g} = \frac{p_1}{\rho_f g} + z_1 + \frac{V_1^2}{2g} + H_p - h_l \tag{5.2}$$

where losses h_l which may include leakage, friction or mechanical losses, have been included. The total losses can be estimated from the mechanical power delivered by the pump, which is a function of the torque. In this work however, the losses have been subtracted from the hydraulic head, whose calculation only depends on the difference of static pressure and elevation. In equation 5.2, the subscripts 1 and 2 denote two locations along a streamline. In our case, location 2 denotes the outlet of the pump where the pressure is fixed, while location 1 obtained from a surface averaging over a cross-section of the inlet pipe below the impeller (as illustated in Figure 5.2). As the inlet and outlet pipes where the pressure is monitored have the same diameter, the pump head finally reads:

$$H_p = \frac{(p_2 - p_1)}{\rho g} + (z_2 - z_1) \tag{5.3}$$

Equation 5.3 assumes hydraulic losses to be negligible. The hydraulic power P_h delivered by the pump can be calculated in terms of the volumetric flow rate and the head, as defined in Equation 5.4 (left). The mechanical input power of a pump is measured in terms of the input torque M on the shaft and at a specific rotational speed Ω. It is also referred to as the coupling power P_c and is defined in Equation 5.4 (right).

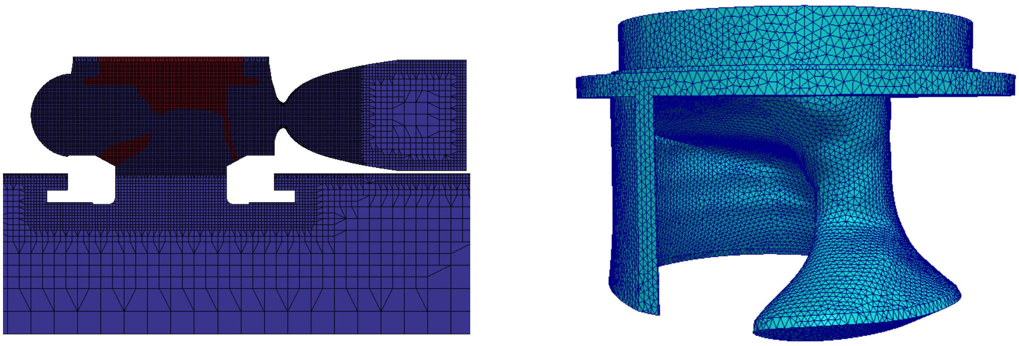
$$P_h = \rho g H_p Q \qquad\qquad P_c = M\Omega \tag{5.4}$$

The torque M can be approximated from pressure and viscous forces acting on the impeller, in this case mechanical losses due to seals and bearings are neglected. The hydraulic efficiency of a turbomachine is finally defined as the ratio between the hydraulic (or useful) power over the power associated to the coupling: $\eta = P_h/P_c$.

Fluid Properties (water at 20°C)	Density Dynamic viscosity	$\rho_f = 998kg/m^3$ $\nu_f = 108 \times 10^{-5}kg/m.s$
Universal constant	Acceleration of gravity	$g = 9.81m/s^2$
Pump layout	Elevation from inlet to outlet Inlet/outlet pipe diameter	$z_2 - z_1 = 0.077m$ $D = 0.15m$
Boundary conditions	Impeller rotational speed Outlet pressure	$\Omega = 150.8rad/s$ $p_2 = 0Pa$
Flow conditions	Reynolds number at outlet Volumetric flow rate	$Re_2 \in [1.6, 7.1] \times 10^5$ $Q \in [20, 90]l/s$

Table 5.1: Fluid, flow and geometrical characteristics for the single-blade pump model.

For the IBM solution, a mesh based on a regular Cartesian grid snapped to the inlet sump and volute boundaries, is generated to discretize the whole fluid domain (visible in Figure 5.3(a)), including the zone swept by the impeller. The presence of the impeller is accounted for as a surface mesh. Removing the impeller boundaries from the grid drastically simplifies the meshing stage, making it possible to use cubic cells everywhere except at wall boundaries. The impeller surface is meshed with triangular faces as illustrated in Figure 5.3(b). The cell size inside the volute is $2mm$, which is also the average size of the triangular faces for the surface mesh. This setting leads to an Eulerian mesh of about 2 millions cells in total.



(a) Cross-section of the Eulerian fluid mesh. The cells inside the fluid are colored in blue, while the cells inside the IB representing the impeller are colored in red.

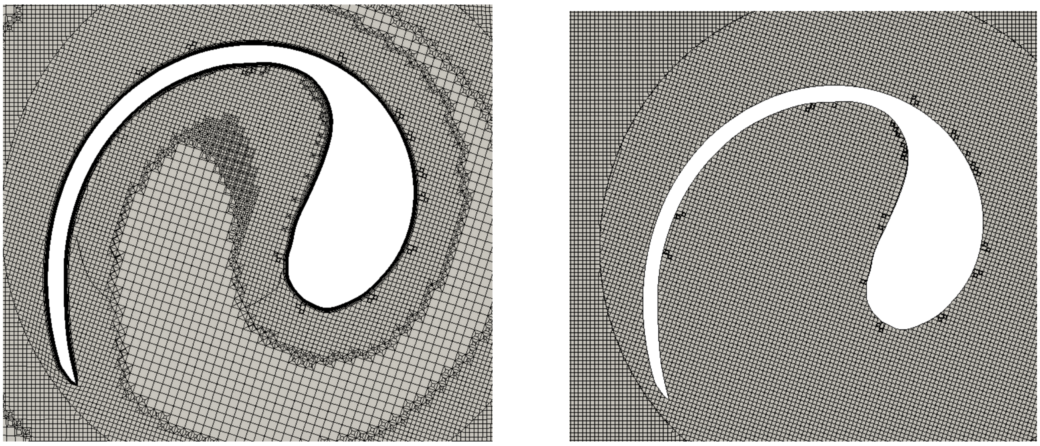
(b) Geometry and Lagrangian surface mesh of the single-blade impeller.

Figure 5.3: Grids used for the single-blade pump modelled with IBM.

In addition to the experimental data, the results from the IBM are compared to equivalent body-conforming simulations. This allows a comparison of local flow features, which are not available with experiments. The pump is modelled using the General Grid Interface (GGI) approach which handles two mesh parts: one for the rotor and one for the stator. The two parts of the mesh are based on cubic cells with edges aligned with the coordinate axes everywhere, except at the interface between the two zones and in the vicinity of the inflation layers at wall boundaries, where tetrahedral and pyramidal cells are used to make the transition. Two types of body-fitted mesh were considered in this research, which are shown in Figure 5.4. The difference between the two mesh lies in the use of an inflation layer at the impeller wall. As such a grid feature is extremely computationally expensive with IBM, those two body-conforming grids make it possible to (i) get a more consistent comparison between body-fitted and IBM simulations, and in the same time, to (ii) quantify the influence of the inflation layer in the prediction of specific local variables or global quantities for this type of engineering application. Similar to the IBM, the cell size in the core of the domain is fixed to $2mm$. The body-fitted grid with inflation layers includes a coarser region with $4mm$ in order to limit the total number of cells. The mesh characteristics are summarised in Table 5.2. The distribution of y^+ indicates that the cells adjacent to the impeller surface lie primarily within the inner boundary layer and within the logarithmic region for Mesh A and Mesh B respectively. For this reason, a wall function based on the Spalding law [122] is used for all body-fitted simulations, which ensures a correct transition of the velocity profile between the different layers and defines the turbulent variables up to the wall.

Mesh	**Impeller Blade y^+**	**Cell count (millions)**
Mesh A	4.0	2.1
Mesh B	235	4.7

Table 5.2: Mesh characteristics.



(a) Mesh A: grid using an inflation layer at the walls. The cell size is $2mm$ close to them and around the GGI interface, and $4mm$ otherwise.

(b) Mesh B: uniform body-fitted grid with a constant $2mm$ cell size.

Figure 5.4: Cross-section of the body-conforming grids used for the single-blade pump.

5.1.2 Results and discussion: without turbulence modelling

Turbulence modelling is necessary to produce realistic and accurate predictions of the flow through the centrifugal pump considered here, given the Reynolds number for this case. However, this assessment will first consider a laminar case to focus on an assessment of the velocity penalization by comparison with body-fitted results. The results from the simple Penalty approach are compared to the inner and outer velocity correction schemes. The sensitivity to the pressure treatment is investigated in Section 5.1.2.

General results

The experimental head and torque are used to normalize the equivalent integrated quantities obtained from simulations. The time is non-dimensionalized by the period of the impeller cycle. Figure 5.5 shows the head evolution obtained with the body-fitted and different penalty approaches at a flow rate of 40 l/s. The transient head is averaged using a moving average on 20 time steps, over a time span of 1 ms. These results indicated that approximately steady statistical characteristics are obtained after five revolutions from static initial conditions. All the penalty approaches are shown to overestimate the head by around 5% when compared to body-fitted results. Few differences can be noted between the penalty approaches. The head is slightly higher when the outer velocity correction is used instead of the inner velocity correction. As a result the inner correction scheme provides head oscillations that are closer to the body-fitted solution. This is consistent with conclusion from Chapter 4 which suggested that the outer velocity correction lead to a thickening of the boundary layer. With the pump simulation this should translate to a larger effective impeller and hence to a higher average head. The benefit of the inner velocity correction is especially visible just after the discharge part of the cycle, when the head is low in accordance with the body-fitted head. One can also remark that the body-fitted simulation lead to high frequency pressure oscillations, which are most visible during the low and high phases of the pressure cycle. The source of the oscillations can lie in both the use of the Arbitrary Mesh Interface (AMI) method for the moving mesh and the presence of skewed cells.

The time averaged numerical pressure heads obtained for the 4 models are listed in Table 5.3. The averaging is performed over five revolutions after the five initial cycles of the transient. Although the overestimation of the head is the largest with outer velocity correction, this estimate is the closest to the experimental quantity. This could be due to compensating errors and it is important to note that the effect of turbulence is not taken into account at this stage. Furthermore, the analysis focuses here on the comparison against body-fitted simulations.

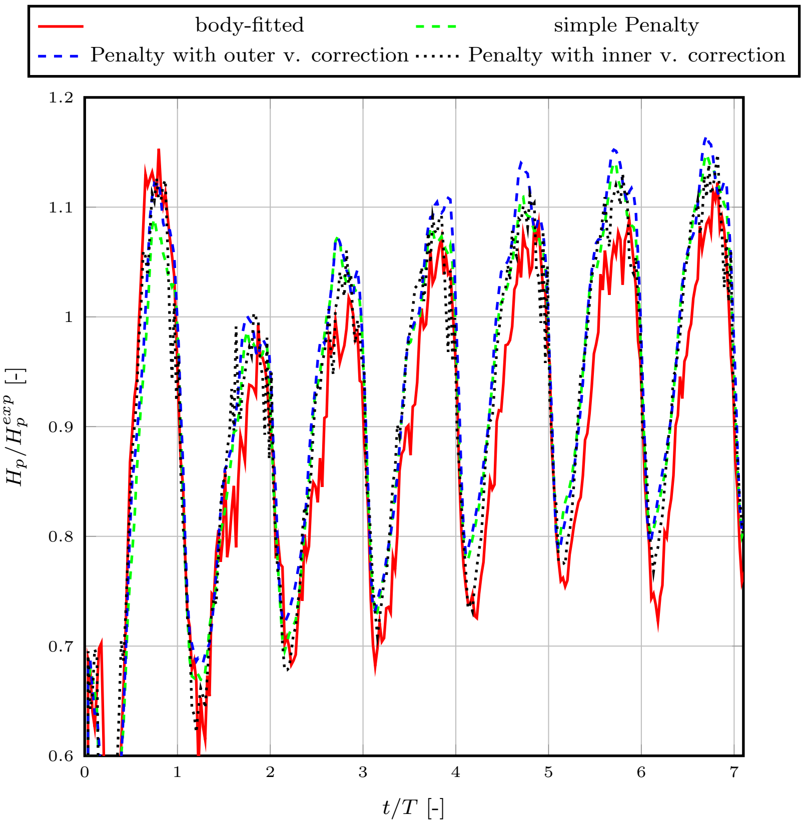


Figure 5.5: Evolution of the hydraulic head of the single-blade pump for different penalty approaches in the laminar case ($Q = 40l/s$).

The velocity fields predicted by the body-fitted approach and the three previous IBM are shown in Figure 5.6 at a time $t = 0.2s$. The general flow features are shown to be well reproduced in terms of the maxima and minima of some of the key trends. The IBM tends however to overestimate mixing in the volute as illustrated by the higher

Model	H_p/H_p^{exp} [-]
Body-fitted	0.93
Simple Penalty	0.99
Penalty with outer v. correction	1.01
Penalty with inner v. correction	0.98

Table 5.3: Numerical hydraulic heads obtained for the body-fitted case and different IBMs.

spacial variability of the velocity. The source of this discrepancy lies certainly for the IBM in the difficulty to capture the quasi absence of separation at the trailing edge. In that regard, the two penalty approaches with velocity correction give better results. The simple penalty approach generates in addition spurious vortices in the suction side of the pump, close to the impeller surface. This is corrected as well with the velocity correction, although some small fluctuations are still present with the inner velocity correction (far right figure). Finally the IBM with inner velocity correction, which reduces the virtual thickening effect, leads to a slightly lower velocity magnitude compared to the two other IBM, and in accordance to the body-fitted results. Although improvements in the prediction of the trailing edge wake can be seen from the penalty methods with inner velocity correction, the conclusion is less clear when we look at the vorticity field for instance.

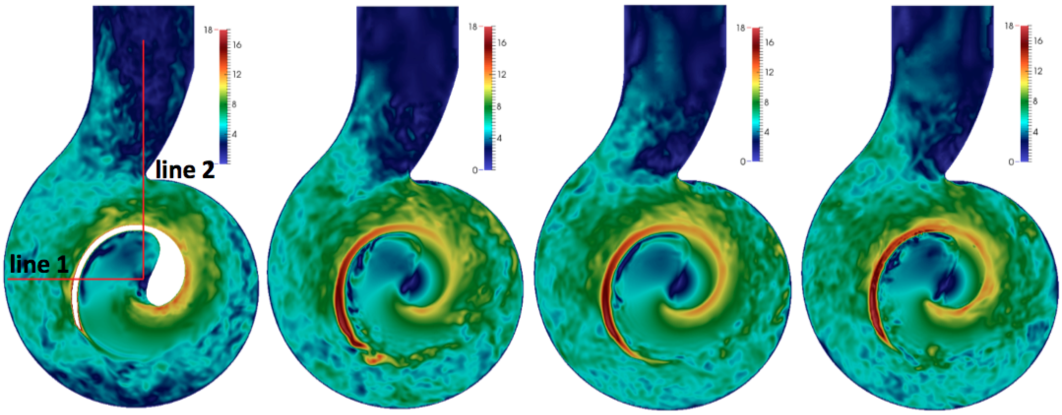


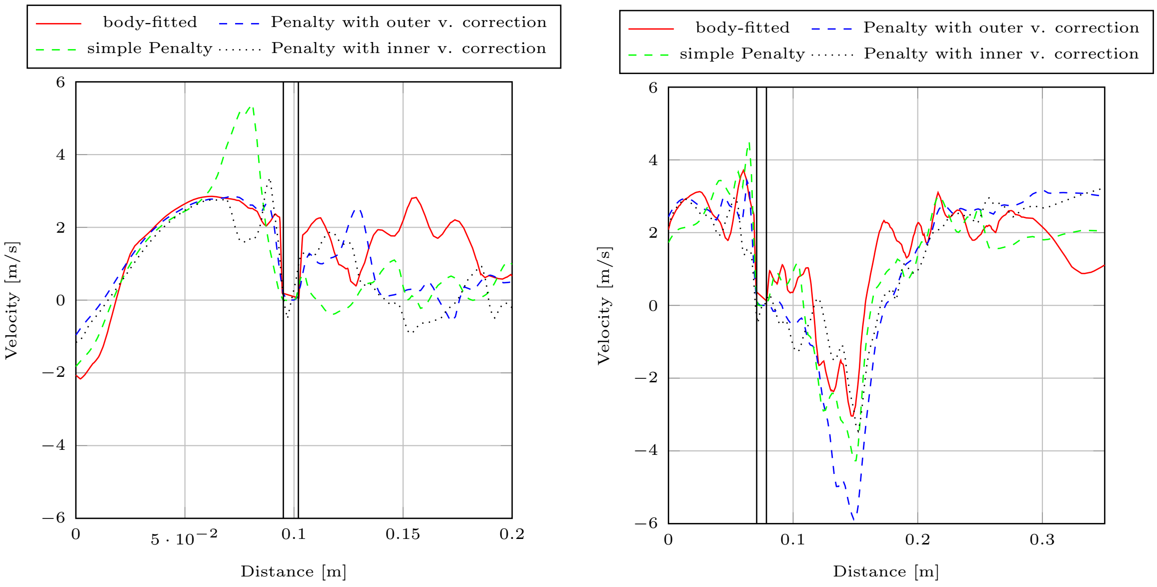
Figure 5.6: Contour of velocity magnitude (m/s) in an horizontal plane section of the pump (located 8 mm below the outlet center point) at $t = 0.2s$. From left to right: body-fitted, simple penalty (no correction), outer velocity correction, inner velocity correction.

The influence of the method of velocity penalization including the type of correction scheme is assessed here by reference to local velocity along two sampling lines (shown in red in Figure 5.6 (left)). These are located in the core region of the volute and have been selected to cover both impeller suction side, where the flow is mainly axial, and discharge side, where the flow is mainly tangential. The lines are starting from the center of rotation and are locally close to normal to the impeller pressure side. The profiles are plotted in Figures 5.7, 5.8 and 5.9, in which the two vertical black lines show for the impeller suction and discharge surfaces. It is difficult to conclude quantitatively on the results of the different IBM because of the high mixing in the volute. However, from the amplitude and the wavelength of the velocity fluctuations on the discharge side, one can assume that the average size of the vortices is similar for both IBM and body-fitted methods. A Fast Fourier Transform (FFT) analysis of the fluctuations in space was performed for the radial velocity on the pressure side of line 1. For each numerical method, three main wavelengths were identified. They are listed in Table 5.4. The three wavelengths obtained with the body-fitted approach correspond to periodic fluctuations in space of respectively $5.95cm$, $3.66cm$ and $2.16cm$. A similar range of fluctuation is predicted with IBMs. One can note the good agreement with the simple Penalty approach for the largest wavelength and with the Penalty approach including inner velocity correction for the medium wavelength. These results support the argument regarding the similar size of vortical structures obtained in both types of CFD methods.

Model	**W1** $[m^{-1}]$	**W2** $[m^{-1}]$	**W3** $[m^{-1}]$
Body-fitted	16.8	27.3	46.2
Simple Penalty	20.0	33.6	44.2
Penalty with outer v. correction	13.7	33.9	56.6
Penalty with inner v. correction	13.4	29.8	51.4

Table 5.4: Mains wavelengths (W1, W2 and W3) obtained from an FFT of the radial velocity on the suction side of line 1.

In general, one can note also a good prediction of the boundary layer at the impeller surface, which is particularly noticeable for the tangential velocity on Figure 5.8(a). The velocity peak obtained for the simple Penalty method on the suction side (see Figure 5.7(a)) reflects for instance the increase in mixing along the impeller suction surface with this method. Concerning the velocity corrections, the recirculation zone near the cutwater in the volute seems to be better predicted with the inner correction, as suggested by the radial velocity in Figure 5.7(b) at a distance of 0,15 m from the volute center. A better agreement is also found on the suction side of the pump for the axial velocity, in comparison to the simple Penalty method.



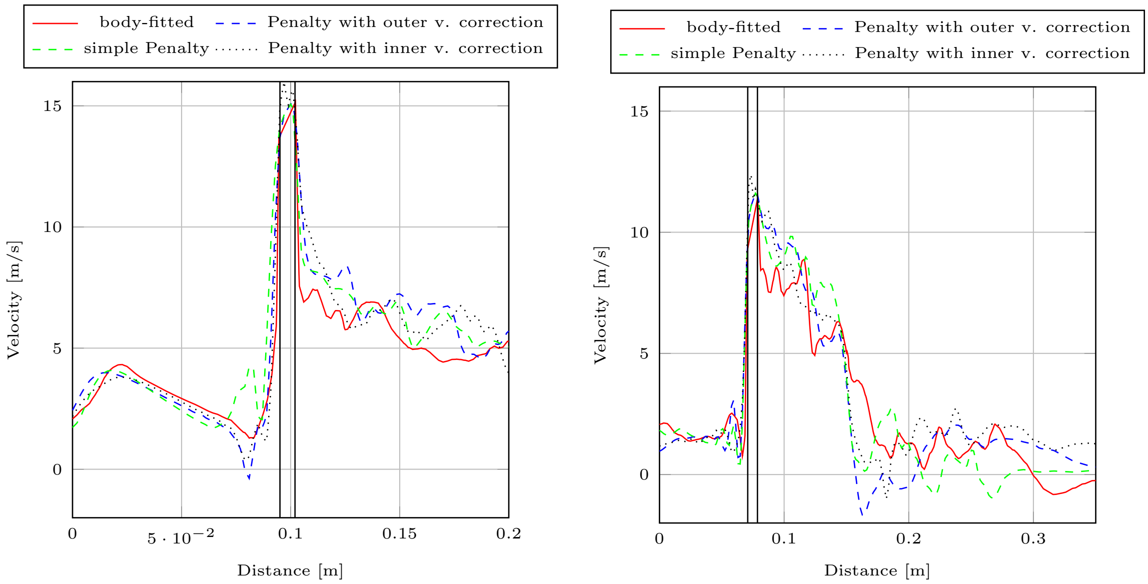
(a) Profile along line 1

(b) Profile along line 2

Figure 5.7: Profiles of the radial component of velocity for different IBM at $t = 0.2s$.

Sensitivity to pressure corrections

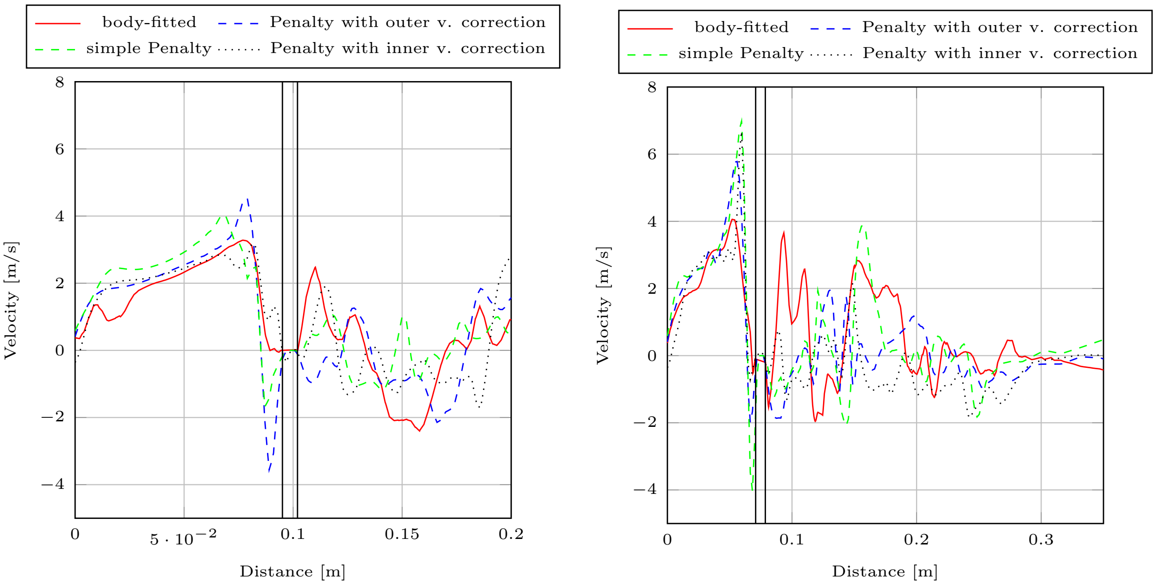
The influence of the pressure correction on the velocity field in the volute has been shown to be relatively small. However, this penalty based correction of the pressure plays a role in the pressure solution, and thus the hydraulic head. In Figure 5.10, the contours of pressure are presented firstly for the body-fitted approach, and the penalty method with outer velocity correction, and secondly for the penalty method with inner velocity correction, without and with pressure correction. It appears that the iso-contours of



(a) Profile along line 1

(b) Profile along line 2

Figure 5.8: Profiles of the tangential component of velocity for different IBM at $t = 0.2s$.

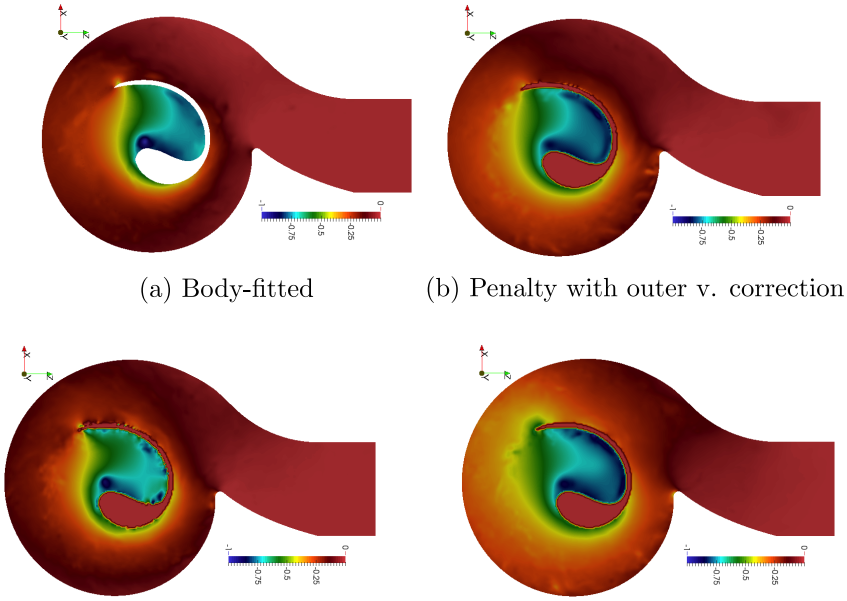


(a) Profile along line 1

(b) Profile along line 2

Figure 5.9: Profiles of the axial (vertical) component of velocity for different IBM at $t = 0.2s$.

the pressure obtained with the body-fitted method and the IBM are well matched. Although the iso-contour of pressure are also correctly estimated with the inner velocity



(a) Body-fitted (b) Penalty with outer v. correction

(c) Penalty with inner v. correction (d) Penalty with inner v. & p. correction

Figure 5.10: Contour of non-dimensional pressure in an horizontal plane section of the pump (located 8 mm below the outlet center point) at $t = 0.2s$.

correction, one can see spurious pressure fluctuations along the suction surface of the impeller, which reflect the presence of small vortices as seen also in Figure 5.6. This is due to the difficulty for the pressure field to account for the high velocity gradients between the solid velocity, the extrapolated velocities (from inner correction), and the surrounding fluid velocities. This phenomenon is amplified here in comparison to the simpler test cases of the previous chapter because of the complex geometry, and the high angular velocity of the solid. These pressure errors are shown to have a low influence on the overall head. When the pressure is penalized as well in order to satisfy a Neumann condition, the pressure is then corrected near the suction surface of the impeller, as shown in the last picture of Figure 5.10. In this case however, the iso-contours are shifted in the volute, and the magnitude of pressure is higher, resulting in an instantaneous hydraulic head 14% higher than the body-fitted one, and whose evolution shows large fluctuations. The fluctuations are also observed when the pressure correction is combined with the outer velocity correction.

The strong fluctuations around the mean head, which are observed with the pressure correction at the interface, are shown in Figure 5.11. The main cause for these fluctuations has been identified in the previous chapter as the proximity of the domain boundaries. In the current case, the volute walls where the zero pressure gradient is imposed is adjacent with cells which are inside the solid or penalized domains. Increasing the minimum distance between the volute walls and the cells where the pressure correction is applied is decreasing the magnitude of the pressure fluctuations. Here, a minimum distance of 4 cells width was considered. In this case, the pressure fluctuations deteriorate slightly the mean head, which remains in a range of 12% difference with the body-fitted prediction. This is still less satisfactory than the Penalty methods including only velocity corrections at the immersed walls.

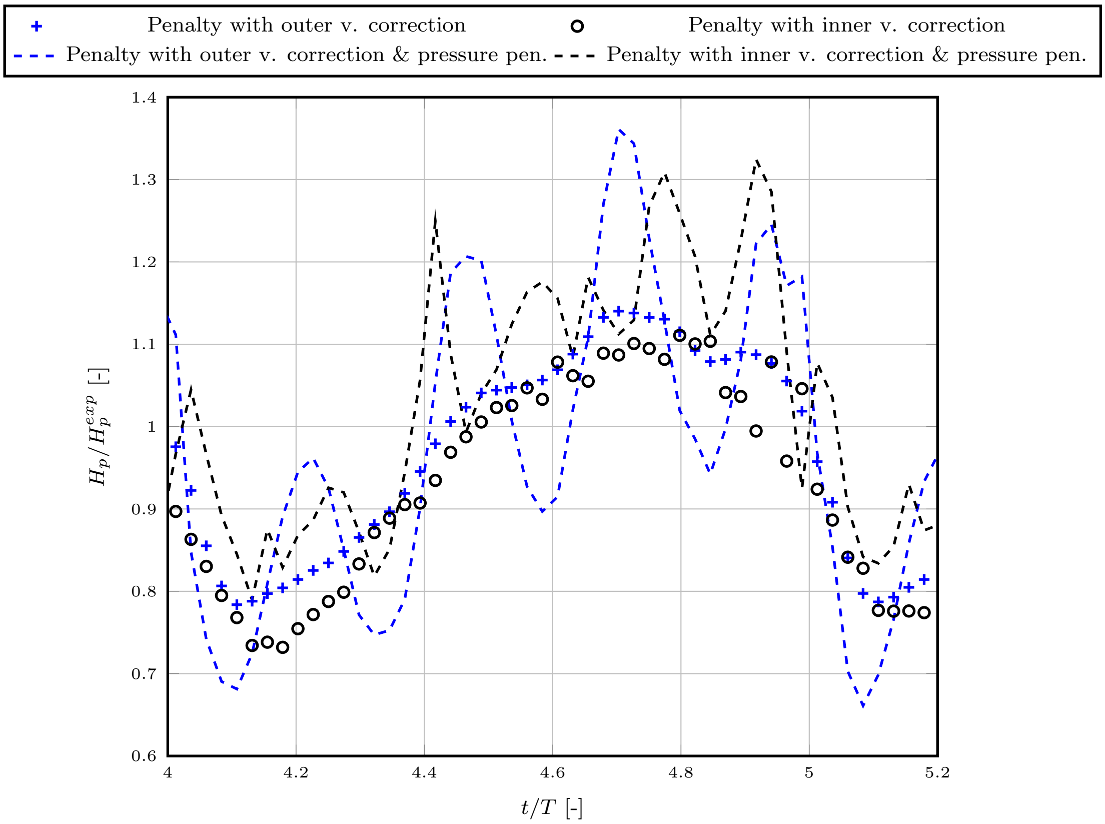


Figure 5.11: Sensitivity of the head to the pressure correction.

5.1.3 Results and discussion: with turbulence modelling

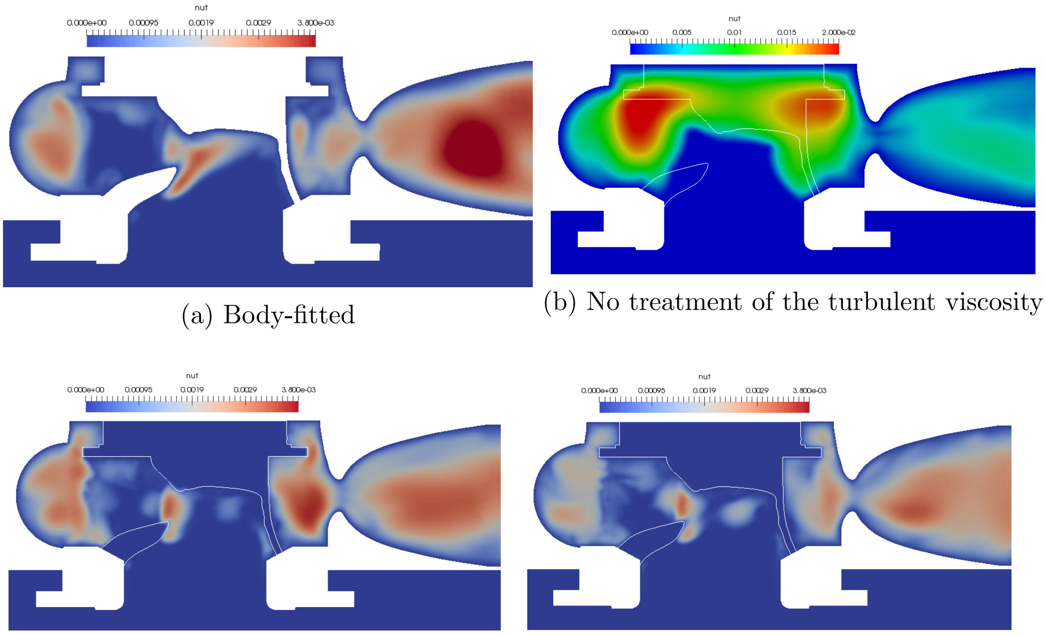
This section presents the assessment of the full model, extended to include turbulence modelling. This is necessary to allow comparison against experimental data as well. Two turbulence IBM formulations are considered: (i) the RANS Spalart-Allmaras model with penalization of the turbulent viscosity at the immersed interface (pSA), and (ii) a hybrid DES model based on pSA for near wall modelling. The hybrid model will be assessed by comparison against both experimental and pSA simulations.

As highlighted in the previous section, the pressure correction scheme leads to spurious pressure fluctuations in time. For this reason, the results presented here are obtained with the penalty approach including only the velocity reconstruction at the immersed interface, if not stated otherwise. The velocity correction is applied internally, i.e. in the first layer of cells inside the solid.

Immersed Boundary formulation of the Spalart-Allmaras model (pSA)

As a first step, the effect of the IBM formulation of the Spalart-Allmaras (SA) model is investigated. The IBM formulation is based on a penalization of the turbulent viscosity, in order to satisfy a Dirichlet condition on the immersed surface. As was the case with the no-slip velocity condition, the penalization of the turbulent viscosity can be either first order, or can include a linear reconstruction scheme (interpolation). The same correction will be applied with RANS and DES models, since in both cases wall adjacent cells are treated using the pSA model.

Figure 5.12 shows the turbulent viscosity field in a central cross section of the pump as predicted by the body-fitted solution and the IBM with and without any viscosity penalization. It is clear from these results that the penalization has a strong effect and is necessary to avoid excessive turbulent stresses. When no treatment of the turbulent variable is taken into account, the turbulent viscosity is shown not to be affected by the immersed surface. The production of effective viscosity in the SA model is determined in terms of $\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}$, where Ω is the magnitude of the vorticity. The production term in the SA equation (see Equation 3.37 in Section 3.3.1) reads:



(a) Body-fitted

(b) No treatment of the turbulent viscosity

(c) Penalization of the turbulent viscosity for a Dirichlet condition on the solid surface (first order)

(d) Penalization and linear reconstruction of the turbulent viscosity for a Dirichlet condition on the solid surface (second order)

Figure 5.12: Contour of the turbulent viscosity (m^2/s) in a cross section of the pump at $t = 0.2s$. The interface of the impeller is shown as a white line for the IBM cases.

$$P = c_{b1}\tilde{S}\tilde{\nu} \tag{5.5}$$

The increased vorticity generated by the high impeller rotation in the solid domain is thus leading to a non-physical overestimation of the production term. The dissipation term in the SA equation is a function of the distance d_w to the wall and reads:

$$D = c_{w1}f_w(\frac{\tilde{\nu}}{d_w})^2 \tag{5.6}$$

If the distance to the closest wall is not corrected to take into account the immersed surface, the dissipation term is then under-predicted. The combination of these two effects lead to a turbulent viscosity which is one order of magnitude higher than with the body-fitted simulation, when the presence of the immersed boundary is not taken into account within the SA model. On the other hand, when the turbulent viscosity is penalized and the distance to the closest wall corrected, the magnitude and distribution of the viscosity field show clearer similarity between the IBM and the body-fitted simulations. Regarding the pattern in the turbulent viscosity field, one can note in particular a good agreement for the region of high viscosity in the back of the volute, and the region of low viscosity in the region of the discharge side above the impeller's leading edge.

The evolution of the head is plotted in Figure 5.13 for the standard body-fitted model and the three formulations of IBM with the different treatments of the turbulent viscosity. As previously, the hydraulic head is time averaged with a moving average over a time period of 1 ms. The higher eddy viscosity predicted when no treatment is applied impacts significantly on the head estimation. The large values of turbulent viscosity are leading to higher modelled Reynolds stresses. Since the flow rate through the pump is imposed as an inlet boundary condition, the increased dissipation due to higher Reynolds stresses translates into higher pressure gain across the pump. The penalization of the turbulent viscosity is found to give average head gains that are closer to experimental data and also closer to body-fitted predictions. Finally, the additional correction, based on interpolation, further improves both comparisons. The mean heads for all numerical simulations are listed in Table 5.5 as a ratio of the experimental heads. The table summarizes also the mean torque acting on the single-blade impeller for each case. One can see that the penalization of the turbulent viscosity improves also the estimation of the torque compared to body-fitted results. So does the second-order reconstruction of the turbulent viscosity in comparison to the first-order reconstruction. Despite the good agreement between the numerical and experimental heads, the numerical torque are largely under-estimated in comparison to the experimental one. This is the case with both IBM and body-fitted simulations. Given that the pressure provides the main contribution to the torque (i.e. viscous shear stress makes a negligible direct contribution to the total torque), it is clear that both the body-fitted and IBM simulation fail to reproduce the exact pressure solution in this case.

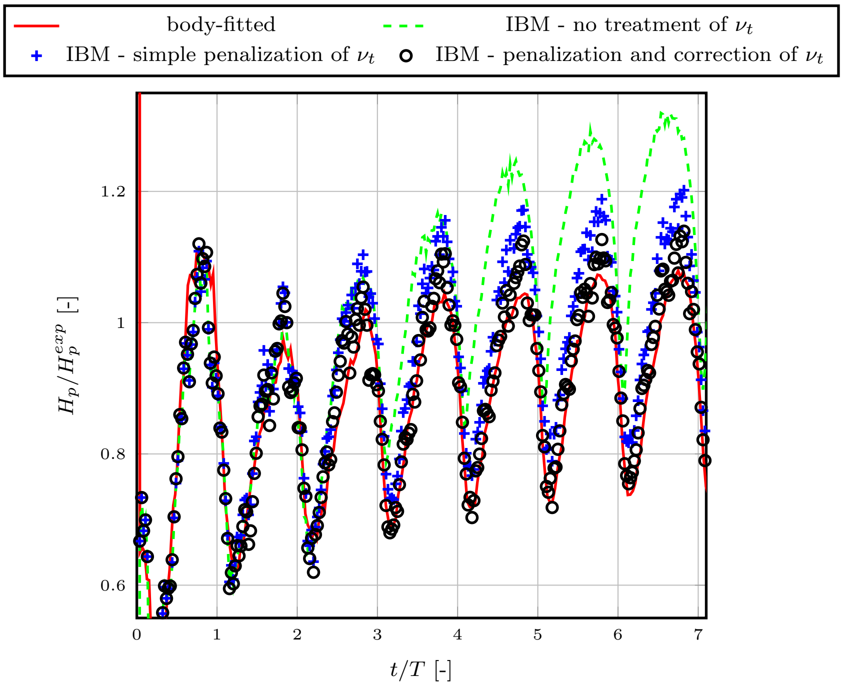


Figure 5.13: Evolution of the hydraulic head of the pump at $Q = 40l/s$, for different treatments of the turbulent viscosity ν_t with the SA model.

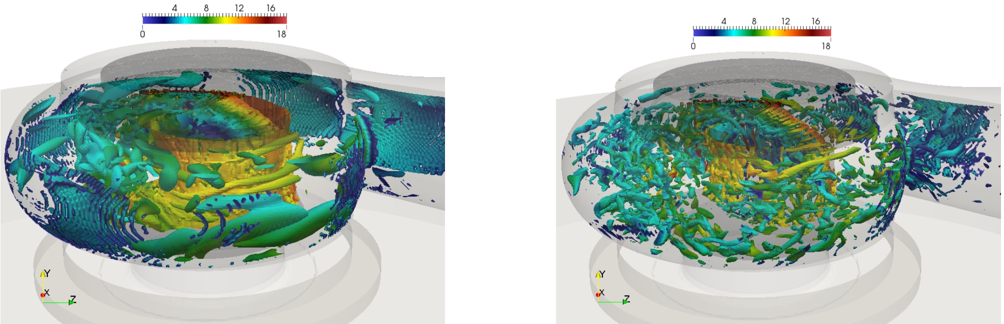
Model	H_p/H_p^{exp} [-]	M/M^{exp} [-]
IBM - no treatment of ν_t	1.18	0.36
IBM + simple penalization of ν_t	1.02	0.70
IBM + pen. and corr. of ν_t	0.96	0.71
Body-fitted	0.93	0.75

Table 5.5: Numerical heads and torques obtained for the body-fitted case and different IBMs.

Hybrid RANS-LES modelling with IBM

Although the differences in the averaged hydraulic head between the full RANS model and the hybrid turbulence models remain within 1% at the flow rate considered ($40l/s$), significant differences can be seen in the structure of turbulent vortices. These differences are illustrated with the iso-contour plot of the Q field in Figure 5.14. Q is defined by $1/2(\Omega^2 - S^2)$, where $S = \|1/2(\nabla\mathbf{u} + \nabla^T\mathbf{u})\|$ and $\Omega = \|tr(\nabla\mathbf{u})\|$ are the strain rate and vorticity rate magnitudes, so that positive iso-contours represent vortices defined as a region where the vorticity is greater than the magnitude of the strain rate. With equivalent mesh, the iso-contour of Q must be reduced by one order of magnitude to visualize vortices from the Spalart-Allmaras model. In addition, the dominant vortices are wider and their stretching remain mostly horizontal in this case, while finer structures with a much broader range of scales and shapes are predicted with a scale resolving DES computation.

The iso-contours of the Q field show as well the generation of rolls along the suction surface of the impeller, which are not present with a body-conforming mesh. They appear to be the result of insufficient mesh resolution of the impeller surface when IBM is used. The size of the rolls can be indeed reduced with mesh refinement. With a finer mesh around the impeller surface, the turbulent structures in the volute remain very similar, underlining the low influence of these rolls on the main flow.



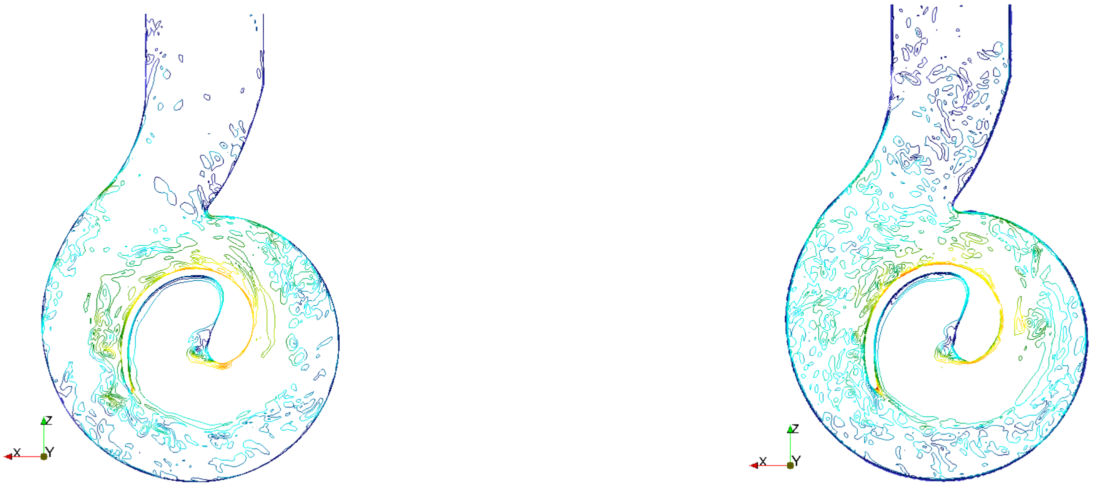
(a) $Q = 10^4$ obtained with a full RANS Spalart-Allmaras model

(b) $Q = 10^5$ obtained with the hybrid DDES model

Figure 5.14: Iso-contour of the Q field obtained with IBM colored by the velocity magnitude (m/s), in the center part of the volute at $t = 0.2s$.

Given their aim and formulation, IBMs cannot incorporate a fine body-conforming mesh suited to a boundary layer flow. In order to assess the impact of poorly resolving the boundary layer, the influence of the inflation layers on the numerical solution is analysed and presented here for body-fitted grids. Figure 5.15 exhibits the different intensity of mixing obtained for the two types of body-conforming grids introduced earlier and illustrated by the iso-contours of the vorticity. One can see the absence of turbulent structures on the suction side in both cases. However, the absence of inflation layers

(Mesh B) appears to break the relative uniformity of the flow behind the trailing edge, which results in a higher mixing on the impeller discharge side. Similar observations were made from the IBM solution. The higher resolution of the boundary layer however does not significantly impact the integrated pump characteristics (pressure and torque). Between the two body-fitted grids, the relative difference is below 5% for the hydraulic head and below 1% for the torque. This result supports the argument that the resolution of the boundary layer is not of primordial importance for applications like pump design, involving highly rotating flows without clear onset of a turbulent boundary layer. In this context, IBM can be seen as a suitable solution in spite of the absence of inflation layers at the immersed wall. Finally, the relative difference in the numerical results is below 2% between the IBM and the body-fitted computations with Mesh B for both the head and the torque.



(a) With mesh A

(b) With mesh B

Figure 5.15: Iso-contours of the vorticity magnitude Ω (from $0s^{-1}$ to $3000s^{-1}$) coloured by the velocity magnitude in an horizontal plane section of the pump (located 8 mm below the outlet center point) at $t = 0.2s$, obtained with the two types of body-fitted grid.

Performance Curve

The hydraulic heads obtained numerically for different flow rates are summarized in the performance curve of Figure 5.16, and compared to the experimental heads. The IBM tends to under-estimate the head. This underestimation grows with the flow rates. While a very good agreement with experimental heads can obtained at low flow rates and around the Best Efficiency Point (BEP), the difference reaches 28% for the highest flow rate tested. The standard body-fitted approach is under-estimating the actual head, and the difference grows with the flow rate. Except for the highest flow rate, IBM gives a better head estimation than the body-conforming method. It is difficult to draw conclusions from this result, which can follow on from compensating errors. Finally, as the head losses are not modelled here, one can expect the CFD models to overestimate the hydraulic head. The fact that this is not the case shows that the head losses are actually relatively small relative to the numerical errors introduced by the discretization.

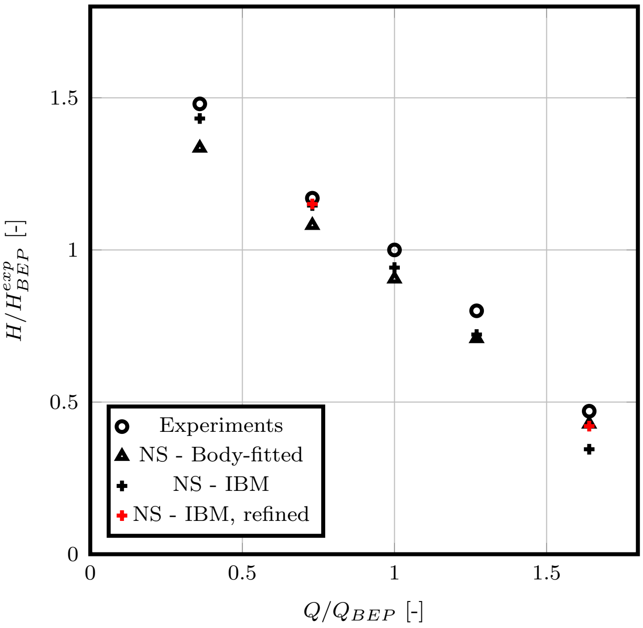
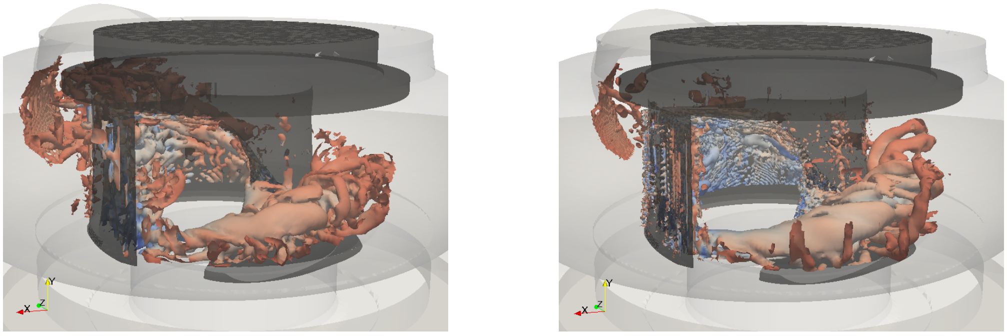


Figure 5.16: Head vs. flow rates: comparison between numerical methods considered in this research and experimental values provided by Sulzer. All data are normalized by the experimental head at the BEP.

The poor IBM estimate at high flow rate can be corrected with a finer grid. The refined results were obtained with adaptive refinement along the immersed boundary interface. In the refined area, the cell size is halved to 1mm, while the general cell size in the volute remains around 2mm. Accurate estimates of the head gains over the full range of flow rates can be achieved with this adaptive refinement. The difference with the body-fitted predictions is below 10% for the highest flow rate with the fine mesh, while it is was about 28% with the coarser grid. The difficulties in having a good solution at high flow rate is illustrated in Figure 5.17, which shows a comparison of the Q field obtained with the standard grid (Figure 5.17(a)) and the dynamically refined grid (Figure 5.17(b)). The elongated vortex which forms over the impeller at the lower leading edge is caused by flow separation at high flow rate and is predicted by both grids. However, the vortical structures on the suction surface of the impeller appear to be much smaller with the refined grid.



(a) $Q = 10^5$ obtained with the standard grid

(b) $Q = 10^5$ obtained with the standard grid + adaptive refinement

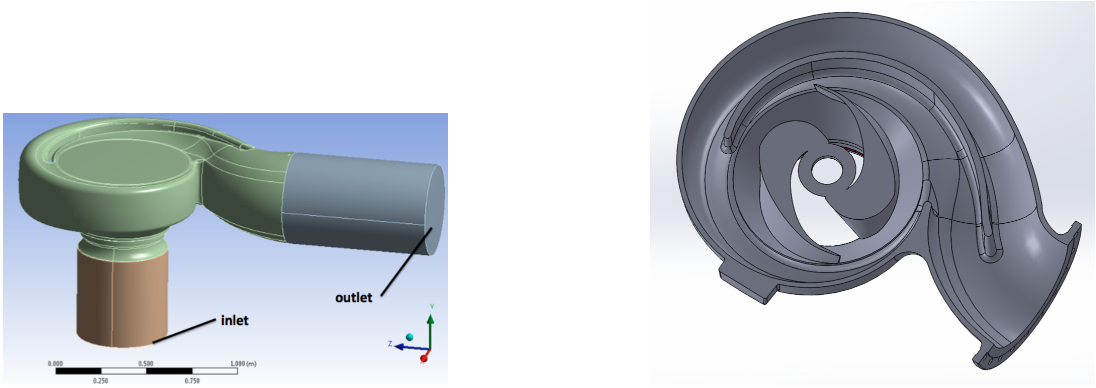
Figure 5.17: Iso-contour of the Q field colored with the pressure magnitude, around the impeller suction surface and leading edge at $t = 0.2s$.

5.2 Two-blade impeller centrifugal pump

The second pump case focuses on a two-blade pump developed by Sulzer. In addition to the higher range of Reynolds numbers delivered by this pump, the main difference compared to the previous pump lies in the impeller design. The specific speed of the two-blades pump is relatively close the single-blade one, with $n_s = 50.6$ in this case.

5.2.1 Presentation of the case

Similar to the previous system, this pump is submersible and single-stage. However, the volute is cut in two by a splitting vane. The CAD geometry of the pump is shown in Figure 5.18. For sake of simplicity, the inlet and outlet boundaries of the domain are defined as sections of the inlet and outlet pipes. The flow characteristics for the two-blade impeller pump are listed in Table 5.6.



(a) Full computational domain.

(b) Cross-section of the pump showing the design of the splitting vane and the impeller.

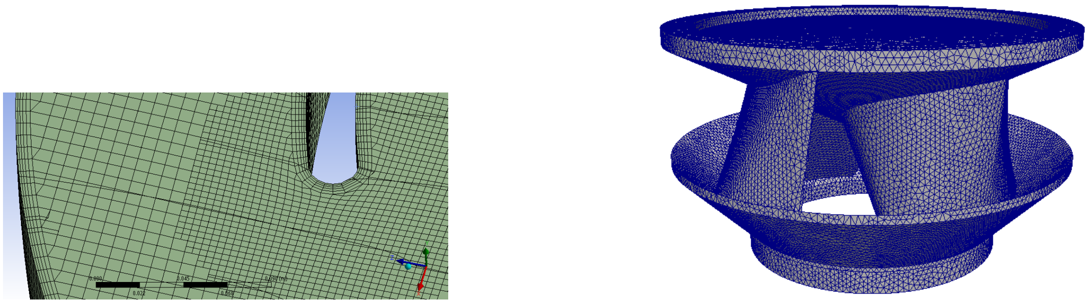
Figure 5.18: Rendering view of the two-blade impeller pump.

Figure 5.19 shows the fluid and the solid grids used for this pump model. The surface of the impeller modelled with IBM is meshed as usual with triangular cells. While the whole pump domain including the impeller sub-domain is meshed with a regular Cartesian grid. Inflation layers are added at the volute walls. Three different grids are considered and their characteristics are listed in Table 5.7. The difference between

Fluid Properties (water at $20°C$)	Density	$\rho_f = 998kg/m^3$
	Dynamic viscosity	$\nu_f = 108 \times 10^{-5}kg/m.s$
Universal constant	Acceleration of gravity	$g = 9.81m/s^2$
Pump layout	Elevation from inlet to outlet	$z_2 - z_1 = 0.4m$
	Inlet/outlet pipe diameter	$D = 0.5m$
Boundary conditions	Impeller rotational speed	$\Omega = 77.45rad/s$
	Outlet pressure	$p_2 = 0Pa$
Flow conditions	Reynolds number at outlet	$Re_2 \in [7.0, 32] \times 10^5$
	Volumetric flow rate	$Q \in [301, 1370]l/s$

Table 5.6: Fluid, flow and geometrical characteristics for the two-blade pump model.

the fine grid (Mesh 3) and the medium grid (Mesh 2) lies only in local refinements applied near sudden change of the pump geometry, where high velocity gradients can be expected. These refinements are achieved by dividing cells into uniform octants. An example of local refinement is shown in Figure 5.19(a) at the upstream extremity of the splitting vane.



(a) Zoom of the Eulerian fluid mesh on the extremity of the splitting vane.

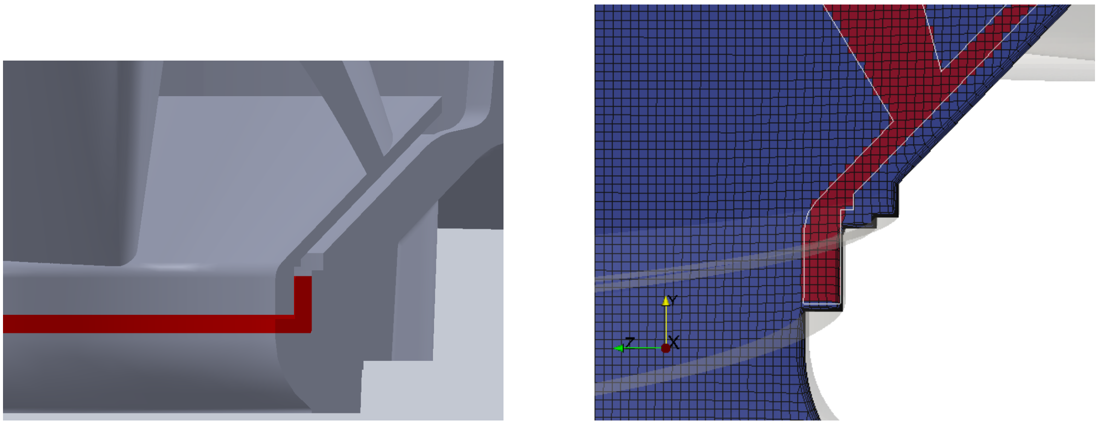
(b) Geometry and Lagrangian surface mesh of the two-blade impeller.

Figure 5.19: Grids used for the two-blade pump model.

Grids	**Cell size [mm]** **(core volute)**	**Cell size [mm]** **(inlet & outlet)**	**Inflation** **layers**	**Cells in total** **[millions]**
Mesh 1	8	16	$1mm$ / 5 layers	2.5
Mesh 2	5	10	$0.5mm$ / 5 layers	10
Mesh 3	5	10	$0.5mm$ / 8 layers	17

Table 5.7: Characteristics of the grids for the two-blade impeller pump.

It is important to note the small thickness of some parts of the impeller. The thickness of the impeller shroud plate is visible for instance in Figure 5.20. Considering the medium or the fine grid, thin solid regions can be 2 or 3 cells thick only. This is also the case for the blade trailing edge. Figure 5.20 is a zoom view of the pump model around the junction between the impeller shroud plate and the volute casing. The aim of the impeller seal (in red in Figure 5.20(a)) is to reduce leakage flow through the gap between the impeller and the pump housing. In reality, such leakage might still be present even with the seal and can contribute to the losses in the energy balance. Given the mesh resolution of our CFD model, this cannot be simulated. However, as it was already shown for the previous case, the losses can be expected to be small in comparison to the uncertainties from CFD.



(a) Geometry and design

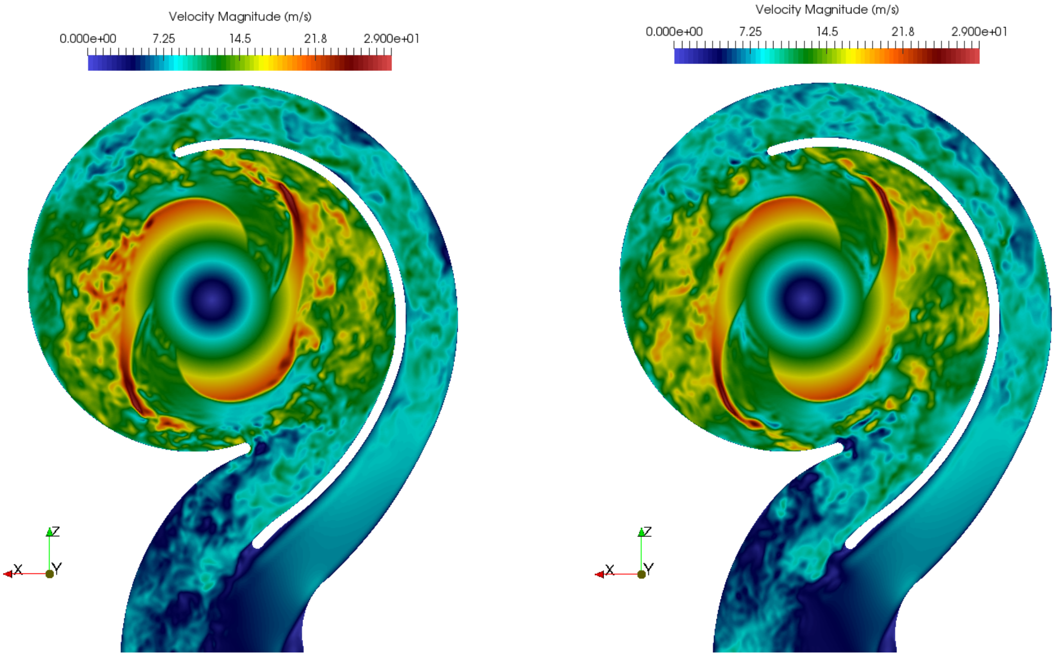
(b) Eulerian fluid mesh

Figure 5.20: Zoom on the junction between the impeller shroud plate and the volute casing. Left: the impeller seal is shown in red. Right: the cells inside the fluid domain are shown in blue, while the cells inside the solid domain (impeller) are shown in red.

5.2.2 Results and discussion

Figure 5.21 shows the magnitude of the velocity in an horizontal cross section of the pump. Passing through the center of the outlet pipe, the cross section is neither cutting the shroud plate nor the upper plate of the impeller. Only the two blades are thus visible in this cross section. The results shown in this figure are obtained with Mesh 2. Light but noticeable differences can be noted between the simple penalty approach and the approach with velocity and pressure correction at the immersed interface. One can note in particular that with the improved penalty: (i) a lower level of spatial variations in the velocity magnitude which may be interpreted as lower mixing is predicted within the volute, (ii) the wake behind the trailing edges is thinner and (iii) a more uniform velocity field develops on the suction side of both blades. Similar observations can be made at different cross sections along the axis of rotation of the pump. This is also consistent with the results obtained for the single-blade pump. One can thus assume that the flow field obtained with the improved IBM is closer to what we can expect from a body-fitted simulation.

The hydraulic head for the BEP is shown in Figure 5.22 during two impeller cycles for different numerical cases. Unlike the previous pump studied, the high frequency fluctuations of the head are not present with the two-blade pump. The instantaneous values of the numerical heads are thus provided in the figure. One can see firstly that the resolution of the coarse grid (Mesh 1) is not sufficient to obtain a satisfactory estimation of the head. While the relative difference with the experimental quantity is around 19% in this case, using Mesh 2 reduces this difference to 7%. Using the finest grid however (Mesh 3), involving only local refinements from Mesh 2, does not improve again the estimation of the head. The influence of the velocity and pressure corrections at the immersed interface are also visible in the figure. The penalty approach with corrections leads to an increase of the amplitude of the head peaks by comparison to the simple penalty approach. However, this change in the shape of the head evolution is most probably due to the pressure fluctuations, which can be obtained with the pressure correction scheme, rather than a real improvement in the head estimation.



(a) Simple Penalty

(b) Improved Penalty

Figure 5.21: Velocity field in an horizontal cross section of the two-blade pump passing through the center of the outlet pipe at time $t = 0.2s$ obtained with Mesh 2.

This conclusion is supported by three results: (i) the mean head is not changed as seen in Figure 5.22, (ii) the amplitude increase of the harmonic was also found for the incorrect pressure field predicted for the static impeller case in Section 4.4.3 when wall boundaries are close to the immersed boundary, and finally (iii) the instantaneous head significantly depends on the minimum distance between the volute walls and the cells where the pressure correction is applied. More precisely, the blue curve in Figure 5.22 depicting the penalty approach with corrections is obtained with a minimum distance of 8 cells width, which excludes the impeller shroud plate. If the pressure correction is applied for the shroud plate as well, the primary and secondary peaks of the head have shown to be even higher in this case.

The influence of the velocity and pressure corrections at the immersed interface is also visible for the estimation of the torque. The instantaneous and time-averaged torque are shown in Figure 5.23 for both the simple penalty approach and the model with corrections, and for two different flow rates. Average torque values obtained experimentally by Sulzer are also used to scale the results for comparison in the figure. The correction schemes lead to an increase in the torque for both flow rates in accordance with the experimental data. This increase is more significant at high flow rate. Here, the change brought by the improved IBM are unlikely to be attributed to the pressure fluctuations as is the case for the head estimations. Indeed, whether the pressure correction is applied or not for the shroud plate, does not change the torque evolution. As the moment of the fluid force is mainly acting on the core of the impeller, i.e. on the two blades, one can thus reasonable assume that the increase in the torque is due to the velocity and pressure corrections. Furthermore, it was seen in Section 4.4.3 that the errors from the penalization scheme affecting the pressure upstream of the immersed solid, do not deteriorate the pressure distribution on the immersed surface. In the light of these results, it appears that the high-order correction schemes are certainly improving the solution locally near the immersed surface, and especially the pressure gradient, while globally the pressure may fluctuate around a mean value depending on the proximity to the other solid surfaces. This conclusion is supported by the previous investigation made for laminar problems in Chapter 4.

The performance curve of the two-blade pump is summarized in Figure 5.24. Satisfactory results are obtained for the hydraulic head. By comparing these results, with the performance curve of the single-blade pump, two differences can be noted in the trend of the numerical head as a function of the flow rates: (i) the numerical head is in good agreement with the experimental one at the highest flow rate as well, and (ii) constant under-estimation of the head is not observed, as the head obtained numerically is higher than the reference value at low flow rates. The first difference identified here is probably due to the absence of large separation with this double blade impeller, which is more difficult to accurately predict. The absence of such large trailing vortices can be seen in Figure 5.25, where only small elongated vortices are visible. This can be contrasted with large separation with the single blade pump (see Figure 5.17 in Section 5.1.3), which was found to lead to poor head estimation at the highest flow rate. Regarding the torque, although the differences obtained with the experimental quantities are still important, one can note however that the numerical estimations are in better agreement than the ones obtained for the previous pump. The discrepancy between the numerical and the experimental torque is increasing with an increasing flow rate, but the torque estimation can be corrected with the improved penalty approach as highlighted previously.

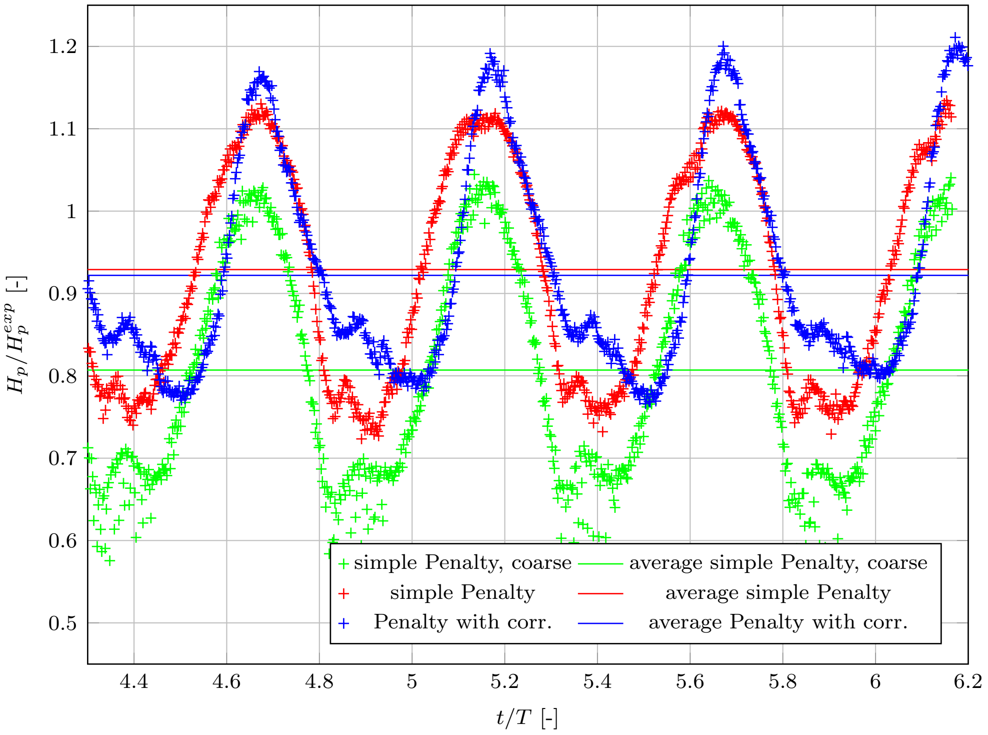


Figure 5.22: Instantaneous hydraulic head of the pump for different numerical specifications of the IBM at the BEP ($Q = 847 l/s$).

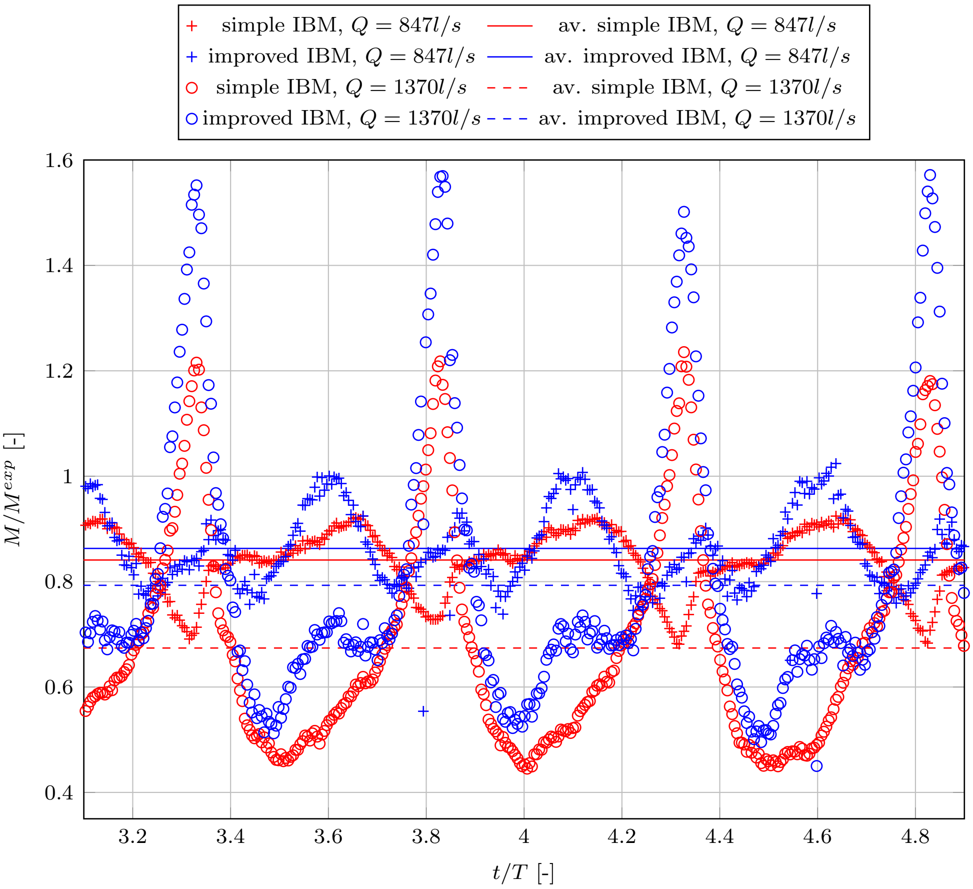


Figure 5.23: Instantaneous torque of the pump for the simple and improved penalty-based IBM for two different flow rates.

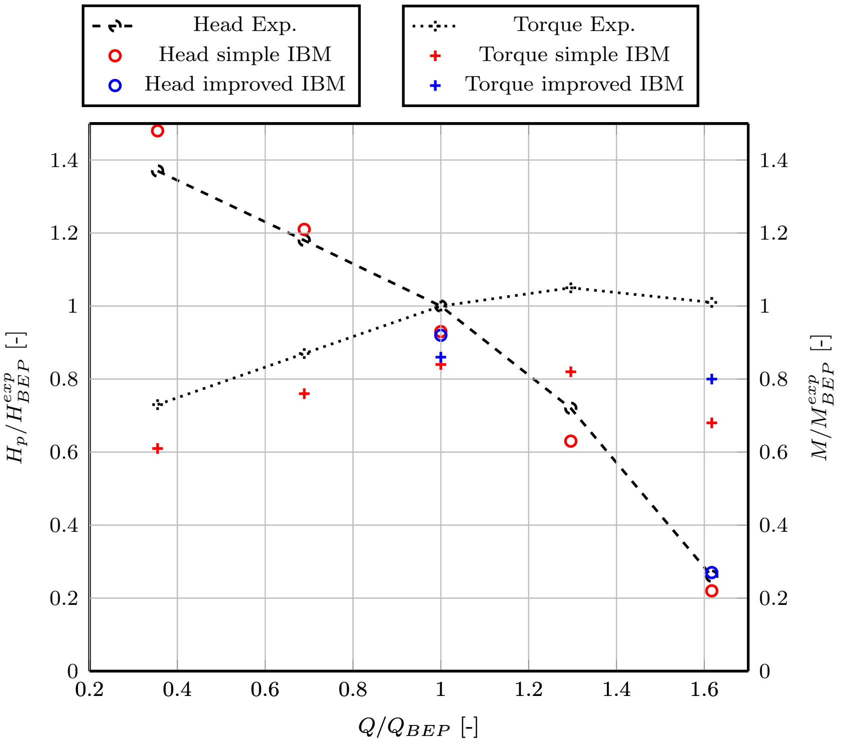


Figure 5.24: Head and torque vs. flow rates: comparison between the current IBM and experimental values provided by Sulzer. All data are normalized by the experimental head and torque at the BEP.

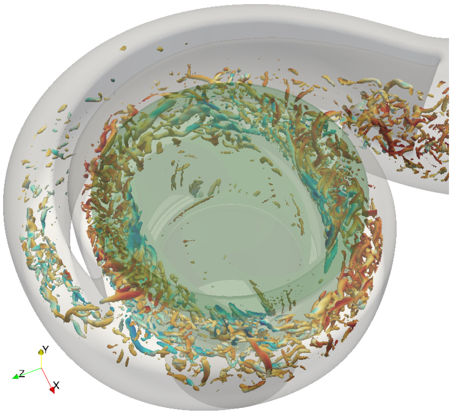


Figure 5.25: Iso-contour of the Q field ($Q = 2 \cdot 10^5$) colored with the velocity magnitude for a flow rate of $1370l/s$ and at $t = 0.5s$.

5.3 Single-phase mixing in a stirred-tank

5.3.1 Case description

This case concerns the cylindrical mixing tank with 4 vertical baffles and the 4 pitched blade impeller studied by Ge et al. [41]. The results reviewed here refer to the 45° pitch impeller with rectangular blades. The tank and impeller assembly has a 90° rotational periodicity. Ge et al. study provides both experimental and computational results. Radial and axial velocities were measured using PIV along two radial sampling lines located upstream and downstream of the impeller at axial position $y = 0.09m$ and $y = 0.13m$ and offset by 5° from the vertical plane containing the baffles. In addition, the phase averaged turbulent kinetic energy is estimated from the 2D planar PIV measurement using the pseudo-isotropic assumption: $k = \frac{3}{4}(\bar{u'}^2 + \bar{v'}^2)$, where $\bar{u'}$ and $\bar{v'}$ are the turbulent radial and axial velocity fluctuations.

The design of the problem is shown in Figure 5.26. The tank used in the experimental study was deeper than in the computational model where the height H corresponds to the level of the free surface in experiments. The impeller rotational speed is set at $N = \Omega/2\pi = 2.5rev/s$ giving the Reynolds number $Re = ND^2/\nu = 5.18 \cdot 10^4$ and the Froude number $Fr = N^2D/g \simeq 3.6$. Under these conditions distortions of the free surface by the liquid stirring can be expected to have little influence [29]. Computational tests were conducted with a Volume of Fluid (VOF) model to account for the influence of the free surface and confirmed this to be the case.

5.3.2 Meshing details and accurate modelling

The computational results from [41] were obtained from a RANS $k - \epsilon$ model and a Generalized Grid Interface (GGI) method to handle the rotating zone and a hybrid mesh. Hexahedral cells were used for the stationary zone and tetrahedral cells ranging from $2mm$ in the impeller discharge region to $6mm$ elsewhere. Zero thickness baffles and impeller blades were used to avoid meshing complication. Similarly for the body-fitted simulations performed in this research, the GGI method was used to model the

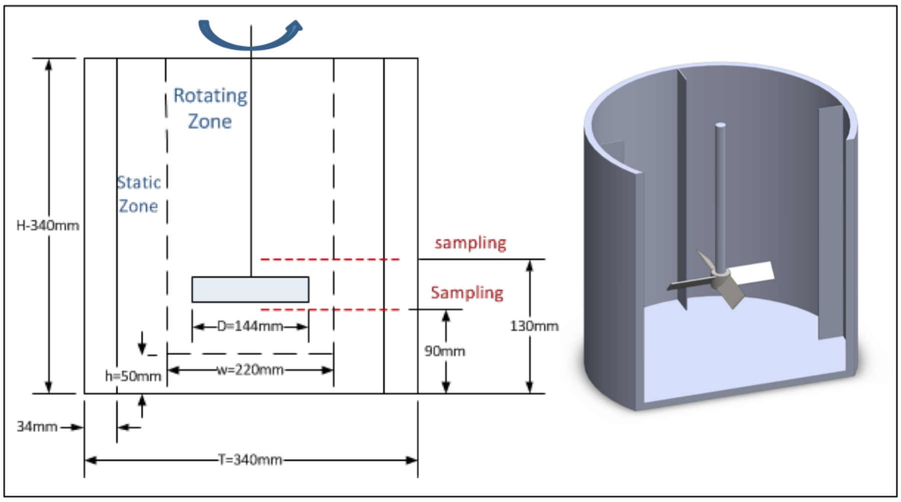


Figure 5.26: Outline of mixer layout dimensions and CAD rendering view. The static and rotating zones are specified for the body-conforming approach.

moving boundaries. A cubic mesh was employed for both the stator and the rotor mesh zones. The edges of the quadrilateral cells are aligned with the coordinate axes everywhere except in the vicinity of inflation layers at wall boundaries where tetrahedral and pyramidal cells are used to make the transition. Local refinements are achieved by dividing cells into uniform octants. The body-fitted mesh is based in the end on $2mm$ cells. A refined zone is defined in the rotating region, extending from its lowest point (see Figure 5.26) to 0.19 impeller diameter. Finally, one additional refinement is applied in four layers adjacent to wall boundaries. Meshing of the impeller wall relies on a single quad based prism layer (inflation layer). Concerning the impeller itself, a $2mm$ wall thickness was used instead of the zero thickness from Ge et al. With these settings, an average y^+ of 5.8 was obtained. A coarser mesh was also considered without the intermediate refined region. Finally for the IBM case, a similar cell size was considered in the tank. However, the refined region with $1mm$ cells is smaller in this case, ranging from $y = 0.095m$ to $y = 0.13m$. In this sense the equivalent IBM resolution is in-between both body-conforming grids. The IBM is using the physical thickness of the impeller ($1.6mm$), hence its discrete representation is only one or two cells thick. The grid details are listed in Table 5.8.

Model	Turbulence model	max. grid res.	impeller thickness
Ge et al.	RANS $k-\epsilon$	$2mm$	0
Body-fitted	DDES	$0.5mm$	$2mm$
IBM	coupled IBM-DES	$1mm$	$1.6mm$

Table 5.8: Numerical settings for the different CFD models

The impeller surface mesh is composed of triangular elements of $0.25mm$ in average, which is four times smaller than the size of the fluid cell. The approximation made for the estimation of the distance to the interface in arbitrary geometry, that is which cannot be defined analytically, is discussed here. We define the average error in the calculation of the distance between fluid cell center points and the interface as:

$$\bar{E_d} = \frac{1}{\int_{\Omega_p} d\Omega} \int_{\Omega_p} E_{d,i} d\Omega = \frac{1}{\int_{\Omega_p} d\Omega} \int_{\Omega_p} (\frac{d(\mathbf{x_i}) - d_{ref}(\mathbf{x_i})}{d_{ref}(\mathbf{x_i})}) d\Omega \qquad (5.7)$$

where $E_{d,i}$ is the local relative error in cell i between the approximate distance $d(\mathbf{x_i})$ and a reference distance $d_{ref}(\mathbf{x_i})$. The reference distance is calculated with a dichotomy algorithm using 80 iterations. The average error is integrated over the penalized domain Ω_p. The average errors obtained with the considered surface mesh and with different dichotomy iterations are listed in Table 5.9. Using 20 iterations for the dichotomy algorithm is shown to give a negligible relative error. This supports the argument that the error in the estimation of the distance, which converges in 2^k, k being the number of dichotomy iterations, is low enough to consider the distance obtained with 80 iterations as the real distance to the interface. From the table, we can see that approximating the distance to the interface by the distance to the closest solid points with a surface mesh of $0.25mm$ elements leads to an average relative error below 1%. This can be considered as acceptable for our simulation case. Furthermore, it should be noted that the approximation made here reduces the CPU time for computing the distance from $8.87s$ to $2.53s$ in comparison to a dichotomy algorithm using 80 iterations. These CPU times were obtained for a sequential simulation and the distances were estimated for all cells in the penalized domains (approximately 20000 cells in total). The same analysis on the error is performed for the Wannier case, described in Section 4.1. The medium mesh is considered (Grid 2) for the fluid domain and here also the surface mesh for the cylinder is composed of triangular elements four times smaller. The reference distance is calculated with the analytical distance to the interface. An average relative error around 2.5% was found in the penalized domain. The larger relative error obtained in this case can be explained by the use of larger volume and surface mesh.

Method	Dicho. 5 it.	Dicho. 10 it.	Dicho. 20 it.	surface mesh
Error $\bar{E}_d$	0.25	0.0059	$5.2e^{-6}$	0.0085

Table 5.9: Volume averaged relative error as defined in Equation 5.7 for different methods of computing the distance to the immersed interface.

Both body-fitted and IBM simulations performed for this study have used a hybrid RANS-LES type of turbulence model. While the IBM is coupled to a DES model (see Section 3.3), the body-fitted case used instead the Delayed Detached Eddy Simulation (DDES), which reduces incorrect behavior triggered by a grid spacing parallel to the wall smaller than the boundary-layer thickness [121]. The resolutions of the two body-fitted grids have been assessed in terms of the kinetic energy spectra. The results have shown that both mesh are fine enough to fully resolve the inertial range, with a spectrum characterized by a curve of slope $-5/3$.

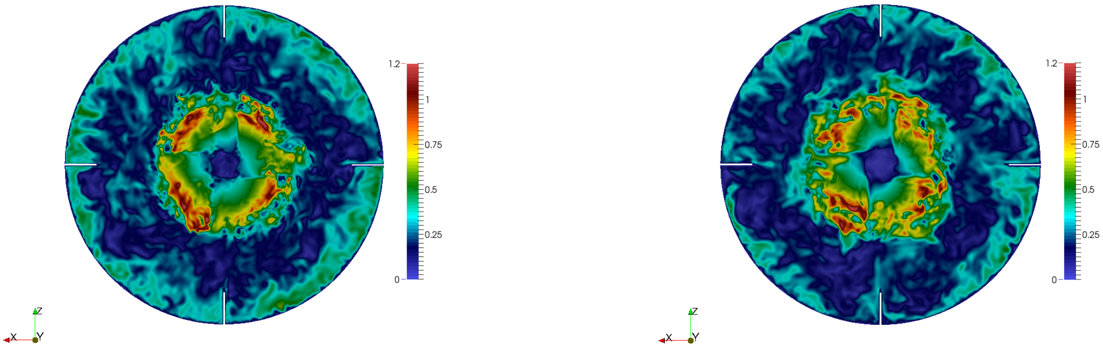
Due to the small thickness of the impeller blade relative to the mesh size, the outer velocity correction scheme is used in the present case. The explicit penalization of the pressure is used as well to ensure a Neumann condition at the interface. It was verified that the pressure fluctuations that can appear with such pressure correction scheme (see Chapter 4) are small enough and negligible when time-averaged. Given the mean y^+ obtained with the body-fitted simulation, it can be assumed that cells adjacent to the impeller boundaries will typically be placed in the logarithmic layer provided that the flow remains attached, when modelled with IBM. For this reason, the velocity correction scheme based on a power-law introduced in Section 3.3.4 is also considered in this case, in order to better describe the turbulent velocity profile of the boundary layer. The next sections present the results obtained with a linear reconstruction of the velocity, while

Section 5.3.5 discusses the sensitivity to the type of reconstruction.

5.3.3 General results

The initial conditions for the simulation were based on a fluid at rest with uniform static pressure distribution gravity being neglected. The velocity contour obtained after 5 impeller revolutions are shown in Figures 5.27 and 5.28. The accuracy of this variable is meaningful for mixing problems. In the considered areas, the mesh size is $\Delta x = 1mm$ in a circular region centered in the impeller with a radius of respectively $0.11m$ for the body-fitted case and $0.08m$ for the IBM case, while the mesh size is $\Delta x = 2mm$ outside these regions. Although the regions of high velocity magnitude look more heterogeneous directly below the impeller in Figure 5.27, the current penalization approach is leading to a similar level of mixing in the tank, as confirmed by the time averaged flow velocity profiles discussed in the next section (Section 5.3.4). The size of the dead zone directly under the impeller as well as the extent of the zone characterized by high downstream velocity are in good agreement with the body-fitted case. For the plane cutting the impeller in Figure 5.28, the extent of the region of high velocity swept by the impeller also shows generally similar characteristics in terms of spread and diameter. There is however some non negligible difference in the intensity and length of the wake trailing behind each impeller blade. The high velocity region is shown to spread further radially downstream of the impeller (at $y = 0.1m$) in the IBM case with a lower velocity magnitude, while the flow remains attached to the impeller for longer with the body-fitted mesh allowing for higher velocity flow to evolve which then diffuses more rapidly in the wake (see Figure 5.29).

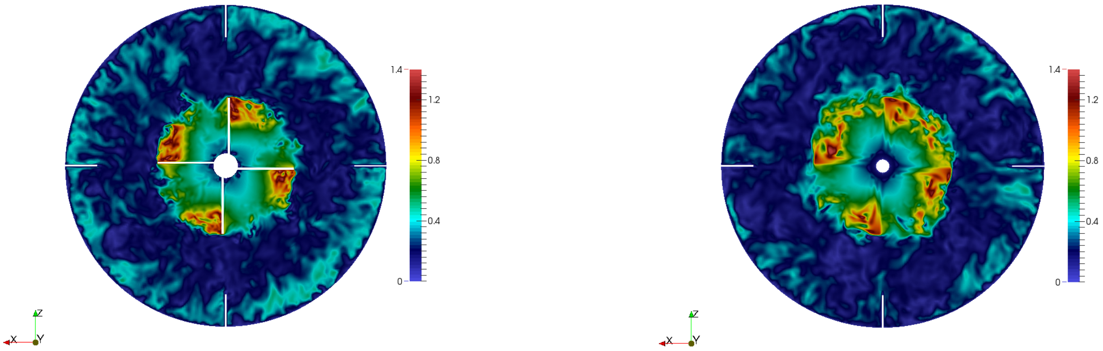
Concerning turbulence modeling, it is worth noting that the penalization of the turbulent viscosity is a necessity when using a full RANS model. Without a correct boundary condition on the immersed interface, the turbulent viscosity computed inside the solid from the Spalart-Allmaras equation can be very high, leading to an overestimated diffusion in the flow. The influence of this penalization process is however found to have a negligible influence when a DES turbulence model is used. Results confirm that with



(a) Body-fitted

(b) IBM

Figure 5.27: Contour of the velocity magnitude (m/s) on a cross-section downstream of the impeller at $y = 0.1m$ for both body-fitted and IBM computations.



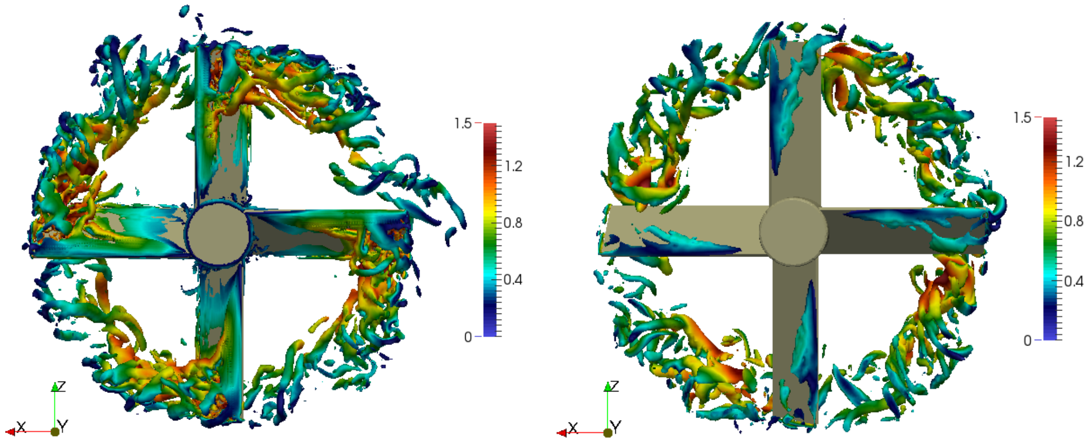
(a) Body-fitted

(b) IBM

Figure 5.28: Contour of the velocity magnitude (m/s) on a cross-section cutting the impeller at $y = 0.11m$ for both body-fitted and IBM computations.

DES models a correct estimation of the turbulent variables in the core is more important, and the latter is not significantly affected by what happens near the boundary layer, in this type of problem where there is significant repeated mixing. A possible reason is that a lot of the turbulence is determined by the interaction of larger turbulent scale with the impeller blade and subsequently the turbulent cascade. This is an important conclusion regarding the relatively low resolution of the impeller wall with the immersed boundary formulation. Figure 5.29 illustrates the similarity of the vortical structures around the impeller for both numerical methods. The absence of inflation layers in the

IBM can explain the failure of the method to resolve the boundary layer growth and separation. This clearly impacts on the formation of vortices at the impeller blade tips. The type and shape of the vortices obtained with IBM however appear to be similar, and the comparison of flow velocity profiles in Section 5.3.4 shows that this effect on time average mixing is not significant.



(a) Body-fitted

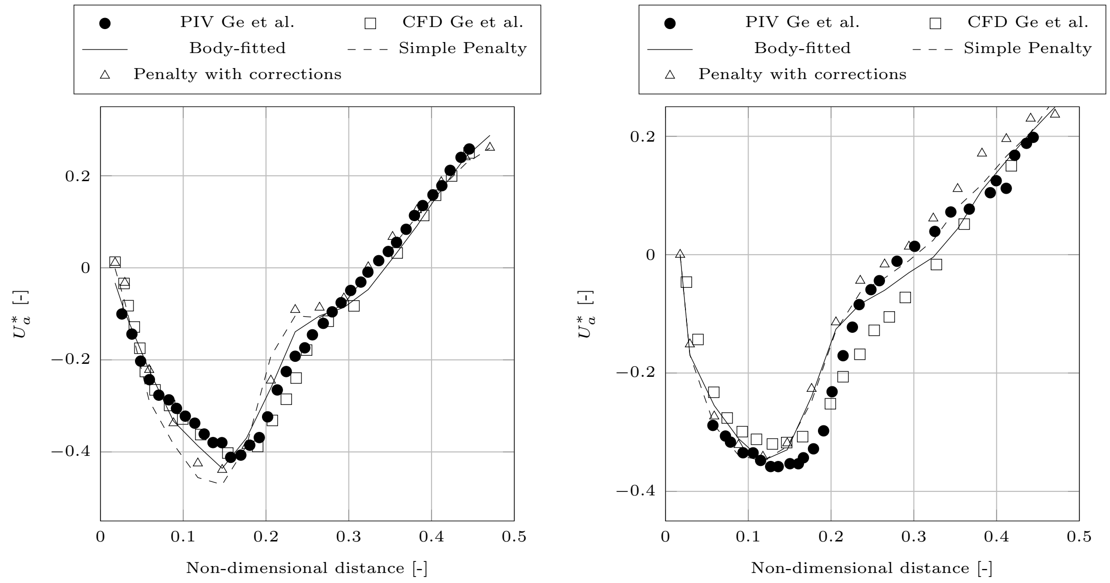
(b) IBM

Figure 5.29: Q field iso-contours with $Q = 10^4$ coloured by the magnitude of the velocity (m/s).

5.3.4 Averaged data

All data averaging were performed over 20 revolutions, after an initial start-up period of 20 revolutions. LES computations reported in [29], and based on similar Reynolds number and averaging procedure, confirmed that the initial start-up phase was sufficient to filter out transients and to provide statistically averaged data. For comparison, 15 impeller revolutions were reported in [52] and 15-20 revolutions in [142]. The results discussed in this section concern phase resolved velocity measured along the sampling lines defined earlier and averaged over these 20 revolutions. The discussion also considers the turbulent kinetic energy which is both phase and time averaged. The phase averaging is performed over 5° angular steps covering one quadrant of the flow domain (i.e. covering the flow domain between two baffles). The axial and radial velocities and the turbulent kinetic energy are normalized in terms of the impeller tip velocity $U_{tip} = \pi N D$ as $U^*_{a,r} = U_{a,r}/U_{tip}$ and $k^* = k/U^2_{tip}$. The total turbulent kinetic energy reported here is determined from the resolved flow velocity components u_i and the sub-grid scales contribution $k_{sgs} = \nu_t/(0.07/\tilde{d})$ with $k = (\bar{u_i^2} - \bar{u_i}^2)/2 + k_{sgs}$ [52].

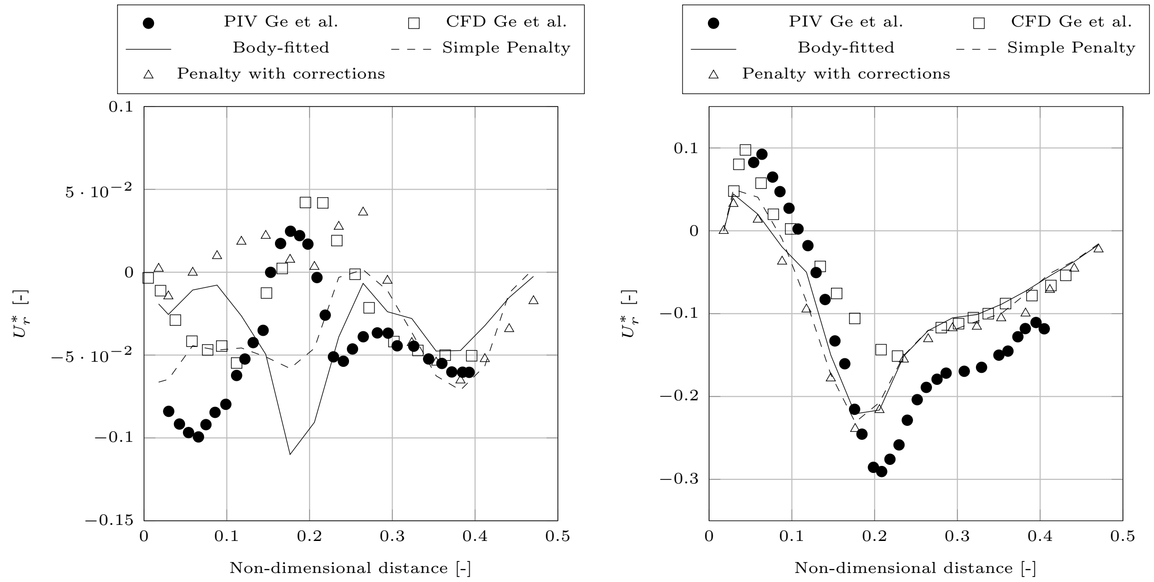
The phase resolved axial and radial velocities and the phase averaged turbulent kinetic energy distributions along the two sample lines (dashed red lines on Figure 5.26) are compared against experimental and simulation profiles from [41] in Figures 5.30, 5.31 and 5.32. Although the general trends and peak velocity values are similar to the experimental data, significant differences are observed over parts of the profiles. The difference is particularly noticeable in the case of the radial velocity downstream of the impeller (Figure 5.31(a)). However, the experimental velocities in this case experience relatively small variations around the average value of $0.045 m/s$ which is approximately 5% of the axial velocity range. Both body-fitted and IBM models predict generally similar results by comparison with the reference profiles obtained with the $k - \epsilon$ RANS model, with two notable differences, (i) gradients are generally better resolved with the DES models and (ii) the peak in turbulent kinetic energy downstream of the impeller, although still significantly dampened is closer to experimental observations. The $k - \epsilon$ model does seem to achieve more accurate kinetic energy predictions upstream of the impeller (Figure 5.32(b)). This however is in conjunction with significant under prediction of the radial velocity which, in the case of the DDES simulation, impacts on the determination of the kinetic energy $k = (\bar{u_i^2} - \bar{u_i}^2)/2 + k_{sgs}$. The IBM is shown to be in very good agreement with the body-fitted simulations for most profiles. Regarding the turbulent kinetic energy, one can note a better estimation of the energy peak upstream of the impeller with the IBM. This overestimation compared to the body-fitted case may be due to the rasterization effect artificially increasing stresses. Finally, it is worth mentioning the improvements achieved with the corrected Penalty method, obtained for the axial velocity downstream of the impeller (Figure 5.30(a)), which reduces the velocity peak in accordance to the other data.



(a) Downstream of the impeller

(b) Upstream of the impeller

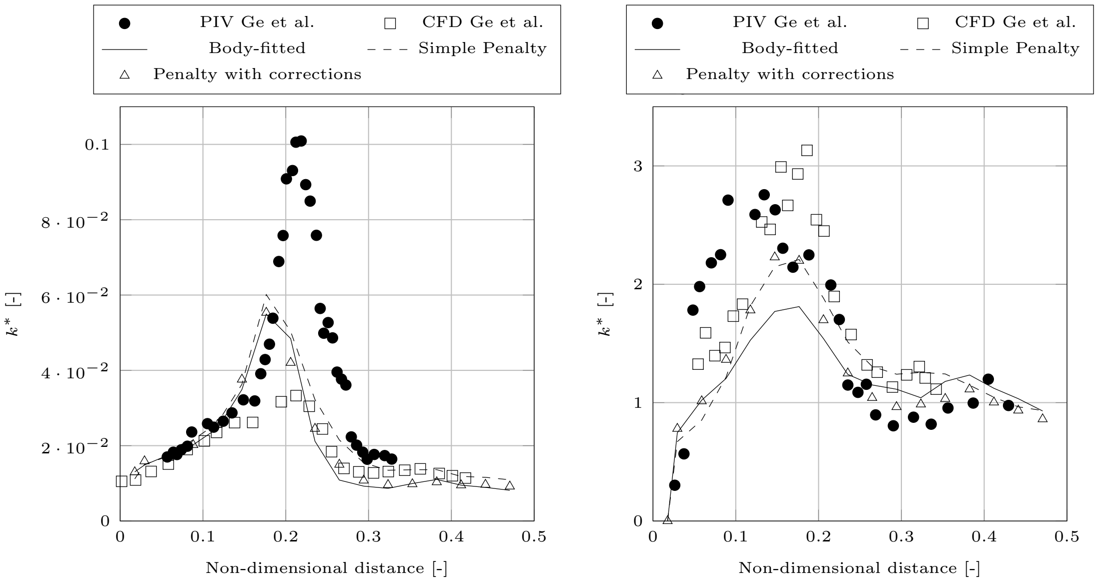
Figure 5.30: Phase and time averaged axial velocity along the two sampling lines.



(a) Downstream of the impeller

(b) Upstream of the impeller

Figure 5.31: Phase and time averaged radial velocity along the two sampling lines.



(a) Downstream of the impeller

(b) Upstream of the impeller

Figure 5.32: Phase and time averaged turbulent kinetic energy along the two sampling lines.

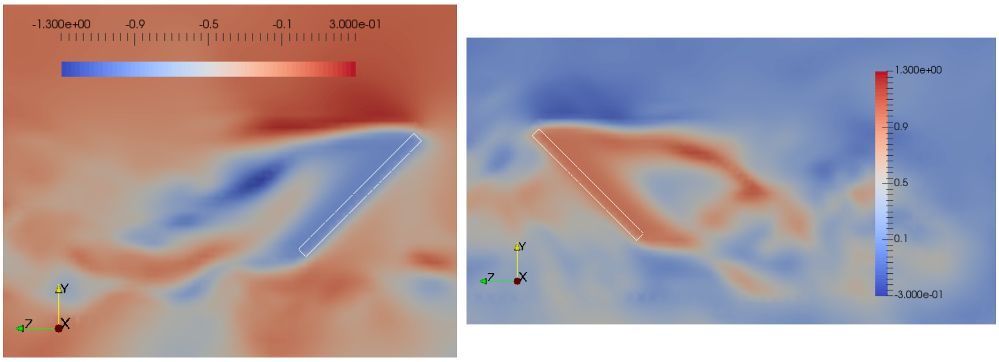


Figure 5.33: Contours of the velocity (m/s), projected in the direction of the impeller rotational speed (+z), obtained with a linear reconstruction at the immersed surface. The contours are shown on cross-sections perpendicular to two impeller blades diametrically opposed. The white contours depict the impeller surface.

5.3.5 Sensitivity to the velocity reconstruction scheme

The influence of the type of velocity correction at the immersed interface is discussed in this section. Figure 5.33 shows the velocity contour in the wake of two blades on

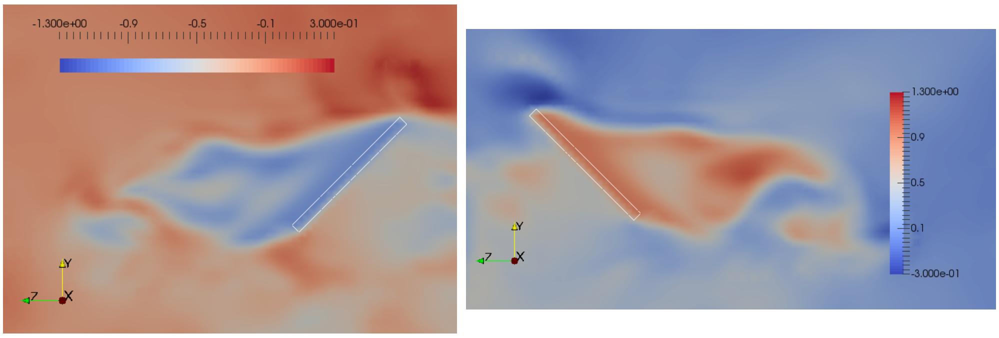
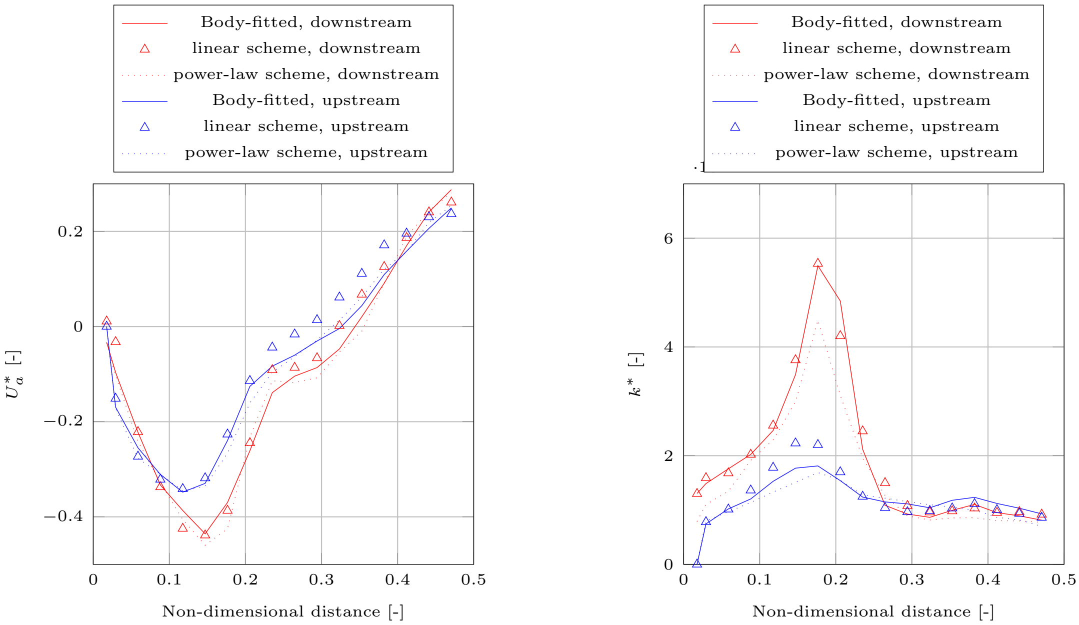


Figure 5.34: Contours of the velocity (m/s), projected in the direction of the impeller rotational speed (+z), obtained with a power-law ($k = 1/7$) reconstruction at the immersed surface. The contours are shown on cross-sections perpendicular to two impeller blades diametrically opposed at a distance of $0.06m$ from the impeller center of rotation. The white contours depict the blades surface.

opposite sides of their axis of rotation, when a linear correction is applied. The velocity is projected in the tangential direction of the impeller rotational speed. Figure 5.34 shows a similarly projected velocity at the same location when a correction based on a power-law $k = 1/7$ is applied. Theses results are shown after 5 impeller revolutions. Different profiles of separation can be observed between the two types of velocity reconstruction. With the linear correction, the wake behind the blade appears broader, while on the contrary the power-law correction generates a narrower wake with a smaller angle of separation from the blades corner. In addition, along the blades, the separation appears closer to the impeller center of rotation with the linear correction. The non-transient nature of these flow characteristics were verified. The same conclusions can be drawn at different locations and different times, although the differences between the two methods slightly reduce with time probably because of the increasing mixing. These results are consistent with the power-law distribution, which enforces more flow tangency to the immersed surface. Similar observations on the delayed separation of the boundary layer in external flow past bluff bodies and on the wake size were made by Choi et al. [21].

Lower levels of turbulent energy are obtained with the power-law approach, as depicted by Figure 5.35(b). This was also observed in [21]. The higher turbulent



(a) Axial velocity

(b) Turbulent kinetic energy

Figure 5.35: Dependence of the averaged quantities in the velocity reconstruction scheme.

intensity generated with a linear reconstruction is due to the earlier flow detachment, which destabilizes the different boundary layers and promotes mixing. For the phase and time averaged mean velocity, differences are also visible for the axial velocity in Figure 5.35(a) but appear less significant. The same conclusion can be drawn for the radial velocity. Both instantaneous and time averaged flow characteristics have been found to be sensitive to the velocity reconstruction scheme. It can be argued that the linear correction gives predictions which are closer to the body-fitted simulation, especially in terms of turbulent quantities and structures. It is argued that this is due to the effect that the power law reconstruction has on promoting attachment of the boundary layer.

5.4 Conclusions

In this chapter, the sharp interface Penalty approach was applied to FSI problems involving turbulent mixing flows. The IBM was assessed in terms of its capabilities to generate stable and accurate solutions of the different flow problems. Particular attention was paid to the improvements brought by the correction schemes used for the boundary conditions at the immersed interface. Inherent shortcomings were also identified in specific configurations. Numerical results obtained with the Penalty methods were compared to equivalent body-fitted simulations, previously published data and experimental results when available. Finally, the coupling of the present IBM with the hybrid DES turbulence model was validated, including the penalization formulation of the Spalart-Allmaras equation.

Regarding centrifugal pump applications, the velocity correction scheme seems to reduce the effect of the rasterization inherent to IBM on Cartesian grids, by reducing both the mixing in the volute and the wake behind the impeller's trailing edge, giving results that are closer to predictions from the body-fitted simulations. The mirrored velocity correction (internal scheme) generates spurious vortices at the immersed surface if the pressure is not corrected in addition. Although fixing this issue, the imposition of the Neumann condition for the pressure leads however to large fluctuations of the pressure level when wall boundaries are close to the immersed surfaces, as was already highlighted in the previous chapter. Those fluctuations are negligible in a case like the Mixer, but become significant for the single blade pump, where a high confinement exists between the impeller and the volute casing. From a general point of view, the improvements of the different corrections schemes are less visible in this case. In addition, numerical estimations of the torque with both body-fitted models and IBM appear to be relatively poor. On the contrary better torque estimations, and more significant improvements due the correction schemes are obtained for the two-blade pump. This difference could be explained by the impeller design, geometry and symmetry of which in the latter pump leads to a narrower range of turbulent structures, making the averaged quantities easier to predict.

The resolution of the boundary layer by standard means is delicate and questionable for the applications considered in this chapter, as the inherent turbulent rotating flows generate high mixing and separations and no clear attached boundary layers. Hence, a low influence of the inflation layers for body-conforming simulations was identified for the estimation of integrated quantities in the first pump case. For IBM, the sensitivity to the velocity reconstruction scheme was assessed for the mixer. A power-law model describing the logarithmic profile of an attached boundary layer was not found to give better predictions of averaged or instantaneous flow variables, in comparison to the linear correction. Regarding the turbulence model, the coupling between our IBM and the hybrid DES model was found to correctly predict turbulent structures and quantities in comparison to body-conforming models. The IBM formulation of the turbulent viscosity equation (modified Spalart-Allmaras equation) was found to improve the results in terms of head and torque, and to be necessary for full RANS simulations.

Chapter 6

Results and discussion: Comparison of Immersed Boundary Solutions for handling rigid 3D geometries

6.1 Introduction

This chapters has a dual aim:

1. the assessment of a diffuse IBM coupled to a Lattice-Boltzmann solver
2. the comparison of this diffuse LB-IBM to the sharp NS-IBM developed during this research and validated in the previous chapters

The formulation of the LB-IBM is introduced first, and its accuracy is then analysed by comparison with other wall treatment methods in the Lattice-Boltzmann context. Finally, the method is compared against the NS-IBM (xIBM) predictions with several of the test cases considered in Chapters 4 and 5.

6.2 A diffuse iterative IBM within Lattice-Boltzmann

This section presents the IBM available in the Lattice-Boltzmann open-source software Palabos. The IBM is based on the Multi Direct Forcing Method (MDFM) developed by Wang et al. [147] initially in the Navier-Stokes context.

6.2.1 The Multi Direct Forcing Method (MDFM)

The MDFM has already been coupled to LBM by Ota, Suzuki & Inamuro ([97], [125]) to study the interaction of fluids with rigid bodies. In this model, the presence of an immersed object in the fluid is accounted through a two-step approach:

1. A particle distribution function field $f'_\alpha(\mathbf{x}, t + \Delta t)$ is evaluated according to the standard discretized LB equation (see Equation 2.6 in Chapter 2), without taking into account the presence of the solid body.

2. This intermediate field $f'_\alpha(\mathbf{x}, t + \Delta t)$ is corrected by the body force accounting for the solid:

$$f_\alpha(\mathbf{x}, t + \Delta t) = f'_\alpha(\mathbf{x}, t + \Delta t) + 3\Delta x \omega_\alpha \mathbf{c}_\alpha \mathbf{g}(\mathbf{x}, t + \Delta t) \tag{6.1}$$

   where $\mathbf{g}$ is the body force representing the action of the solid on the fluid. The body force is determined to enforce the no-slip boundary condition at the fluid-solid interface.

6.2.2 Iterative estimation of the Immersed Boundary force

The second point of interest is the definition of the body force $\mathbf{g}$ injected in Equation 6.1. The iterative nature of the MDFM helps achieve the no-slip condition in comparison to standard direct forcing approaches [147]. In the following equations, l stands for the iteration and it was shown that $l = 5$ is sufficient to keep a no-slip condition on the boundary points [125]. The IBM force $\mathbf{g}$ is computed as following:

$$\mathbf{g}_{l+1}(\mathbf{X}_k, t + \Delta t) = \mathbf{g}_l(\mathbf{X}_k, t + \Delta t) + \frac{\mathbf{U}_k(t + \Delta t) - \mathbf{u}_l(\mathbf{X}_k, t + \Delta t)}{\Delta t} \tag{6.2}$$

where $\mathbf{g}_l$ and $\mathbf{u}_l$ are respectively the lth iteration of the body force and the fluid velocity estimations. The latter is corrected from the intermediate velocity field $\mathbf{u}'$, itself obtained from the intermediate particle distribution function field $f'_\alpha(\mathbf{x}, t + \Delta t)$, introduced earlier. This correction procedure reads:

$$\mathbf{u}_l(\mathbf{x}, t + \Delta t) = \mathbf{u}'(\mathbf{x}, t + \Delta t) + \mathbf{g}_l(\mathbf{x}, t + \Delta t)\Delta t \tag{6.3}$$

To translate the physical variables from the Eulerian grid node $\mathbf{x}$ to the Lagrangian points $\mathbf{X}_k$, and vice-versa, a standard interpolation procedure is performed with the weight functions given by the Dirac delta function δ used in Lai & Peskin [78]:

$$\mathbf{u}_l(\mathbf{X}_k, t + \Delta t) = \int \mathbf{u}_l(\mathbf{x}, t + \Delta t)\delta(\mathbf{x} - \mathbf{X}_k)d\mathbf{x} \tag{6.4}$$

This procedure spreads the forcing to the surrounding Eulerian nodes, making the IBM diffusive. As the forcing is only applied in the vicinity of the interface and not inside the totality of the solid domain, non-physical internal flows may be generated (see next section).

The iterative method presented here starts with the initial estimation of the body force at the Lagrangian points, as:

$$\mathbf{g}_0(\mathbf{X}_k, t + \Delta t) = \frac{\mathbf{U}_k(t + \Delta t) - \mathbf{u}'(\mathbf{X}_k, t + \Delta t)}{\Delta t} \tag{6.5}$$

6.2.3 A corrected estimation of the force acting on the body

For further applications, the instantaneous force acting on the immersed solid can be computed from the force accounting for the solid presence in the fluid, which is merely the IBM force. The discrete formulation of this total force $\mathbf{F}_{tot}$ reads then:

$$\mathbf{F}_{tot}(t) = -\sum_{\mathbf{x}} \mathbf{g}(\mathbf{x}, t)(\Delta x)^3 \tag{6.6}$$

In Equation 6.6, Δx refers to the lattice spacing and $\mathbf{g}(\mathbf{x}, t)$ is the body force computed on the Eulerian grid similarly to Equation 6.3. In reality, the sum of the body force $\mathbf{g}$ over the immersed object involves both contributions from the external and the internal fluids. For moving solids, a part of this summation is used to move the internal fluid. The latter has to be removed from the total force in order to obtain the physical force from the external fluid only, acting on the immersed solid. This issued has been addressed by Suzuki & Inamuro among others in [125]. The physical force $\mathbf{F}(t)$ can then be obtained as $\mathbf{F}_{tot}(t) = \mathbf{F}(t) + \mathbf{F}_{in}(t)$, where $\mathbf{F}_{in}(t)$ represents the contribution from the internal fluid. Also called the internal mass effect, the latter is defined as:

$$\mathbf{F}_{in}(t) = -\frac{d}{dt} \int_{\Omega_s(t)} \mathbf{u}(\mathbf{x}, \mathbf{t}) d\mathbf{x} \tag{6.7}$$

Suzuki & Inamuro have considered two ways to discretize Equation 6.7. The first method, which is adopted in this research, assumes a rigid body motion inside the solid domain Ω_s. In this case the internal mass effect is simplified as in Equation 6.8, with $\mathbf{U_c}(t)$ being the instantaneous velocity of the solid center of mass.

$$\mathbf{F}_{in}(t) \approx -\Omega_s(t) \frac{\mathbf{U_c}(t) - \mathbf{U_c}(t - \Delta t)}{dt} \tag{6.8}$$

In the second method, artificial internal Lagrangian points are used to compute an internal velocity field interpolated from the fluid velocity. A similar model can also be defined for the estimation of the physical torque acting on the immersed object. Suzuki & Inamuro have shown that taking into account the internal mass effect is necessary for this kind of IBM and for Reynolds numbers above 10, if one is interested in the physical forces acting on the object. The two formulations of $\mathbf{F}_{in}(t)$ lead to reasonably similar results. The second approach gives a better estimation of the torque when the angular motion of the body is distinct [125]. As it was not found to be the case in the rotating flow considered here, the choice of the first approach is relevant.

6.3 Accuracy of wall treatment models in LBM: application to the fixed cylinder in a cross flow

This comparative study is based on the fixed cylinder case in a cross flow described in Section 4.2. The accuracy of three different models for handling the presence of a solid in internal flows within LBM are assessed and compared: (i) IBM, (ii) Bounce-Back approach and (iii) Guo's approach [49]. All models were introduced previously in Section 2.1.2. In the Navier-Stokes context, the results of the simple first-order Penalty and the improved Penalty, based on velocity and pressure corrections, are used for comparison. The same grid resolutions are used for both types of solver and for all wall treatment models.

The standard model of the Lattice-Boltzmann equation as described in Section 2.1.2 achieves a second order accuracy. As shown in Table 6.1, an IBM or a Bounce-Back representation of the cylinder obstacle reduces the local order of the LBM. The IBM leads to an order around 1 for both the drag coefficient and the re-circulation length, as it is expected for the MDFM. On the contrary, the reduction is less important with Bounce-Back modelling. This is due to the fact that the latter can achieve in theory a second order accuracy, when the cylinder wall is exactly midway between the lattice nodes. One can note finally that Guo's model [49] for a curved wall, which is in principle second order accurate, achieves for this test case an even higher order of accuracy. Regarding the absolute values, it appears that the diffusive nature of the LB-IBM generates a significant over-estimation of the two physical parameters, compared to higher methods and experimental results (see Table 4.4 in Section 2.1.2). C_D and L_S from the simple and improved NS-IBM are also listed in Table 6.1 as indicators. Although the simple NS-IBM is also a first order approach, the sharp definition of the immersed object does not lead to an over-estimation of the parameters in this case.

Model	C_D **(Grid 3)**	C_D **(Order)**	L_S **(Grid 3)**	L_S **(Order)**
LB-IBM	1.67	1.1	2.45	1.0
Bounce-Back	1.57	1.5	2.32	1.6
Guo [49]	1.56	2.9	2.23	5.0
NS-IBM simple Penalty	1.58	/	2.21	1.0
NS-IBM improved Penalty	1.60	/	2.30	1.5

Table 6.1: Estimations and associated order of accuracy of the drag coefficient and the re-circulation length, for different wall treatment models in the Navier-Stokes and Lattice-Boltzmann solvers.

6.4 Comparison of IBMs: Two-dimensional flow across a fixed array of cylinders

This case considers a two-dimensional flow across an array of cylinder and is described in Vincent et al. [143]. The goal is to compare the accuracy of the sharp and diffuse types of IBM for determining the pressure discharge in applications like fluidized beds. The theoretical value of the pressure discharge can be obtained by balancing the forces acting on the cylinders (drag and pressure). For a two-dimensional case, it reads:

$$\nabla p = -\alpha(1-\alpha)^{-5}\rho_l \frac{18\mu_l}{d_p^2}[1+015Re_p^{0.687}]U_0 \tag{6.9}$$

where α is the solid fraction obtained from the cylinder's blockage, d_p is the cylinder diameter and U_0 stands for the inlet velocity in the channel. Re_p is the resulting particulate Reynolds number. The fluid considered for this test case is thiocyanate KSCN, characteristics of which are listed in Table 6.2, along with the other physical parameters. In such configuration, the theoretical pressure discharge is about $306.5Pa$. The channel dimensions are $0.6m$ in length and $0.1m$ in height. Three grid resolutions are considered with 990×165 cells, 1980×330 cells and 3960×660 cells, which corresponds respectively to approximately 10, 20 and 40 cells in the diameter of the cylinder.

Fluid Properties	Density Dynamic viscosity	$\rho_f = 1400kg/m^3$ $\mu_f = 3.8 \cdot 10^{-3}Pa.s$
Geometrical Properties	Cylinder diameter solid fraction	$d_p = 6mm$ $\alpha = 0.1$
Flow Properties	Inlet velocity	$U_0 = 0.12m/s$

Table 6.2: Fluid, flow and geometrical characteristics for flow across an array of cylinder.

For the sharp NS-IBM, the simple Penalty approach is considered. This method is compared to two wall treatments within the Lattice-Boltzmann framework: (i) the standard Bounce-Back modelling and (ii) the diffuse LB-IBM. The pressure and velocity fields are shown in Figures 6.1 and 6.2 for the NS-IBM. As expected, one can observe a global linear decrease of the pressure in the channel. Attached recirculating bubbles are visible behind the cylinders. After the first few rows where they are completely stable, the recirculating zones exhibit fluctuations in the transverse direction. This is due to the non-symmetrical layout of the array of cylinders.

The pressure discharge obtained with the different wall treatments are listed in Table 6.3 for the three grid resolutions. From these results, the diffuse IBM appears not surprisingly as the least accurate method and leads to an overestimation of the discharge, due to an overestimation of the recirculation bubble size. The two other wall treatments involving a sharp interface give reasonably good results. One can note in particular the good agreement with the theoretical discharge obtained for the bounce-back method starting from the coarse grid. Also, the sharp NS-IBM exhibits a good convergence behaviour.

Model	**Bounce-Back (LBM)**	**diffuse LB-IBM**	**sharp NS-IBM**
coarse grid	328.8	398.0	262.2
medium grid	334.4	362.8	322.8
fine grid	312.8	339.2	337.0

Table 6.3: Estimations of the pressure discharge (in Pa) for different wall treatments and grid resolutions.

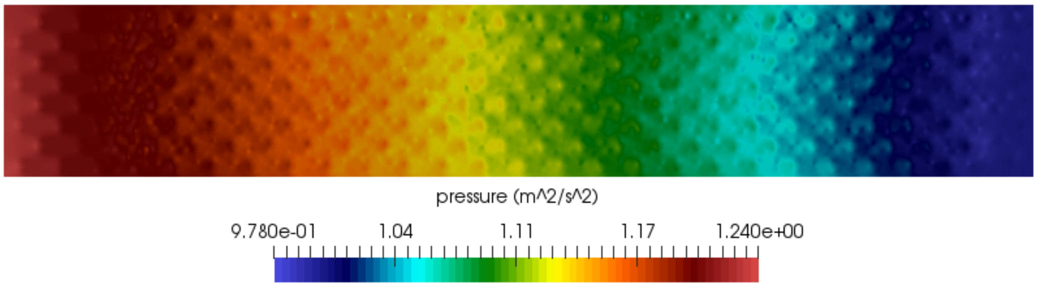
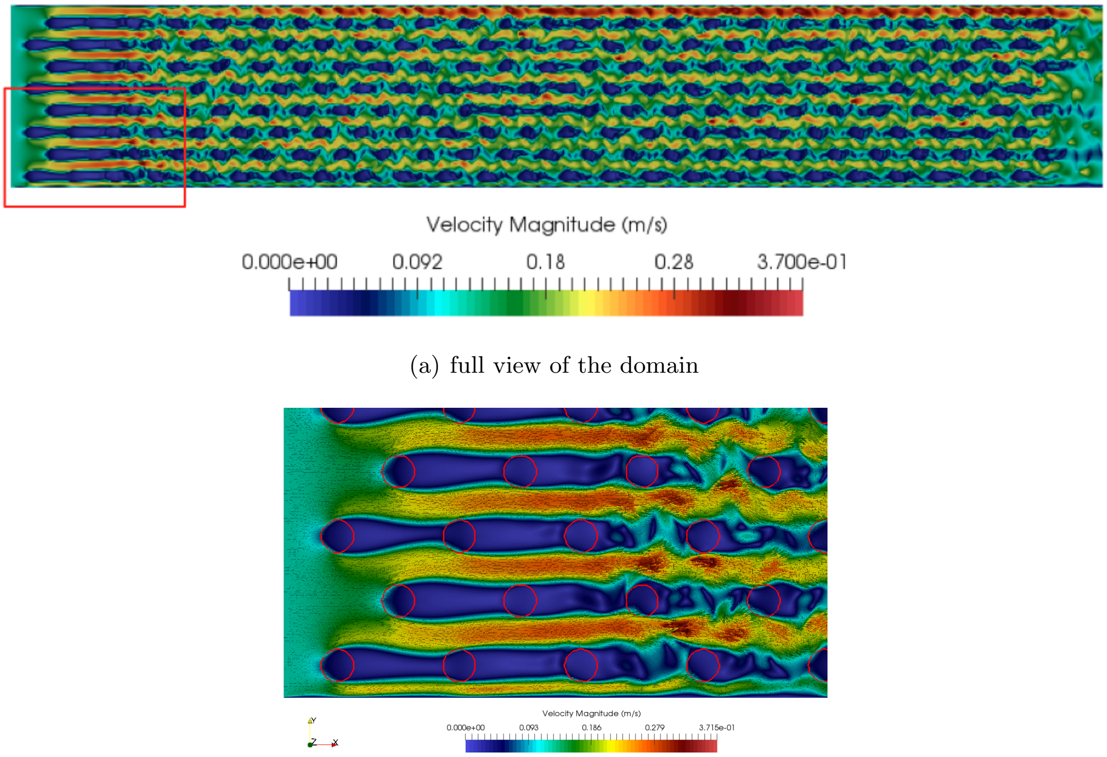


Figure 6.1: Pressure contour of a two-dimensional flow across a fixed array of cylinders with NS-IBM.



(a) full view of the domain

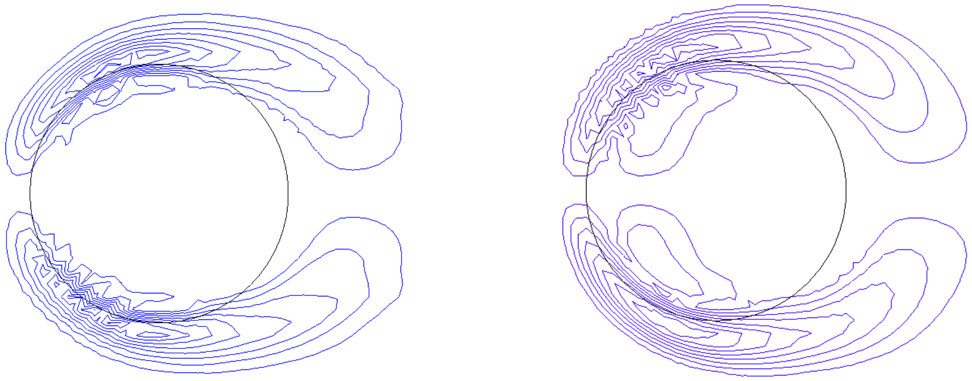
(b) close-up view near the inlet

Figure 6.2: Contour of velocity magnitude for a two-dimensional flow across a fixed array of cylinders with NS-IBM. In the close-up view, the velocity vectors as well as the iso-contour of the mask function (in red) depicting the immersed cylinder interfaces are shown in addition.

6.5 Comparison of IBMs: in-line oscillating cylinder in a fluid at rest

In this section, the in-line oscillating cylinder case in a fluid at rest is considered, as described in Section 4.3. The results of the diffuse LB-IBM are compared to the sharp NS-IBM.

Figure 6.3 shows the iso-contours of the vorticity magnitude in the vicinity of the moving cylinder. The contours obtains from both NS-IBM and LB-IBM compare well and are in good agreement with the numerical results of Liao et al. [85] at the same cylinder position.



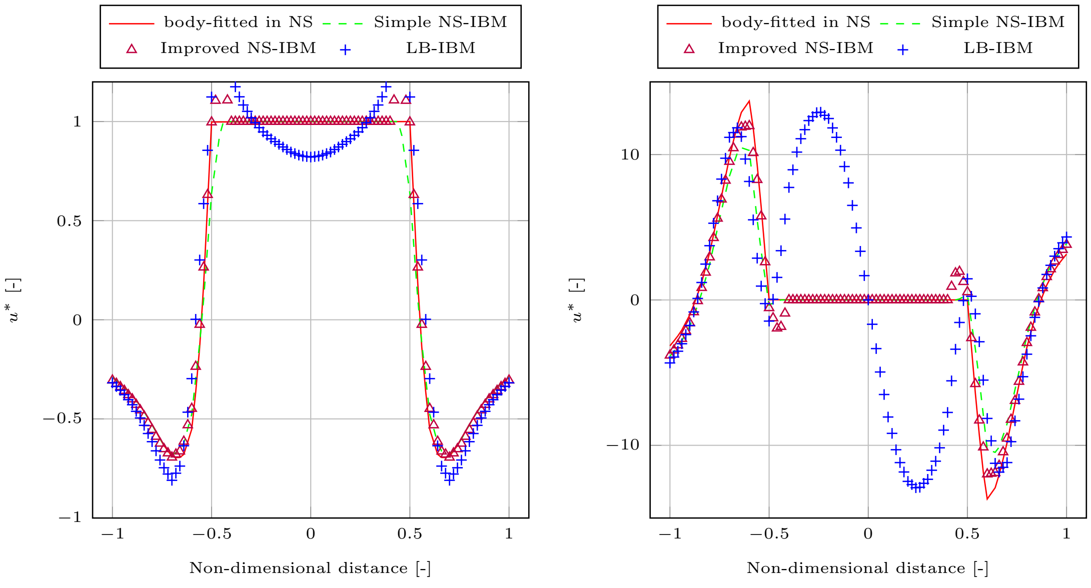
(a) NS-IBM (improved penalty)

(b) LB-IBM

Figure 6.3: Iso-contours of the vorticity magnitude around the cylinder for a phase angle $\omega t = 180°$.

The profiles of the streamwise and the transverse velocity component are shown in Figure 6.4 and 6.5 for $x = 0$ and $x = 0.6D$ respectively. The results from the body-fitted simulation performed with OpenFOAM$^{®}$ are used as a reference. The spreading of the LB-IBM forcing is largely visible. This generates a thickening of the real size of the cylinder more important than for NS-IBMs, which is shifting the maxima of velocity in the outward radial direction. For the profile in front of the cylinder in Figure 6.5, the thickening leads to a small increase of the maxima of velocity. The same conclusion can be drawn behind the cylinder at $x = -0.6D$ for instance. As the IBM forcing in the

Lattice-Botlzmann solver is only computed in the vicinity of the object's interface, the internal flow generated with moving boundaries is here visible between $-0.5D$ and $0.5D$ in Figure 6.4.

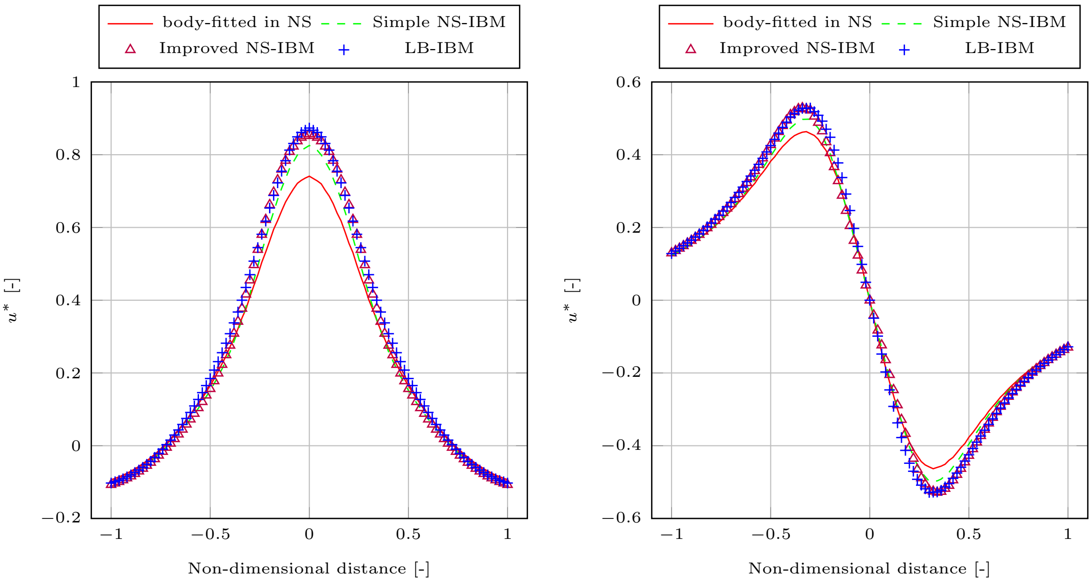


(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 6.4: Velocity profiles at $x = 0$ with a phase angle $\omega t = 180°$ for NS-IBMs and LB-IBM.

The evolution of the drag coefficient is given in Figure 6.6 for the LB-IBM and both simple first order NS-IBM and its corrected version xIBM (with velocity and pressure correction schemes). The experimental results of Dutsch et al. [32] are given as reference. Firstly, two ways of computing the drag force are compared for the Lattice-Boltzmann approach. It shows clearly that it is necessary to take into account the internal flow for a correct estimation of the fluid action on the object's surface. When the internal mass effect is taken into account, the drag force is however still slightly overestimated in comparison to the sharp IBMs implemented in the Navier-Stokes solver. A combination of two factors is identified to explain this overestimation: (i) the diffuse nature of LB-IBM which thickens the solid and (ii) the assumption of the internal rigid body motion for the estimation of the fluid force acting on the solid, which is not exactly satisfied in this case. Despite this overestimation, the advantage of the diffusive approach lies



(a) Streamwise velocity profile

(b) Transverse velocity profile

Figure 6.5: Velocity profiles at $x = 0.6D$ with a phase angle $\omega t = 180°$ for NS-IBMs and LB-IBM.

in the smooth evolution of the drag, while significant instabilities are generated with sharp approaches. These instabilities have found to be larger when correction schemes implemented for velocity and pressure at the interface are used in the improved NS-IBM.

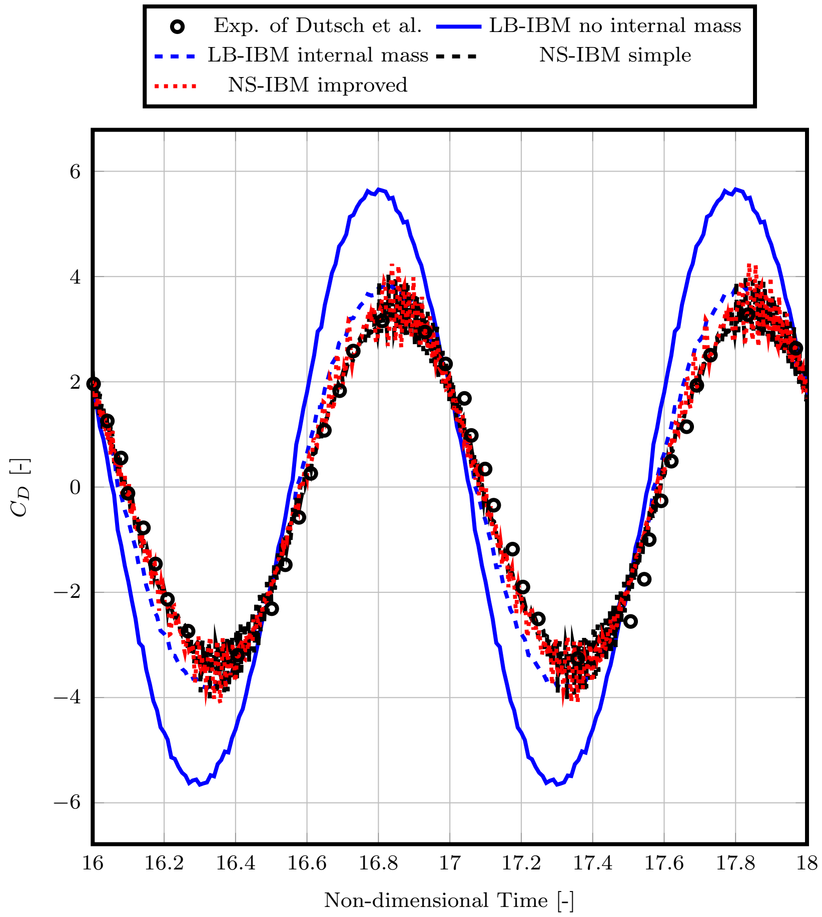


Figure 6.6: Comparison of the evolution of the instantaneous drag coefficients.

6.6 Comparison of IBMs: single-blade centrifugal pump

This case is based on the single-blade pump model described in Section 5.1 for the sharp NS-IBM. Regarding the Lattice-Boltzmann framework, the features of the computational domain are outlined in Figure 6.7. The pump volute and impeller are immersed in a fluid domain. The impeller wall is modelled with the diffuse IBM while the volute walls are modelled with the bounce-back technique. The fluid nodes above the inlet surface and exterior to the volute are considered as dead nodes, for which the fluid equations are not solved. Similar to the configuration described in Section 5.1, the flow enters the "simulation" domain vertically downward through the planar inlet surface (in which the inlet pipe has been extruded). The lattice grid spacing is constant and uniform with a size of $2mm$, and similar size is used to mesh the surface of the impeller with triangular elements, in accordance to the mesh resolution used for the sharp NS-IBM. Regarding the turbulence model, a Smagorinsky based LES model is used to describe the effect of subgrid scales in the flow.

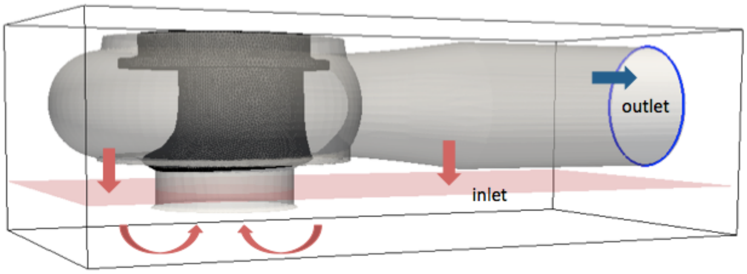
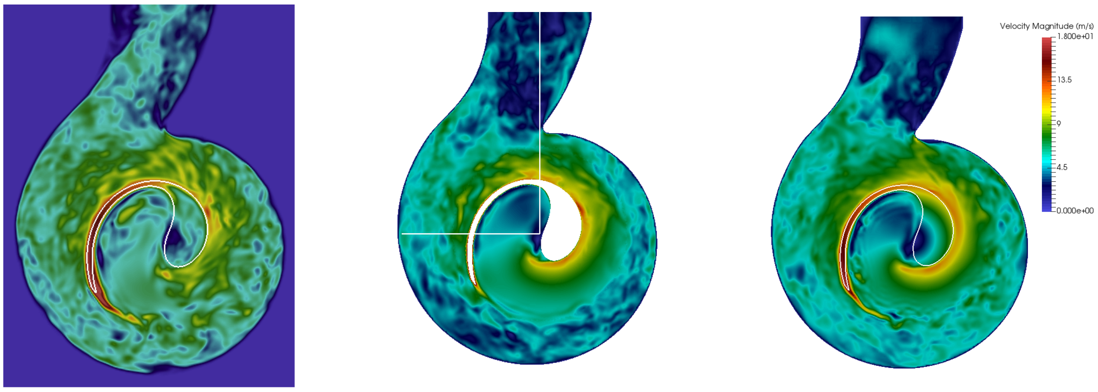


Figure 6.7: Computational domain of the Lattice-Boltzmann - Immersed Boundary Method (LB-IBM): the pump volute is represented in grey, the impeller modelled with IBM and its triangular mesh is represented in dark grey, the red plane denotes the inlet surface and the blue circle surrounds the outlet surface.

The contours of the magnitude of the velocity are shown in Figure 6.8 for the two types of IBM solution along with the body-fitted solution. The two IBMs generate an average velocity magnitude in the volute which is higher than the equivalent body-fitted level, especially on the discharge side of the impeller. The sharp IBM leads however to more accurate results in comparison to the diffuse LBM for two noticeable flow characteristics: (i) a better flow uniformity around the impeller leading edge and on the impeller suction side, and (ii) a thinner wake behind the trailing edge, both in accordance with the body-fitted simulation. These results highlight the shortcoming of the diffuse nature of the LB-IBM, due to the spreading of the IBM force. The failure of the latter model in predicting a correct mixing level in the volute is clearly visible in Figure 6.9, which shows for the three numerical methods the vorticity field for the same iso-contour. The LB-IBM based on a diffuse interface leads to a large overestimation of vortical structures. On the other hand, a good agreement is obtained between the sharp NS-IBM and the body-fitted simulation, although a high vorticity seems to be found especially near the trailing edge with the IBM.



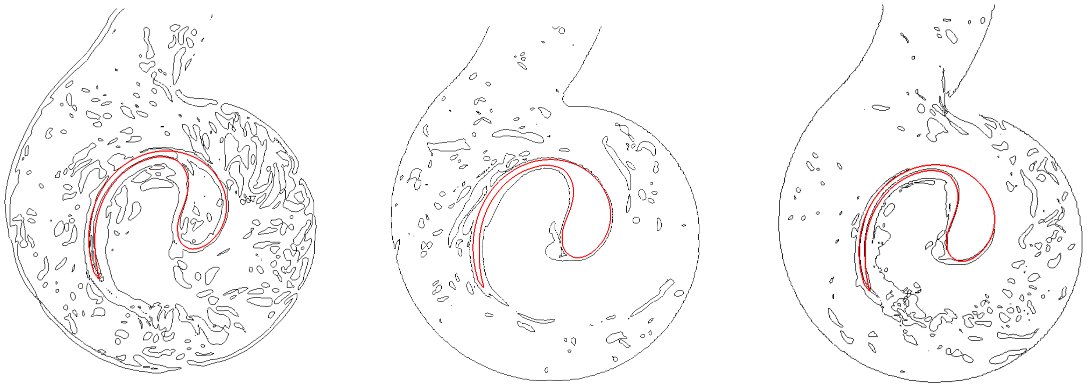
(a) LB-IBM

(b) Body-fitted (NS)

(c) NS-IBM

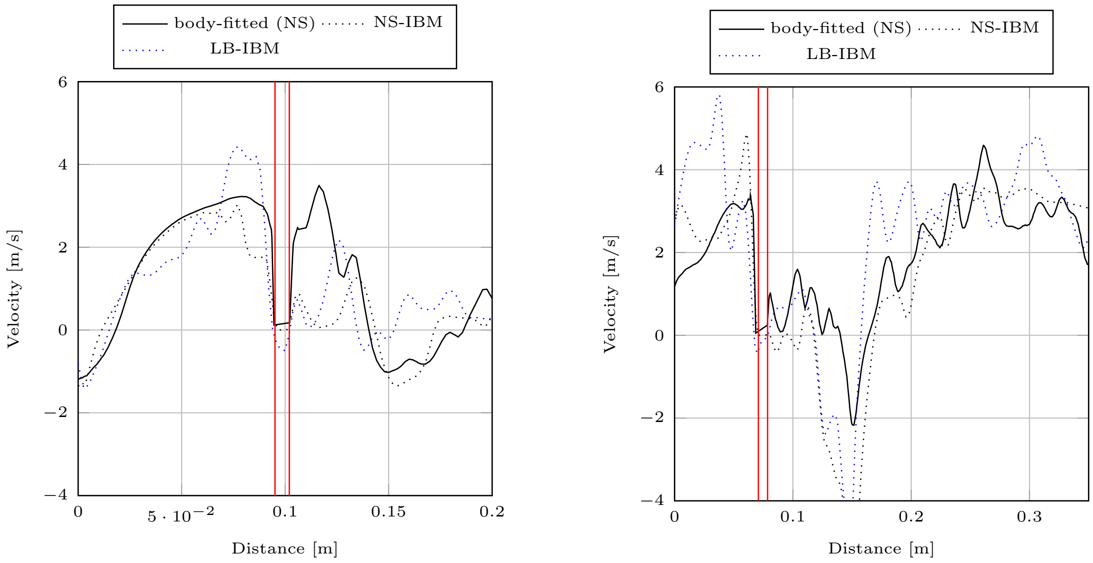
Figure 6.8: Contours of the velocity magnitude on the cross section of the pump passing through the center of the outlet pipe at $t = 0.2s$. The interface with the impeller is represented by a thick white line for IBM cases. The two straight white lines in the body-fitted solution represent the profiles where the instantaneous velocity values are extracted.

The higher mixing obtained with LB-IBM is also visible on the local velocity profiles in Figures 6.10 and 6.11. The difference in mixing is especially noticeable on the suction side of the impeller, illustrated by the high variability of the instantaneous velocity in comparison to the two other approaches. However, it is important to note the relatively good agreement found for the size of the vortices, which can be characterized by the velocity fluctuations.



(a) LB-IBM (b) Body-fitted (NS) (c) NS-IBM

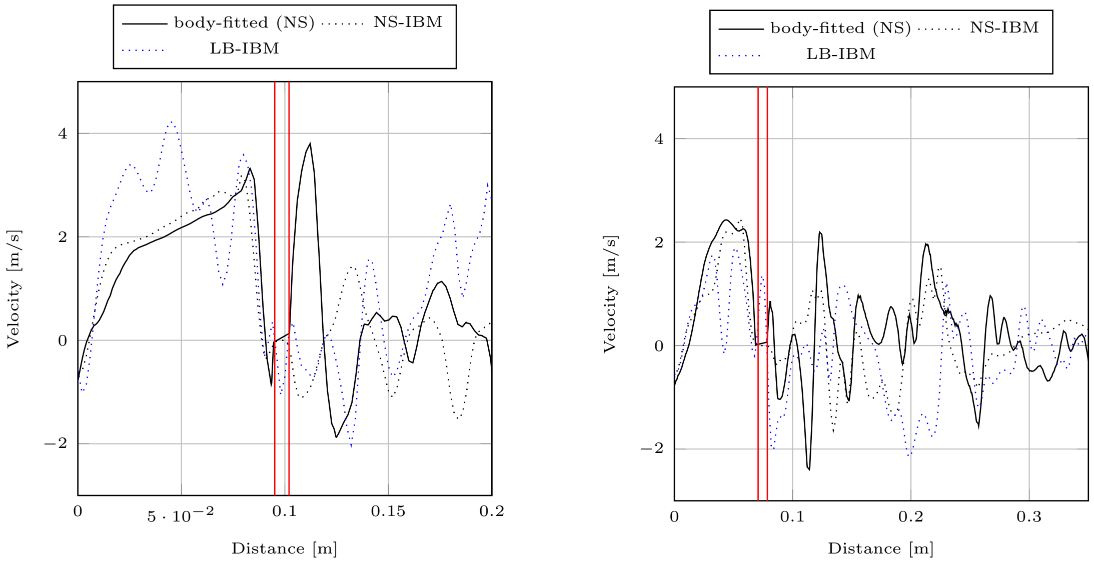
Figure 6.9: Iso-contour of the vorticity magnitude $\omega = 600s^{-1}$ on the cross section of the pump passing through the center of the outlet pipe at $t = 0.2s$. The interface with the impeller is represented by a thick red line.



(a) Profile 1 (b) Profile 2

Figure 6.10: Instantaneous profiles of the radial component of the velocity. The impeller position is represented by the vertical red lines.

Although significant discrepancies have been identified regarding the predictions of the instantaneous flow in the pump with the diffuse IBM, a good estimation of the static pressure converted from the average kinetic energy developed by the impeller, can be found in this case, similarly to the two other numerical approaches. This argument is



(a) Profile 1

(b) Profile 2

Figure 6.11: Instantaneous profiles of the vertical component of the velocity. The impeller position is represented by the vertical red lines.

supported by a performance study. The hydraulic heads obtained experimentally as well as the numerical heads predicted with both IBMs and the body-fitted model are summarised in Figure 6.12 for five different flow rates. The heads are non-dimensionalized by the experimental head obtained for the Best Efficiency Point (BEP). Both sharp and diffuse IBMs are giving head estimates which are in very good agreement with the body-fitted predictions and which compare also well with experimental data around the BEP. All numerical approaches tend however to underestimate the head, especially with the appearance of large separation and the formation of large training vortices at high flow rates, as this was already identified previously for the single-blade impeller in Section 5.1.3. Predictions in this case can indeed be expected to be strongly influenced by accurate modelling of boundary layers and would require much finer wall adjacent mesh. Adaptive meshing could provide a practical solution in this case without seriously compromising computational efficiency.

Experimental and numerical efficiencies are derived from the torques and shown in Figure 6.12 as a function of the flow rate, for all models except the NS-IBM. The latter model was shown to give very similar results compared to the body-fitted method, and both approaches fail at giving good estimates of the torque, as already seen in the previous chapter. Surprisingly, the efficiency obtained with the diffuse LB-IBM is in good agreement with experimental values up to $Q = 70l/s$. At the highest flow rate, the poor prediction is not due to a poor prediction of the torque, but to the poor head estimation in this case. Otherwise, the maximum relative difference between the LB-IBM solution and the experimental efficiency is 7%, obtained at $Q = 70l/s$. Two hypotheses to explain the good torque estimation with LB-IBM are that (i) the slight increase of the velocity magnitude in the volute altered the torque and compensated for other errors or that (ii) the smaller time step (one order of magnitude) required to ensure independence of the results with the Lattice-Boltzmann solver allowed to capture transients that are significant for the average torque. The instantaneous torque is shown in Figure 6.13 for the three numerical models. One can clearly see the periodic variations for the LB-IBM, which correspond to the impeller cycles. Such variations are not captured with the two Navier-Stokes models.

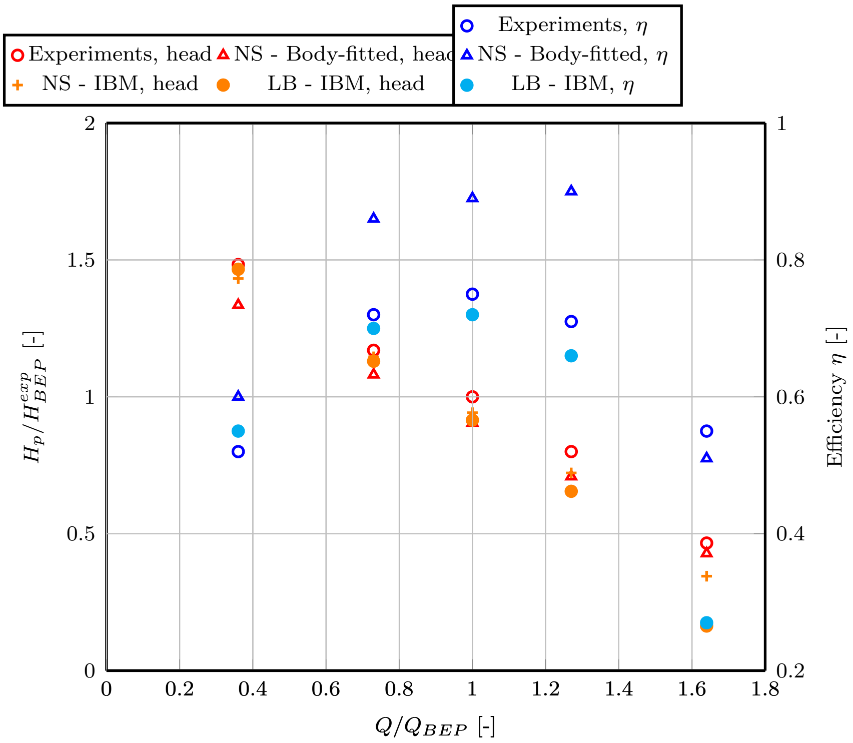


Figure 6.12: Performance curve for the single-blade pump obtained with different numerical approaches.

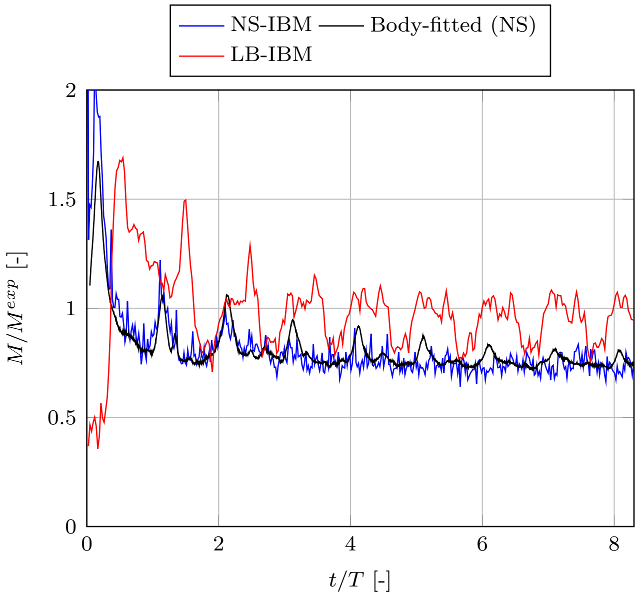


Figure 6.13: Instantaneous torque on the impeller of the pump at the BEP obtained with different numerical approaches.

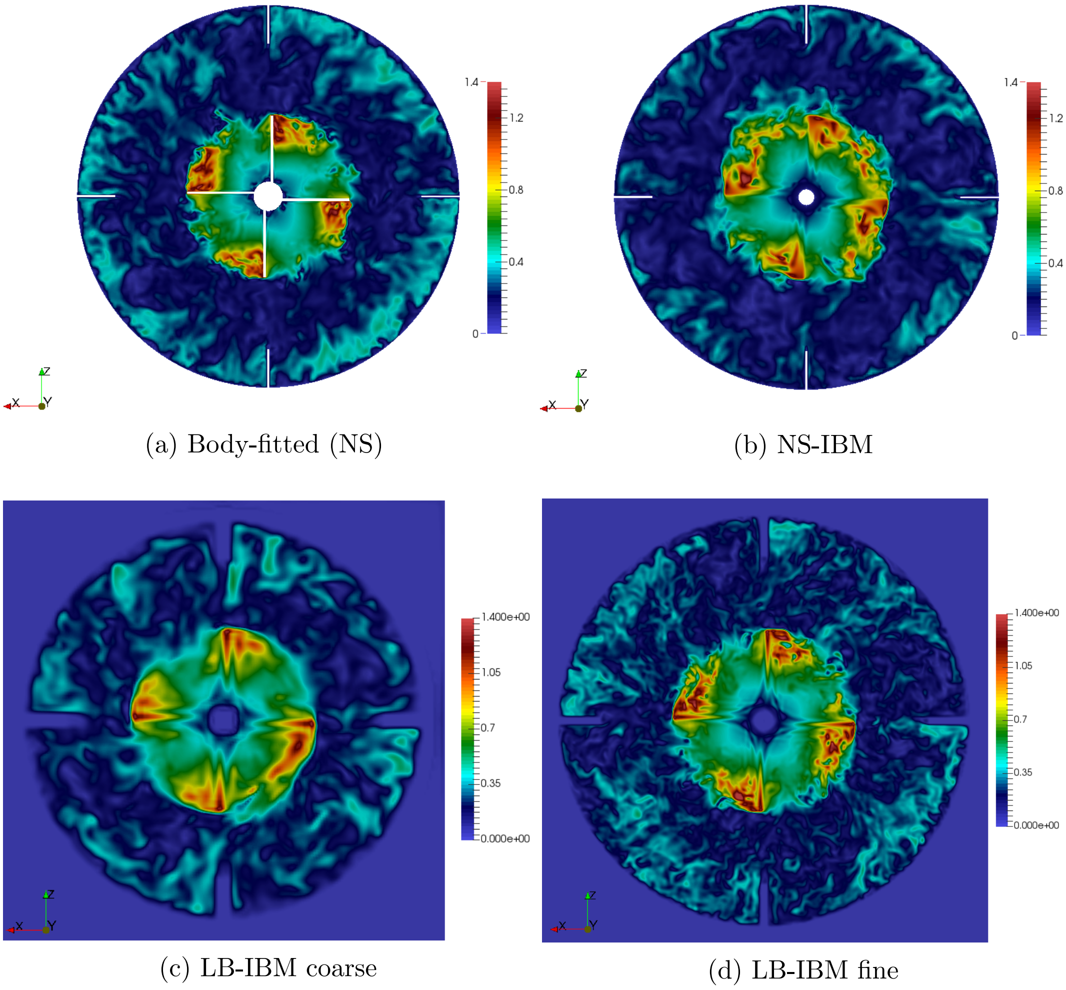
6.7 Comparison of IBMs: single-phase mixer

This last section aims at comparing the two types of IBM for the single-phase mixing in a stirred tank. The physics, the domain and the numerical parameters of this case were described in the previous chapter in Section 5.3. Two grids were built for the diffuse LB-IBM with uniform lattice everywhere in the domain: (i) a coarse grid with a Lattice spacing of $2mm$ and (ii) a fine grid with a Lattice spacing of $1mm$. The mesh resolution used for the sharp NS-IBM methods is thus in-between the two Lattice grids, as $1mm$ hexahedral cells were used in a region close to the impeller and $2mm$ hexahedral cells elsewhere. The body-fitted mesh includes an additional refined region, where the minimum cell size reaches $0.5mm$. A summary of the grids and the turbulence models used is provided in Table 6.4.

Model	**Turbulence model**	**max. grid res.**	**impeller thickness**
Body-fitted	DDES	$0.5mm$	$2mm$
sharp NS-IBM	coupled IBM-DES	$1mm$	$1.6mm$
LB-IBM coarse	LES	$2mm$	$1.6mm$
LB-IBM fine	LES	$1mm$	$1.6mm$

Table 6.4: Numerical settings for the different CFD models

Figure 6.14 shows the instantaneous contour of the velocity magnitude after 5 impeller revolutions for the three numerical methods. The regions of high velocity swept by the impeller blades are in good agreement with the body-fitted results, for both coarse and fine simulations. However, both types of IBM lead to larger wake trailing in comparison to the body-conforming case. Although the resolution in the central area is similar for the fine Lattice grid and the NS-IBM mesh, the NS-IBM exhibits also a more uniform flow than its Lattice counterpart in these high velocity regions. Finally the diffuse nature of the immersed interface with the LB-IBM is clearly visible on this figure from the the spread influence of the two parallel surfaces of each blade.



(a) Body-fitted (NS)

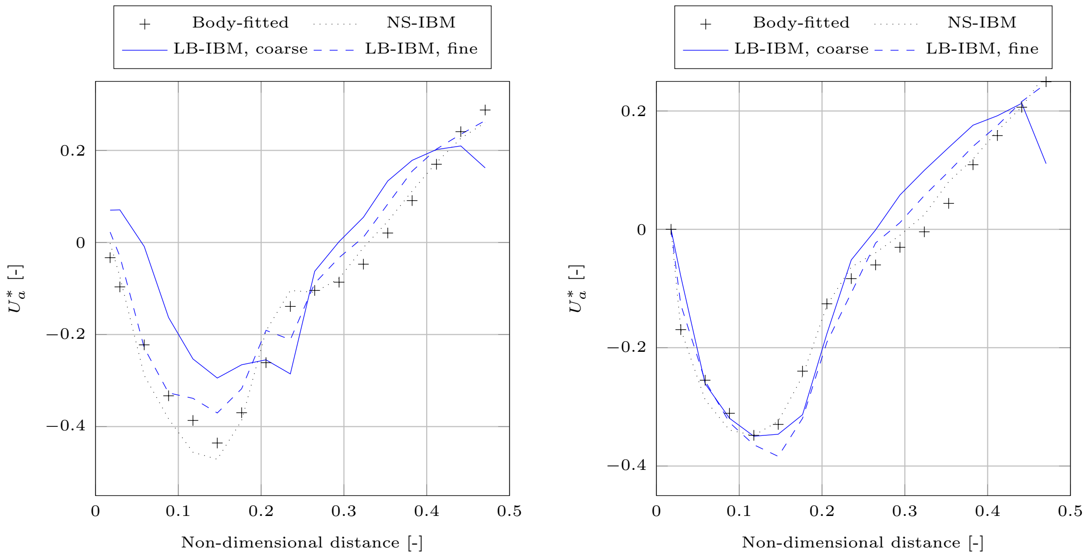
(b) NS-IBM

(c) LB-IBM coarse

(d) LB-IBM fine

Figure 6.14: Contour of the velocity magnitude (m/s) on a cross-section cutting the impeller at $y = 0.11m$.

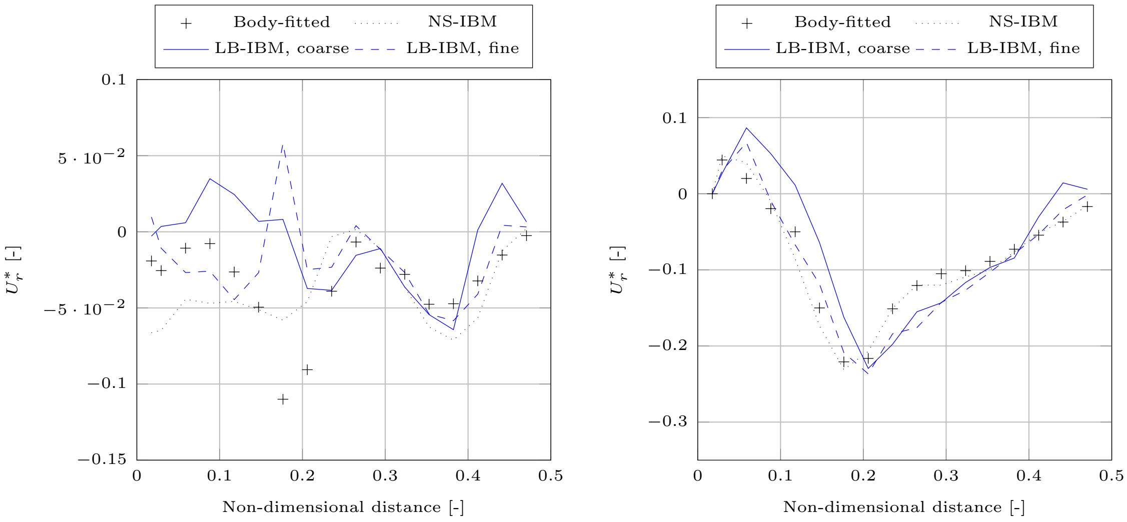
The distribution of the phase resolved averaged quantities are shown in Figures 6.15, 6.16 and 6.17 respectively for the axial velocity, the radial velocity and the turbulent kinetic energy. The same sampling lines as those described in Section 5.3 are used here, downstream and upstream of the impeller. From these averaged quantities, it appears that the coarse LB-IBM simulation is not able to predict the general trends and peak velocity values with a satisfying level of accuracy by comparison to the body-fitted case. Using a finer resolution does correct the prediction of both the velocity and the turbulent kinetic energy. This is especially noticeable for the impeller thrust force characterised by the axial velocity below the impeller in Figure 6.15(a) and the turbulent intensity level in Figure 6.17(a). Although a good agreement is obtained with the fine LB-IBM, the sharp penalty IBM in Navier-Stokes seems to give more accurate results. One can note again the higher mixing and turbulent intensity levels obtained with the diffuse LB-IBM in comparison to the sharp NS-IBM. For instance, the peak of turbulent kinetic energy downstream of the impeller is over-predicted by approximately 100% and 30% with the coarse and the fine LB-IBM respectively, in comparison to the body-fitted solution.



(a) Downstream of the impeller

(b) Upstream of the impeller

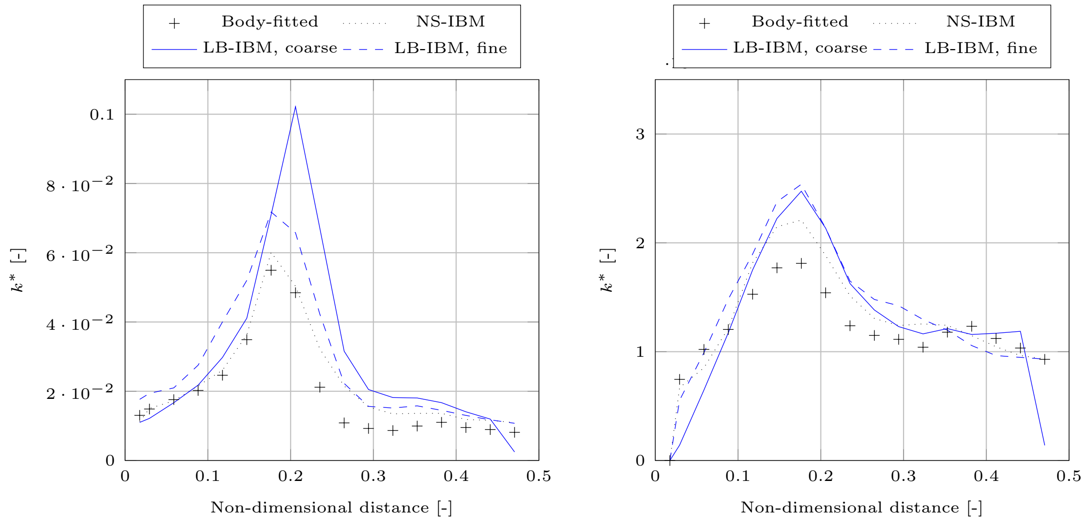
Figure 6.15: Phase and time averaged axial velocity along the two sampling lines.



(a) Downstream of the impeller

(b) Upstream of the impeller

Figure 6.16: Phase and time averaged radial velocity along the two sampling lines.



(a) Downstream of the impeller

(b) Upstream of the impeller

Figure 6.17: Phase and time averaged turbulent kinetic energy along the two sampling lines.

6.8 Conclusions

The coupled Lattice-Boltzmann - Immersed Boundary Method based on a diffuse definition of the immersed interface has been validated and compared to the sharp NS-IBM in this chapter through a range of laminar and turbulent test cases. The results have shown that the spread definition of the immersed interface in LB-IBM has a significant influence on the local and instantaneous velocities. The method leads also to an overestimation of the pressure stress on the immersed solid and an overestimation of the intensity of the mixing generated by its motion in turbulent cases. Despite this shortcoming, the LB-IBM has given good results in terms of time averaged quantities, as can be seen for the pump and the mixer cases, with a level of accuracy close to the sharp NS-IBM. In addition, the evolution of the distribution of pressure and viscous stress on the immersed surface show less noise in comparison to sharp interface methods. These conclusions give confidence in using the latter method in FSI applications involving the transport of secondary components.

Finally, Table 6.5 provides interesting insights into the computational costs of the different numerical methods used in this research for the specific mixer case. All simulations were performed on 48 processors, except for the body-fitted one which used 96 processors. The total CPU time in this case was thus doubled in order to make an approximate comparison. The time T_1 is defined as the CPU time per time iteration and per cell (or lattice node). From these results, it appears that the body-fitted and the simple Penalty NS-IBM have a similar cost in terms of computational time. The resources required for the improved NS-IBM, which includes the different corrections schemes, depend significantly on the resolution of the surface mesh. With a relatively fine surface mesh, the increase in the computational cost is important. In this case, one can see that the overall CPU time has been multiplied by a factor 4.7 from the simple to the improved NS-IBM. However, the higher time step achievable with NS-IBMs due to the absence of inflation layer allows to speed up the computation *in fine.* Although a finer resolution and a smaller time step are required in order to obtain stable and accurate solutions with LB-IBM, the fully explicit nature of the Lattice-Boltzmann solver makes it still the least demanding method in terms of computational resources. These results show that in addition to the time saved during the meshing stage of an engineering problem, reductions of time cost can also be achieved with IBMs during the simulation.

Model	**Cell count (millions)**	**Overall CPU time (h)**	**time step (s)**	T_1 **(s)**
Body-fitted	9	218	$2.5e^{-4}$	$1.4e^{-6}$
simple NS-IBM	5.3	34	$1e^{-3}$	$1.4e^{-6}$
improved NS-IBM	5.3	160	$1e^{-3}$	$6.8e^{-6}$
LB-IBM (fine)	30.8	49	$1e^{-4}$	$3.7e^{-8}$

Table 6.5: Computational costs of the mixer simulation with the different numerical methods.

Chapter 7

On the transport of thin flexible solids with diffuse IBMs

7.1 Introduction

This chapter presents the work carried out on the modelling of the interactions between fluids and thin flexible structures. The formulation of the model is introduced before discussing the results. The overall model combines four different numerical methods for each aspect of the interactions:

- a fluid solver for the governing equations of the carrier phase
- a solid solver to model the deformation of the thin structure
- an IBM for the coupling between both fluid and solid solvers
- two types of collision models for the coupling solid-solid

The two different approaches are considered to solve the fluid equations, which may be expressed as in the Navier-Stokes or the Lattice-Boltzmann framework. While the solid solver remains identical in both cases, the IBM coupling is obviously influenced by the choice of the fluid equations and inherent solver.

The FSI model is validated with common benchmarks in literature and assessed for engineering applications. A stronger emphasis is made on the LB-IBM as it was developed during this research. The NS-IBM is based on Huang's model [56]. The results obtained with the latter are used for comparison with the LB-IBM for the most relevant cases as defined in Section 7.3.

7.2 Modelling

7.2.1 Structural dynamics

The equation of motion of an elastic slender body is derived in a Lagrangian frame $(s_i,\ s_j)$ using the variational derivative of the deformation energy [56]. The governing equation for a two-dimensional solid, defined by the Lagrangian points $\mathbf{X}$, in a 3D flow reads:

$$\Delta\rho\frac{\partial^2\mathbf{X}}{\partial t^2} = \sum_{i,j=1}^{2}[\frac{\partial}{\partial s_i}(\sigma_{ij}\frac{\partial\mathbf{X}}{\partial s_j}) - \frac{\partial^2}{\partial s_i\partial s_j}(\gamma_{ij}\frac{\partial^2\mathbf{X}}{\partial s_i\partial s_j})] - \lambda\frac{\partial\mathbf{X}}{\partial t} + \Delta\rho\mathbf{G} - \mathbf{F_{ibm}} + \mathbf{F_c} \quad (7.1)$$

In Equation 7.1, $\sigma_{ij} = \phi_{ij}(T_{ij} - T_{ij}^0)$ and $\gamma_{ij} = \zeta_{ij}(B_{ij} - B_{ij}^0)$, where the terms $T_{ij} = \frac{\partial\mathbf{X}}{\partial s_i} \cdot \frac{\partial\mathbf{X}}{\partial s_j}$ and $B_{ij} = (\frac{\partial^2\mathbf{X}}{\partial s_i\partial s_j})^2$ refer to the stretching/shearing effect and the bending/twisting effect respectively. The superscripts 0 stands for the initial value. The constants ϕ_{ij} and ζ_{ij} are the tension and bending coefficients. The term $-\lambda\frac{\partial\mathbf{X}}{\partial t}$ denotes potential damping effects with the associated coefficient λ. $\mathbf{G}$ is the gravity acceleration. The term $\Delta\rho = \rho_s - c\rho_f$ denotes the density difference between the filament and the surrounding fluid (which is null for neutrally buoyant objects), with ρ_s and ρ_f the solid and fluid densities respectively and c the physical thickness of the solid. This density difference comes from Archimedes' law for the gravity term and from the virtual mass for the inertial term [56]. Finally, $\mathbf{F_{ibm}}$ and $\mathbf{F_c}$ represent respectively the influence of the fluid on the solid through the IBM forcing and the influence from other rigid solids or boundaries through a collision force.

A first order Finite Difference scheme is used to discretized the motion equation of the slender solid, similarly to the approach developed in [56]. The time discretization of Equation 7.1 is semi-implicit. The external forces are treated explicitly, while the internal elastic force is treated implicitly. However, the non-linearity introduced by the elastic and bending terms T_{ij} and B_{ij} is solved by using the position vectors $\mathbf{X}$ of the previous time step. Other methods to tackle this issue in literature include the use of a Newton method [35] or a linearization around $\delta\mathbf{X} \approx 0$ for $\delta\mathbf{X} = \mathbf{X^{n+1}} - \mathbf{X^n}$ as [22].

The final form of the discretized equation is:

$$[(\Delta\rho+\Delta t\frac{\lambda}{2})\mathbf{I}-\Delta t^2\mathbf{K}]\mathbf{X^{n+1}} = (2\Delta\rho)\mathbf{X^{n}}-(\Delta\rho-\Delta t\frac{\lambda}{2})\mathbf{X^{n-1}}+\Delta t^2\Delta\rho\mathbf{G}-\Delta t^2\mathbf{F^{n}_{ibm}}+\Delta t^2\mathbf{F^{n}_{c}} \tag{7.2}$$

where the superscript n stands for the time step, and $\mathbf{I}$ and $\mathbf{K}$ represent respectively the identity matrix and the discretized operator of the elastic force. A staggered grid is used for the discretization of the flexible structure. The Lagrangian position vectors are defined at the nodes, as well as the bending coefficients. On the contrary, the tension coefficients are defined at the interfaces between the nodes. Such a discretization procedure is illustrated in Figure 7.1. The linear system resulting from Equation 7.2 is solved with an iterative biconjugate gradient stabilized method (BiCGSTAB) [141].

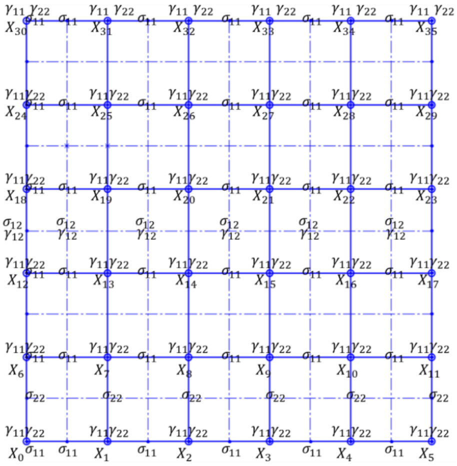

$\gamma_{11}\ \gamma_{22}$
σ_{11}
σ_{12}
γ_{12}
σ_{22}
X_{30} X_{31} X_{32} X_{33} X_{34} X_{35}
X_{24} X_{25} X_{26} X_{27} X_{28} X_{29}
X_{18} X_{19} X_{20} X_{21} X_{22} X_{23}
X_{12} X_{13} X_{14} X_{15} X_{16} X_{17}
X_{6} X_{7} X_{8} X_{9} X_{10} X_{11}
X_{0} X_{1} X_{2} X_{3} X_{4} X_{5}

Figure 7.1: Lagrangian discretization of the flexible structure for a 5 × 5 grid.

Depending on the case, two different types of boundary conditions can be applied (for $i = 1, 2$) for Equation 7.1 to mimic specific constraints at the extremities of the structure:

- a fixed condition with $\mathbf{X} = \mathit{constant}$ and $\frac{\partial^2\mathbf{X}}{\partial s_i^2} = (0, 0)$. This defines a simply-supported condition, where the structure position is constrained. In addition, the structure is free to rotate where the condition is applied, but does not experience any bending moments.

- a free end condition with $\frac{\partial^2 \mathbf{X}}{\partial s_i^2} = (0,0)$, $\frac{\partial^3 \mathbf{X}}{\partial s_i^3} = (0,0)$, $\sigma_{ij} = 0$ and $\gamma_{ij} = 0$. This boundary condition assumes that there is neither bending moment nor shearing force at the free end of the structure.

Finally, it is worth mentioning that the inextensibilty criterion can be approximately achieved if needed by using a large stretching coefficient for the tension term. Such a method is also employed in [56], [133] and [28]. The inextensitbilty criterion can also be strictly achieved with an additional constraint for the solid equation as it is proposed in [55] and [35].

7.2.2 Coupling fluid-solid with IBM

In this section, the formulation of the force $\mathbf{F_{ibm}}$ in Equation 7.1 representing the influence of the fluid on the solid is presented. The estimation of this hydrodynamic force is challenging. One way to do it is to directly consider the pressure and the shear stress on the solid surface as in [99] and [28]. Such a method can be described as a physical approach, but requires sufficient refinement and specific care for the interpolation procedures. The other way for estimating $\mathbf{F_{ibm}}$ is to base it on the opposite action from the solid on the fluid, as given by the reaction principle. This method can be described as a numerical approach as it is based on the numerical forcing term ensuring the no-slip condition. Although also used for slender flexible solids [133], [159],[35], the main disadvantage of the method lies is in the fact that the numerical estimation of the force does not necessarily provide a correct estimate of the stresses governing the motion of the flexible solid. This is especially the case when it concerns structure thickness at scales lower than the grid scale, for which the action of the fluid on both sides of the quasi two-dimensional structure cannot be decomposed. A compromise has to be made between accuracy on one hand, and stability and low computational cost on the other hand. The second option is adopted in this study for both physical solvers.

Coupling in the Lattice-Boltzmann solver

In the Lattice-Boltzmann approach, the action of the fluid on the solid is derived from the Lagrangian definition of the body force $\mathbf{g}$ computed with the MDFM introduced in Section 6.2. The momentum forcing term in Equation 7.1 reads then:

$$\mathbf{F_{ibm}} = \rho_f \epsilon \mathbf{g} \tag{7.3}$$

A length ϵ characterizing the structure is appearing in Equation 7.3, because of the necessity to translate the volumetric body force into a surface force over the solid. This length stands for the hydrodynamic thickness of the influence of the rag, rather than the physical rag thickness itself [35], but its definition is still discussed in literature. In this study, a thickness close to the lattice cell size represents a good approximation and was found to give good results, in terms of dynamic response for a flapping flag as compared to published results (see Section 7.3.2). An estimation of the thickness is also proposed in Section 7.3.3. The total force acting on the thin solid can be calculated from both the Lagrangian and the Eulerian forcing term (as already introduced in Equation 6.6) as following:

$$\mathbf{F_{tot}} = -\rho_f \sum_{\mathbf{x}} \mathbf{g}(\mathbf{x})(\Delta x)^3 = -\rho_f \epsilon \sum_{\mathbf{X}} \mathbf{g}(\mathbf{X}) \Delta s \tag{7.4}$$

where Δs is a small element of surface assumed constant over the solid and ϵ is a thickness, which is necessary to guarantee dimensional consistency. The relationship between the Eulerian force and its Lagrangian counterpart is:

$$\mathbf{g}(\mathbf{x}) = \sum_{\mathbf{X}} \mathbf{g}(\mathbf{X}) \delta_h(\mathbf{x} - \mathbf{X})(\Delta x)^3 \tag{7.5}$$

In Equation 7.5, δ_h stands for the discrete version of the molifier used by Lai and Peskin in [78] and introduced for instance in Equation 6.4. This weighting function is defined as a multiplication of each spatial contribution as $\delta_h(x,y,z) = \frac{1}{(\Delta x)^3}\phi(\frac{x}{\Delta x})\phi(\frac{y}{\Delta x})\phi(\frac{z}{\Delta x})$, where the function ϕ reads:

$$\phi(r) = \begin{cases} \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}) & \text{if} \quad |r| < 1 \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}) & \text{if} \quad 1 \leq |r| < 2 \\ 0 & \text{if} \quad 2 \leq |r| \end{cases} \tag{7.6}$$

With Equation 7.4 and Equation 7.5, and assuming $\Delta s = (\Delta x)^2$, one can conclude that $\epsilon = \Delta x$. Using this definition of the thickness in Equation 7.3 is merely assuming that this global approximation is also valid locally. While Favier et al. [35] used a local definition of the hydrodynamic thickness based on the discrete molifier δ, Huang et al. [55] adopted the use of the length scale.

Coupling in the Navier-Stokes solver

The momentum forcing term employed in [56] and [57] is adopted for the Navier-Stokes approach to deal with the fluid - solid interaction. In this formulation, two sets of Lagrangian points are used to describe the flexible structure and to solve its dynamics: the physical structure points $\mathbf{X}$ calculated from the motion of the flexible solid (Equation 7.1) and the immersed boundary points $\mathbf{X_{ib}}$ obtained from the local fluid velocity $\mathbf{U_{ib}}$. The momentum forcing $\mathbf{F_{ibm}}$ in the solid equation reads:

$$\mathbf{F^n_{ibm}} = -K_{ibm}(\mathbf{X^{n+1}_{ib}} - 2\mathbf{X^n} + \mathbf{X^{n-1}}) \tag{7.7}$$

where $\mathbf{Xib^{n+1}}$ is the new estimated position of the IB points which are obtained from the local velocity as $\mathbf{X^{n+1}_{ib}} = \mathbf{X^n_{ib}} + \mathbf{U^{n+1}_{ib}}\Delta t$. The local velocity is obtained from linear interpolation of the velocities in the surrounding Eulerian cell center. This type of forcing can be considered as a penalty approach as it uses a very large constant value K_{ibm}.

In order to complete the coupling, the influence of the solid has to be spread to the surrounding fluid domain. This Eulerian forcing term introduced in the Navier-Stokes momentum equation is obtained from the Lagrangian momentum forcing in the following way:

$$\mathbf{f^n} = \int_\Gamma \mathbf{F^n_{ibm}}(\Gamma)\delta(\mathbf{x} - \mathbf{X^n}(\Gamma))d\Gamma \tag{7.8}$$

where Γ is the surface and support of the solid. The smearing process is performed here also thanks to the standard discrete delta function δ employed by Lai and Peskin in [78].

7.2.3 Collision modelling

In realistic engineering applications, solid-solid interactions may occur and need therefore to be modelled. These interactions include collision between two flexible solids (or collision between two parts of the same flexible solids i.e. self-collision) and collision between a flexible and a rigid solid. As a preliminary study, only the latter is considered in this work. While models have already been developed for interactions between rigid solids and boundaries, as collision between particles for instance, investigations on collision problems for flexible solids are extremely scarce in literature. Furthermore, an accurate modelling of the collision mechanism can be challenging, due to multiple effects taking place. To the authors knowledge, Huang et. al [55] have developed the unique collision model for flexible solids based on a repulsive interaction force, as adopted previously by [46] for rigid particles. In Huang's work [55], this force avoids the penetration of two side-by-side flapping filaments. However, the computation of this force does not take into account the dominant forces from the fluid-structure interactions.

In the present study two types of approach are considered. The first method is merely based on a correction of the position of the 2D flexible solid to avoid penetration with another rigid body. The second method uses a repulsive force to mimic the physical effect of the collision between flexible and rigid solids. The difference with Huang's model in [55] is that the force is derived from the fluid action in the current approach. The first collision model has been implemented in two different ways in both physical solvers, while the second collision model is identical in Navier-Stokes and Lattice-Boltzmann solvers.

In the current problems, the flexible and the rigid structures interacting together are both modelled with IBMs. For a description of the collision models, let assume $\mathbf{X_i}$ to be the Lagrangian position vectors of the flexible solid, whose motion are solved with Equation 7.1, and $\mathbf{X_j}$ to be the Lagrangian position vectors of the rigid solid, whose motion are prescribed. The distance between two points of each solid is defined as $\mathbf{d_{ij}} = \mathbf{X_i} - \mathbf{X_j}$. Finally, in the two models the Eulerian spacing Δx is used to trigger the activation of the collision algorithm.

Collision based on position enforcement

In these kinematic approaches, instead of adding a collision force, the position of the flexible solid is directly corrected after solving Equation 7.1.

Consider first the Lattice-Botlzmann solver. For each point $\mathbf{X_i}$ of the flexible solid, an ensemble S_i is generated with the points belonging to the rigid body that fall below a collision distance threshold. Hence, each points $\mathbf{X_j}$ in S_i are contributing to the correction of the position of the considered point $\mathbf{X_i}$ through an offset $\tau_\mathbf{j}$. The corrected position $\tilde{\mathbf{X}}_\mathbf{i}$ of the flexible solid point reads then:

$$\tilde{\mathbf{X}}_\mathbf{i} = \mathbf{X_i^n} + \sum_{j \subset S_i} \tau_\mathbf{j} \tag{7.9}$$

The offset avoids the flexible solid getting closer than the collision distance threshold. It is defined by:

$$\tau_\mathbf{j} = -(1 - \frac{2\Delta x}{\|\mathbf{d_{ij}}\|})\mathbf{d_{ij}} \tag{7.10}$$

In Equation 7.10, the collision distance threshold is set to $2\Delta x$. With this definition of the offsets, the flexible solid is "bouncing" in the opposite direction and it makes sure that the latter does not get closer to the chosen threshold. It is important to note that the contribution to the correction algorithm from one point of the rigid body can be contradicting a contribution from another rigid body point. In other words, a newly computed offset $\tau_\mathbf{j2}$ can break the distance threshold requirement between $\mathbf{X_i}$ and a rigid solid point $\mathbf{X_{j1}}$ considered previously for the correction. To tackle this issue, the position enforcement is computed iteratively. About ten iterations are sufficient to reach a converged solution.

For the kinematic approach in the Navier-Stokes solver, the Lagrangian points are corrected according to the local fluid velocity $\mathbf{V_l}$ as defined in Equation 7.11.

$$\tilde{\mathbf{X_i}} = \mathbf{X_i^n} + (\mathbf{V_l} \cdot \mathbf{d_{min}}) \frac{\mathbf{d_{min}}}{\|\mathbf{d_{min}}\|} \Delta t = \mathbf{X_i^n} + ((\mathbf{X_{ib,i}^{n+1}} - \mathbf{X_i^n}) \cdot \mathbf{d_{min}}) \frac{\mathbf{d_{min}}}{\|\mathbf{d_{min}}\|} \tag{7.11}$$

where $\mathbf{d_{min}} = min_j(\mathbf{d_{ij}})$ is the mimimum distance vector between $\mathbf{X_i}$ and the rigid solid points $\mathbf{X_j}$. With Equation 7.11, the contribution of the flexible solid displacement toward the rigid surface is cancelled. Assuming the no-slip condition at the fluid-solid interface is satisfied, the local fluid velocity is estimated from the solid velocity.

The kinematic collision models generate a sliding mechanism parallel to the rigid body interface. Although the physics involved in the collision process is not modelled in this method, one can argue that the liquid film, which should be present between the two solids, also allows the flexible structure to slide on the rigid body.

Collision based on a repulsive force

Under the assumption of fluid lubrication, there is no physical contact between the thin elastic structure and the rigid body. A film of liquid forms between the two solid objects which repulses the flexible one away from the rigid one. The presence of this film will indeed lessen the acceleration of the flexible structure towards the rigid body, avoiding penetration so as to stop its progression into the solid. In order to mimic this mechanism, the collision force can be based on the opposite of the external forces. As these forces are driving the transport of the flexible structure, such formulation of the collision force is indeed representing a deceleration of the flexible solid towards the rigid one. The collision force is thus calculated using both the gravitational acceleration and the momentum forcing terms which define the external forces on the slender structure. The formulation of this repulsive force for each Lagrangian point of the flexible structure is given in Equation 7.12:

$$\mathbf{F_c} = k_w((\mathbf{F_{ibm}})_n - \Delta\rho(\mathbf{G})_n)\frac{2\Delta x - \|\mathbf{d_{min}}\|}{\Delta x} \tag{7.12}$$

where k_w is a stiffness constant parameter used to control the strength of the collision force. The force is active only in the vicinity of the rigid body and its activation is triggered by a threshold defined as $2\Delta x$. One can note that the force intensity is linearly decreasing with the distance between both solids. There are two limitations for this approach: (i) the estimation of the user-defined parameter k_w can largely influence the collision process, (ii) the repulsive force only depends on local information on the IBM forcing.

Influence of friction

If one assumes that the fluid is not likely to be trapped between the flexible slender structure and a solid obstacle, it can then be relevant to consider a decelerating force modelling the solid-solid friction. This may be necessary with certain materials for the flexible solids or under specific flow conditions. In the FSI model presented here for the LB-IBM, an attempt to represent a deceleration force mimicking the influence of friction is proposed. This algorithm can be combined with the kinematic collision model. The collision force $\mathbf{F_c}$ is thus set to zero, but an artificial friction force $\mathbf{F_f}$ is added on the right-hand-side of Equation 7.1. For a point $\mathbf{X_i}$ of the flexible solid, this force reads:

$$\mathbf{F_{f,i}} = -\frac{\gamma_i}{\Delta t}\mathbf{V_{T,i}} \tag{7.13}$$

Rather than a physical friction model based on the normal load exerted by the flexible surface on the rigid one, the current approach merely reproduces the effects of friction, which can be defined as a tangential deceleration. The friction force defined in Equation 7.13 is thus based on the acceleration force for the driving force, and uses an user-defined coefficient. The friction force involves the estimation of the local tangent velocity $\mathbf{V_{T,i}}$ to the flexible surface. This velocity is defined as a relative velocity between the two solids in contact:

$$\mathbf{V_{T,i}} = \mathbf{V_i} - \bar{\mathbf{V_R}} - [(\mathbf{V_i} - \bar{\mathbf{V_R}}) \cdot \mathbf{n_i}]\mathbf{n_i} \tag{7.14}$$

where $\bar{\mathbf{V_R}}$ represents the mean velocity of the rigid solid. The local velocity of the flexible solid in Equation 7.14 is defined as $\mathbf{V_i} = \frac{\mathbf{X_i^n} - \mathbf{X_i^{n-1}}}{\Delta t}$. Equation 7.13 involves also the coefficient γ_i, which adjusts the strength of the friction through the coefficient c_f as detailed in Equation 7.15:

$$\gamma_i = c_f \frac{\sum_{j \subset S_i} \tau_{\mathbf{j}} \cdot \mathbf{n_i}}{\|\mathbf{V_{T,i}}\| \Delta t} \tag{7.15}$$

The coefficient γ_i is scaled with the normal component of the offset corrections $\tau_{\mathbf{j}}$ computed during the kinematic collision model (see Equation 7.10). For a given tangential velocity, this allows to have a friction force which increases linearly with the closeness to the rigid obstacle wall. Such a formulation reproduces the effect of a realistic friction force, which must be a function of the normal component of the force applied by the flexible structure on the rigid boundary. In Equation 7.14 and Equation 7.15, $\mathbf{n_i}$ stands for the local normal to the flexible surface, as illustrated in Figure 7.2. The latter summarizes the different quantities involved in the friction algorithm. One can note finally that for the sake of simplicity the current approach uses the direction tangential to the flexible surface for computing the deceleration force, rather than a more physical correct use of the direction tangential to the rigid solid surface. However, in the cases investigated in this research, the flexible structure is very likely prone to wrap up around the rigid solid. In such configuration, both tangential directions are locally the same.

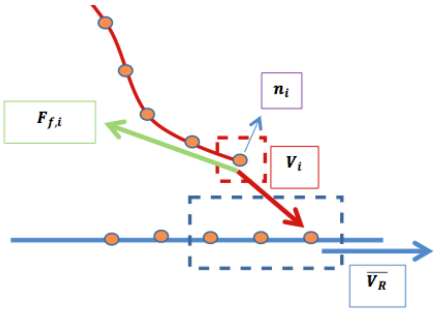


Figure 7.2: Sketch of the friction model. The focus is done on the Lagrangian point $\mathbf{X_i}$ with velocity $\mathbf{V_i}$ which belongs to the flexible solid. The delimited dashed blue domain represents the Lagrangian points $\mathbf{X_j}$ which belong to the rigid solid and contribute to the both the kinematic collision and the friction force (ensemble S_i).

7.2.4 Summaries of the FSI algorithm

In this section, the models developed to study the transport of thin flexible structures in fluids are summarized into diagrams. Figure 7.3 shows the FSI model within the Lattice-Boltzmann solver and Figure 7.4 shows the FSI model within Navier-Stokes.

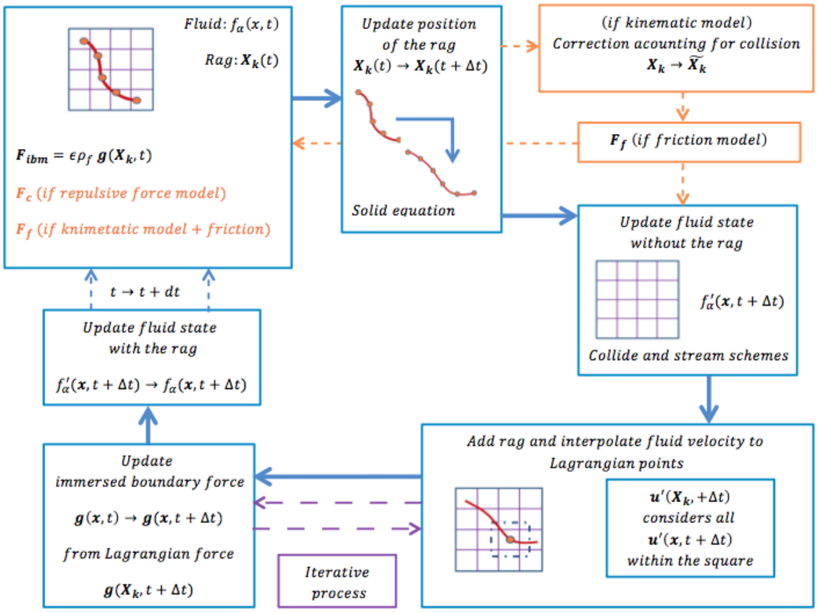
Besides the difference of fluid solvers, the main differences between the two FSI algorithms are listed below:

- The no-slip condition is satisfied with a penalization approach in the NS-IBM involving a penalization coefficient. In the LB-IBM this condition is satisfied with an iterative direct-forcing approach.

- With the NS-IBM, the momentum forcing is defined firstly for the Lagrangian structure to follow the fluid flow. The forcing term is then translated in an Eulerian forcing for the fluid equations. With the LB-IBM, the momentum forcing is defined firstly in the Eulerian context for the fluid to be at the solid velocity. The Lagrangian counterpart of the force is then used to solve the solid equation.

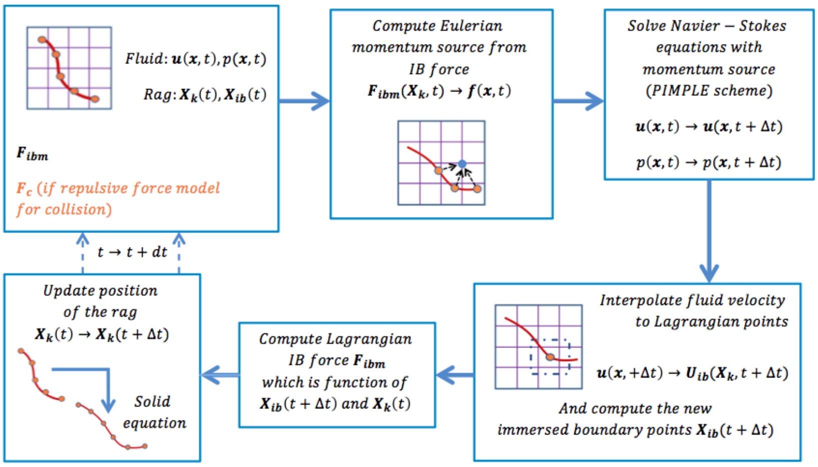
- Over one time step, the solid equation is solved first followed by the fluid equations for the LB-IBM. The opposite is performed for the NS-IBM.

- The FSI model of the NS-IBM uses a second set of Lagrangian points to estimate the forcing term and thus to solve the solid equation.
- The kinematic collision model is slightly different for both FSI models.



Fluid: $f_\alpha(x,t)$
Rag: $\boldsymbol{X}_k(t)$
$\boldsymbol{F}_{ibm} = \epsilon\rho_f\, \boldsymbol{g}(\boldsymbol{X}_k,t)$
$\boldsymbol{F}_c$ (if repulsive force model)
$\boldsymbol{F}_f$ (if knimetatic model + friction)
Update position of the rag
$\boldsymbol{X}_k(t) \rightarrow \boldsymbol{X}_k(t+\Delta t)$
Solid equation
(if kinematic model)
Correction acounting for collision
$\boldsymbol{X}_k \rightarrow \widetilde{\boldsymbol{X}}_k$
$\boldsymbol{F}_f$ (if friction model)
Update fluid state without the rag
$f'_\alpha(\boldsymbol{x},t+\Delta t)$
Collide and stream schemes
$t \rightarrow t + dt$
Update fluid state with the rag
$f'_\alpha(\boldsymbol{x},t+\Delta t) \rightarrow f_\alpha(\boldsymbol{x},t+\Delta t)$
Update immersed boundary force
$\boldsymbol{g}(\boldsymbol{x},t) \rightarrow \boldsymbol{g}(\boldsymbol{x},t+\Delta t)$
from Lagrangian force
$\boldsymbol{g}(\boldsymbol{X}_k,t+\Delta t)$
Iterative process
Add rag and interpolate fluid velocity to Lagrangian points
$\boldsymbol{u}'(\boldsymbol{X}_k,+\Delta t)$ considers all $\boldsymbol{u}'(\boldsymbol{x},t+\Delta t)$ within the square

Figure 7.3: Summary diagram of the coupled Lattice-Boltzmann - Immersed Boundary Method for solving the transport of thin flexible structures in fluids.



Fluid: $\boldsymbol{u}(\boldsymbol{x},t), p(\boldsymbol{x},t)$
Rag: $\boldsymbol{X}_k(t), \boldsymbol{X}_{ib}(t)$
$\boldsymbol{F}_{ibm}$
$\boldsymbol{F}_c$ (if repulsive force model for collision)
Compute Eulerian momentum source from IB force
$\boldsymbol{F}_{ibm}(\boldsymbol{X}_k,t) \rightarrow \boldsymbol{f}(\boldsymbol{x},t)$
Solve Navier – Stokes equations with momentum source (PIMPLE scheme)
$\boldsymbol{u}(\boldsymbol{x},t) \rightarrow \boldsymbol{u}(\boldsymbol{x},t+\Delta t)$
$p(\boldsymbol{x},t) \rightarrow p(\boldsymbol{x},t+\Delta t)$
$t \rightarrow t + dt$
Update position of the rag
$\boldsymbol{X}_k(t) \rightarrow \boldsymbol{X}_k(t+\Delta t)$
Solid equation
Compute Lagrangian IB force $\boldsymbol{F}_{ibm}$ which is function of $\boldsymbol{X}_{ib}(t+\Delta t)$ and $\boldsymbol{X}_k(t)$
Interpolate fluid velocity to Lagrangian points
$\boldsymbol{u}(\boldsymbol{x},+\Delta t) \rightarrow \boldsymbol{U}_{ib}(\boldsymbol{X}_k,t+\Delta t)$
And compute the new immersed boundary points $\boldsymbol{X}_{ib}(t+\Delta t)$

Figure 7.4: Summary diagram of the coupled Navier-Stokes - Immersed Boundary Method for solving the transport of thin flexible structures in fluids.

7.3 Results and discussion

The problems investigated in this study are very demanding in terms of computational resources because of (i) the three dimensional nature of the cases and (ii) the importance of capturing the fluid structure interactions. A refinement limit is quickly reached. Critically coarse grids have been found to be unsuitable for accurate modelling of flow at immersed solid boundaries. For these reasons, performing a grid convergence analysis would require an extremely dense grid and would become quickly impractical. The grids considered in the present work are either based on refinement comparable to those found in the literature or merely limited by computational power. A grid convergence study is proposed for Case 3 in Section 7.3.3 for a straightforward geometry.

Different types of test cases are considered to validate the overall Lattice-Boltzmann FSI model. The practicality of the model is assessed in terms of stability and for being able to handle large deformations. Qualitative comparisons and quantitative validations with previously published results are carried out when available. A focus is made on the accuracy of the FSI model, which is also compared with its Navier-Stokes equivalent. The different test cases are listed below:

- Case 1: This first case from literature is used to assess the correctness of the FSI model in the Lattice-Boltzmann context. The behaviour of a light fabric material under the influence of wind and gravity is investigated. This case is interesting as the two internal forces (inertia force and elastic force) and the two external forces (gravity force and fluid force) are playing a role, and it is their interactions which are governing the dynamics of the structure.

- Case 2: This case is the most common benchmark case in literature for the deformation of a 2D flexible solid. The case deals with the flapping of a flag sustained by a flow and allows a quantitative validation of the FSI model. Contrary to the previous case, the gravity is not taken into account here.

- Case 3: This case presents a comparison of the two FSI models developed in each fluid solver for the transport of a thin structure. The focus is made on the IBM coupling as a driving force.

- Case 4: The results of both FSI models are qualitatively compared with experimental results for the neutrally buoyant condition. This case focus on the transport and the deformation of a rag released in a water tunnel.
- Case 5: The collisions models are compared in this last case, and the sensitivity to specific parameters on the collision mechanism are studied. This problem also makes up an interesting case to compare both fluid solvers.

Regarding the meshing, the D3Q27 lattice model with twenty-seven vectors is used for the space disctretization of the fluid equations. The 2D flexible structures are discretized with uniform Cartesian grids for all the test cases. Additional three-dimensional rigid solids are considered for cases dealing with collision. These solids are modelled with surface grids composed of triangular cells. Similar resolutions are always used for the fluid and the different solid meshes in each validation cases, if not stated otherwise. This choice improves both the stability and the accuracy of the FSI models. Regarding the turbulence modelling, when relevant, a Smagorinsky based LES model is combined with the Lattice-Boltzmann fluid solver.

A one-dimensional version of the solid equation has also been implemented and coupled to the D2Q9 lattice model used for a two-dimensional discretization of the fluid equations. In this case, the elastic structure is a filament, which has been used to model flexible insect wings. The assessment of the FSI model for this type of physical problems is shown in Appendix B

7.3.1 Case 1: Flag in air under the influence of wind and gravity

A qualitative comparison to one of the case presented by Kim & Peskin in [74] is performed in this section. The study investigates the combine effect of wind and gravity. As the gravity force is acting downward and the flag is fixed from one side, the latter is expected to sag. In the same time a wind flow is imposed from the side of the domain to counterbalance the gravity, and can generate a flapping mechanism because of the flag's inertia.

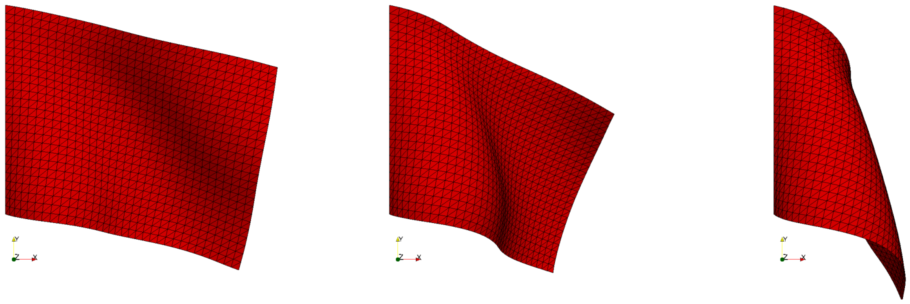
Two different computations are performed. Theirs parameters are listed in Table 7.1. For both simulations, the gravity is taken into account. In the first simulation no wind is imposed to compensate for the gravity, while a wind speed of $4.6m/s$ is imposed for the second simulation. In their model, Kim & Peskin [74] derived also the internal solid forces from the variational derivative of the energy, leading to the tension and bending terms present in Equation 7.1. However, as the author's do not provide the constant coefficients governing these two forces, a precise quantitative comparison with their numerical results is difficult. For this reason, a purely qualitative comparison is made for this case.

The current problem is interesting as it investigates the realistic behaviour of a light fabric in air. With the parameters of Kim & Peskin [74], the case considers a density difference between the solid and the fluid which is about one order of magnitude smaller than the density of the carrier phase.

Case	Re	Fr	$\Delta\rho$	ϕ	ζ	Δt
Sim. 1	370	∞	0.09	100	$1 \cdot 10^{-6}$	$4 \cdot 10^{-5}$
Sim. 2	370	2.94	0.09	100	$1 \cdot 10^{-6}$	$4 \cdot 10^{-5}$

Table 7.1: Parameters for Case 1 (flag in air under the influence of wind and gravity).

The shape of the flag is shown at different times in Figures 7.5 and 7.6 for the Sim. 1 and 2 respectively. In the absence of wind, the flag falls down immediately and will eventually stop moving. Depending on the tension and bending parameters, the flag might fold over itself or might rotate around its fixed edge, because of the momentum created from the fall. For the second case with the presence of the wind, the flag is still sagging slightly due to the gravity force. However, the combination of the wind force and the flag inertia generates a flapping mechanism.

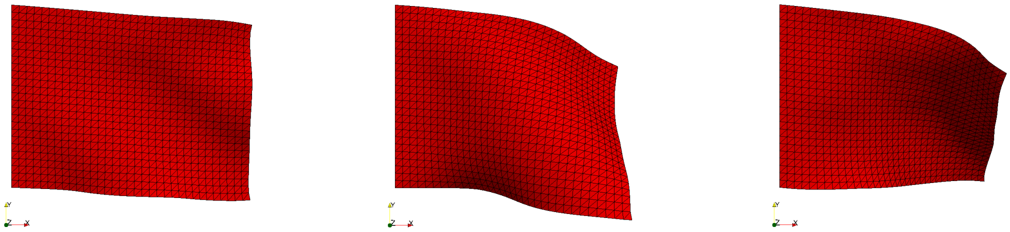


(a) $t = 0.4s$

(b) $t = 0.8s$

(c) $t = 1.2s$

Figure 7.5: Snapshots of the flag influenced only by gravity at different times (Sim 1.)



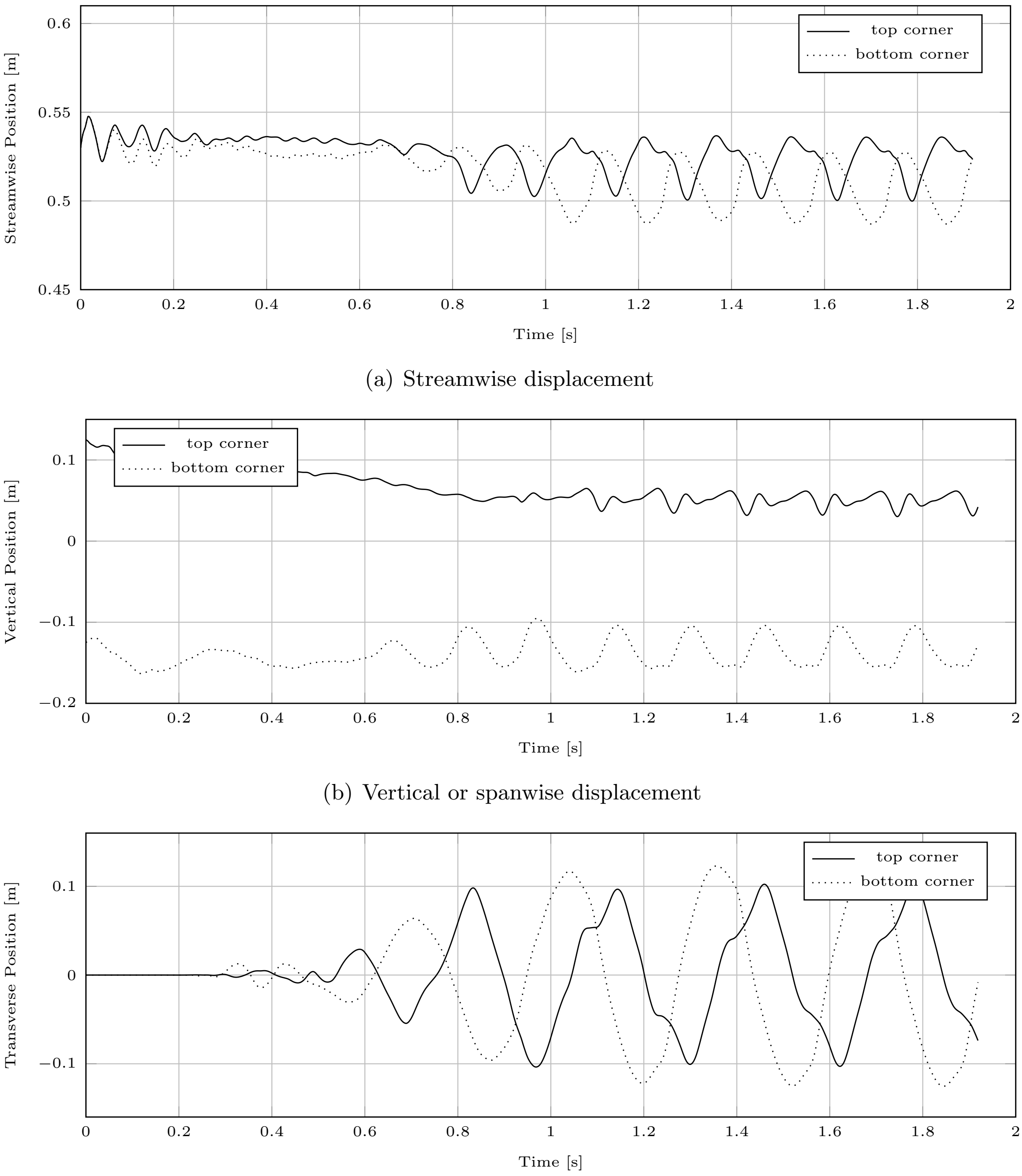
(a) $t = 0.4s$

(b) $t = 0.8s$

(c) $t = 1.2s$

Figure 7.6: Snapshots of the flag influenced by both the gravity force and the wind force at different times (Sim 2.).

Figure 7.7 shows the displacement of the flag top and bottom free corners in the presence of a wind force (Sim 2.), for each coordinates. Since the gravity force is applied in the plan formed by the streamwise and the vertical directions, it is affecting both components of the displacement. Hence, we can see the sagging effect which is breaking the symmetry of the oscillations in the direction of the wind as well as in the vertical direction. On the contrary, the symmetry of the flapping mechanism in the transverse direction, perpendicular to the gravity plan, is not affected by the sagging. The behaviour of the flag is in good agreement with the motion described by Kim & Peskin [74]. As seen by authors in [74], our current FSI model generates also a flapping motion, which is out of phase between the top and the bottom of the flag. In addition, the oscillations show a greater amplitude at the bottom in accordance with these previously published results. Approximately half a second is required for the flag to reach a periodic steady-state. The amplitudes of the oscillations obtained here are yet larger than [74], certainly because of different orders of magnitude chosen for the internal elastic forces. One can note however that Kim & Peskin do not get a perfect symmetry for the transverse displacement as it is with our model (see Figure 15 in [74]).



(a) Streamwise displacement

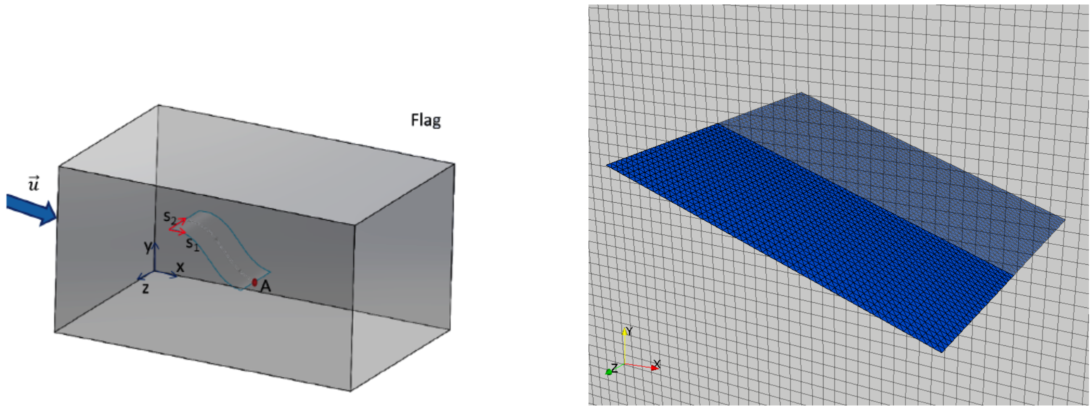
(b) Vertical or spanwise displacement

(c) Transverse displacement

Figure 7.7: Evolution of the top and bottom free corners position for a flag under the combined influence of wind and gravity.

7.3.2 Case 2: Non-dimensional flapping flag

This test case has already been defined in literature for 2D flexible structures in 3D flows ([57], [132], [82], [28]). For comparisons against the published numerical data, the following non-dimensional parameters are defined using the flag length L and the free stream velocity U_∞: the non dimensional time $t^* = tL/U_\infty$, the non dimensional length $y^* = y/L$, the Reynolds and Froud number $Re = \frac{\rho_f U_\infty L}{\mu_f}$ and $Fr = gL/U_\infty^2$, the non dimensional tension coefficient and bending rigidity, $\phi^* = \frac{\phi}{\Delta\rho U_\infty}$ and $\zeta^* = \frac{\zeta}{\Delta\rho U_\infty^2 L^2}$, and finally the non dimensional mass ratio $\rho^* = \frac{\Delta\rho}{\rho_f L}$.



(a) Computational domain

(b) Grids

Figure 7.8: Schematic diagram of the computational configuration and coordinate system and zoom on the grids

The set-up considered in this case is similar to Huang & Sung [57]. The flag shape is square with length L and initial position inclined at an angle 0.1π from the streamwise plane. The size of the computational domain is $[-L, 4L] \times [-2L, 2L] \times [-L, L]$. Both Eulerian and Lagrangian grids are discretized uniformly with a step size of $L/25$ and $L/50$ respectively. Using a larger domain as in Huang & Sung [57] was found to have a low influence on the flapping mechanism of the flag. The dimensions of the problem and its initial state are shown in Figure 7.8. The flag is fixed (pinned) from one end and free from the other ends so that the flapping under the effect of the fluid flow will be perpendicular to the flow. The non-dimensional parameters of the problem are listed in Table 7.2 for the different methods. In the present NS and LBM models, the diagonal tension coefficients had to be reduced by one order of magnitude in comparison to Huang & Sung [57], because of stability issues in the solid solver, which may be due to the different coupling techniques. For this reason, an additional simulation is considered for the LB-IBM approach with higher bending and cross coefficient for the tension. In all simulations, the time step is chosen the same for the fluid and the solid solvers.

Case	Re	Fr	ρ^*	ϕ^*_{ii}	ϕ^*_{ij}	ζ^*	Δt^*
Huang & Sung	200	∞	1	1000	10	$1 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
NS	200	∞	1	100	10	$1 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
LBM Sim. 1	200	∞	1	100	10	$1 \cdot 10^{-4}$	$3 \cdot 10^{-4}$
LBM Sim. 2	200	∞	1	100	100	$1 \cdot 10^{-3}$	$3 \cdot 10^{-4}$

Table 7.2: Non-dimensional parameters for Case 2 (flapping flag).

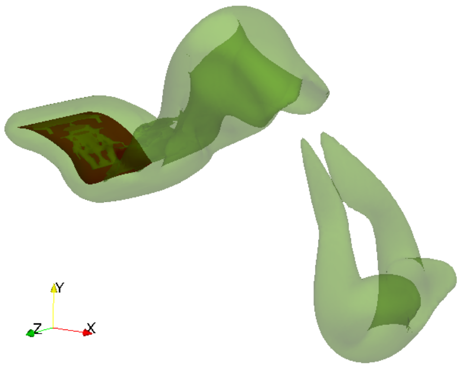


Figure 7.9: Iso-contours of the vorticity when the flag's trailing edge is near its lowest transverse position. Results obtained with a larger domain (similar to Huang & Sung [57]).

A flapping mechanism similar to the one observed by Huang & Sung [57] is obtained with both physical models. Figure 7.9 shows for instance the vortical structures shedding from the flag obtained with the Lattice-Boltzmann coupling. Their hairpin-like geometry is in good agreement with [57] and [82]. The flag undergoes sustained oscillations through the IBM coupling. The time history of the transverse displacement for the mid-point of the flag trailing edge (point A in Figure 7.8) is shown in Figure 7.10.

Although the oscillations are consistent with those observed by Huang & Sung [57], two differences can be noticed between LBM results and the reference data: (i) the amplitude of the oscillations is slightly overestimated and (ii) the presence of small perturbations at the beginning of the flapping process. The perturbations are likely to be due to the small pressure waves that can be observed during the transient, before obtaining a fully developed flow, and which are a consequence of the quasi-incompressible nature of the flow modelled with LBM. Regarding the difference of amplitude, the lower tension coefficient used in the present LB-IBM cannot solely explain this disparity. Indeed, in the NS-IBM a low tension coefficient is also used and leads to a better match with the results of Huang & Sung [57]. As the current NS-IBM is based on the same physical approach and the same IBM coupling than Huang, the differences obtained with the LB-IBM are certainly primarily due to the different flow solvers and coupling mechanisms, which can be expected to impact the solid response, especially considering the highly dynamic nature of the motion. The relative amplitude of the flag's oscillations and the resulting Strouhal number obtained with the two IBM solvers are listed in Table 7.3 and compared to other IBMs in literature. It is surprising to note that the Lattice-Botlzmann approach gives a better estimation of the Strouhal number, regarding Huang's result, in comparison to the Navier-Stokes approach used in this research. Regarding the second simulation case for LB-IBM, it appears that increasing the cross tension coefficient, and the bending coefficients, can compensate for the lack of physical stiffness due to the lower diagonal tension coefficient. The modified parameters of this simulation reduce the amplitude of the flag oscillations in accordance with Huang & Sung [57]. The trajectory of point A shows however a small delay in the flag response in this case, as it takes 4 swings to reach a fully periodic state. Finally, one can note that tuning the hydrodynamic thickness of the rag involved in the coupling (introduced in Equation 7.3) would also be a possibility in order to obtain a better match between the numerical results of LB-IBM and the previously published data. The results presented here highlight the high sensitivity to the mechanical parameters. Rather than exact physical properties, the tension and bending coefficients can be seen as numerical parameters used to approximate a desired elastic behaviour for the flexible structure. As explained in the numerical formulation of the solid solver, the tension coefficient is also used to satisfy the inextensibility criterion. This criterion is verified for both LBM simulations presented here and for the Navier-Stokes solution as well. In all cases, the maximum change in the flag surface area is found to be below 0.5%.

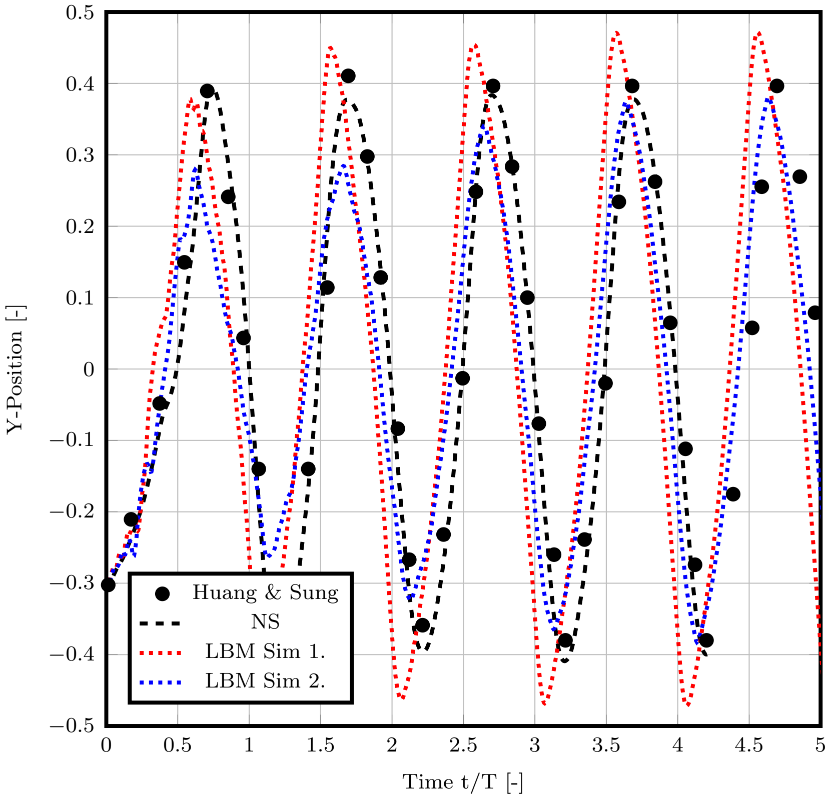


Figure 7.10: Time evolution of the transverse displacement at the trailing edge center point (point A) of the flag. The time is here non-dimensionalized with the flapping period T.

The phase relation of the transverse position and the streamwise position of point A is shown in Figure 7.11. The characteristic ”infinite-shaped” trajectory is obtained for all solvers. The phase relation computed with NS-IBM leads to the best match with Huang’s results. The shape of the trajectory is very similar in both cases, with only a slightly higher folding in the streamwise direction that is certainly due to the

Case	**Amplitude** A/L	St
Huang & Sung [57]	0.780	0.260
Tian et al. [132]	0.812	0.263
Lee & Choi [82]	0.752	0.265
de Tullio & Pascazio [28]	0.795	0.265
NS	0.783	0.317
LBM Sim. 1	0.940	0.243
LBM Sim. 2	0.758	0.289

Table 7.3: Comparison of the amplitude and the Strouhal number for Case 2.

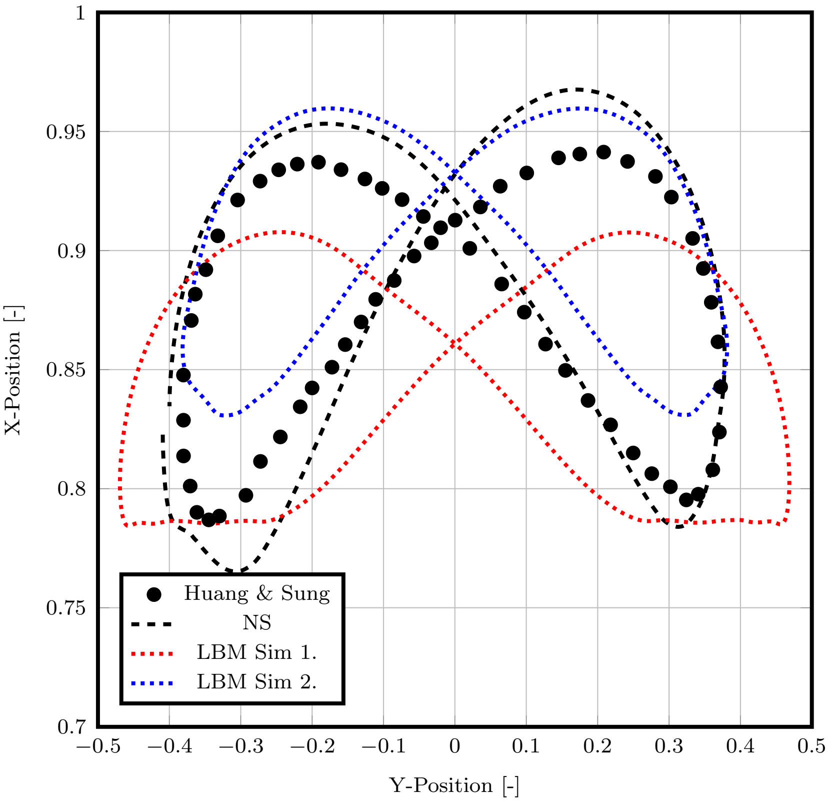
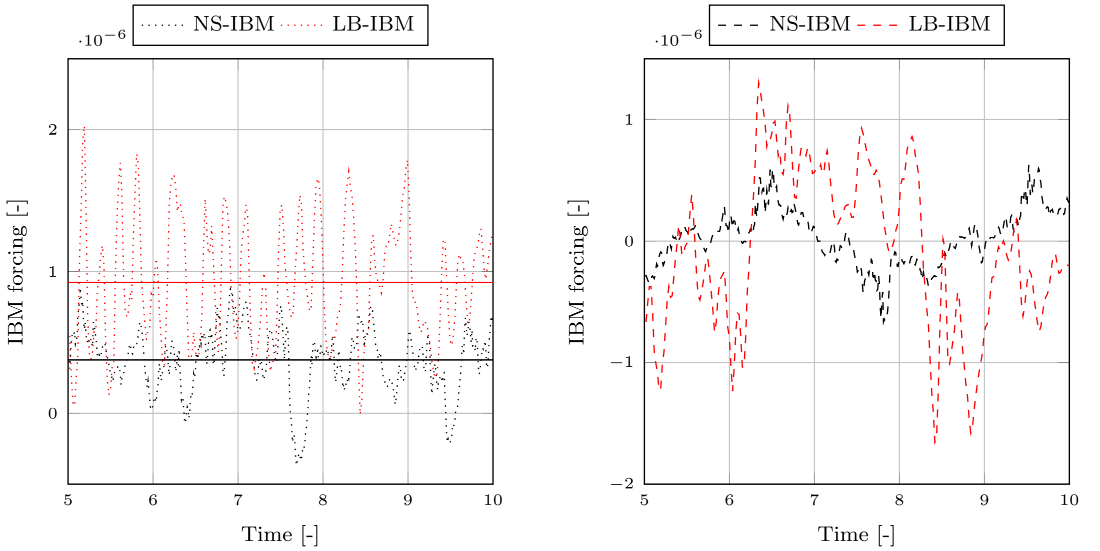


Figure 7.11: Trajectory of the trailing edge center point (point A) during one flapping period.

lower bending coefficient used. Regarding the Lattice-Boltzmann results, the higher amplitude of the oscillations can be seen as well on this figure for the Simulation 1. The parameters from Simulation 2 correct this overestimation and are giving a trajectory, which is closer in shape to the phase relation from the Navier-Stokes solvers. Finally, it is worth mentioning that a perfect symmetry is obtained with the Lattice-Boltzmann solution, but not with the Navier-Stokes based models. This was also observed in the first test case in Section 7.3.1.

In the equation governing the solid dynamics, the IBM forcing accounting for the action of the fluid is the only difference between the two coupled solvers NS-IBM and LB-IBM. It is thereby interesting to look at the magnitude of this force for the two cases. Figure 7.12 shows the evolution of the IBM force during a bit more than one swing for a corner point at the free end of the rag. The results exhibit a factor 3 between the magnitude of the forcing for the two solvers (for both the mean magnitude of the streamwise component and the magnitude of the oscillations). The transverse component of the force is centred on zero, but shows a similar difference in the magnitude of the peaks. Such large differences lead finally only to small discrepancy in the flag displacement for two reasons. Firstly, the IBM force is not the dominant force in this case, in terms of magnitude. Secondly a stronger action from the fluid is counterbalanced by a stronger effect of the internal elastic forces, which are finally restricting the displacement. As the coupling of the two solvers are based on the same principle, it is however interesting to remark that this difference in the forcing means that satisfying a no-slip condition in both solvers is leading to different level of constraint for the velocity.

The present test case has been shown to be very sensitive to the different forces and their interaction due to the choice of parameters which creates a highly dynamic system. Given the significant differences between the two numerical approaches, which lie principally in the fluid solvers and the formulation of the coupling, the level of agreement that is found here between the solvers is satisfactory for this type of highly dynamic problem. For instance, the NS-IBM used in this research is based on the exact same solid solver and IBM coupling than Huang et al. [56]. However, the different flow solver (PIMPLE scheme for our model and fractional step method with block LU decomposition for the decoupling of the velocity and pressure in [56]) are sufficient to generate non-negligible differences in the flapping mechanism.



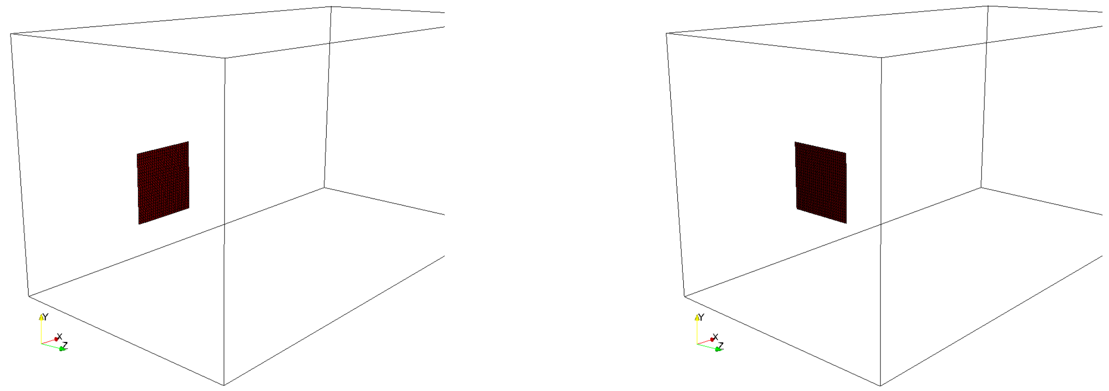
(a) Streamwise component

(b) Transverse component

Figure 7.12: Evolution of the local IBM forcing at the corner points of the rag's free end for the two fluid solvers. The capital lines represent the mean value of the streamwise component of the IBM force over this period of time. The mean value of the transverse component is close to zero for one swing.

7.3.3 Case 3: Comparison of the two solvers for the transport of a rag

In this case, the balance between the inertia force and the fluid force is investigated for the two fluid solvers. This allows to compare to what extent satisfying the no slip-condition at the fluid-solid interface through the IBM coupling is governing the motion of the thin structure. The test considers a square rag of length $L = 0.1m$ in a rectangular channel of dimensions $(8L, 4L, 4L)$. A resolution of $4mm$ is used for both fluid and solid grids. Two cases are simulated: (i) with a rag parallel to the stream and (ii) with a rag perpendicular (or normal) to the stream. In the first configuration, the viscous stresses are in theory dominant and driving the thin structure with the flow. The second configuration leads on the contrary to a pressure driven motion, as the structure is offering a significant cross-section area to the flow. It is then interesting to analyse how a numerical IBM coupling, as based on a penalty type of approach for the NS-IBM or a direct-forcing type of approach for the LB-IBM, is able to reproduce the action of the fluid, without integrating the pressure and the viscous stresses over the surface of the solid.



(a) Parallel configuration

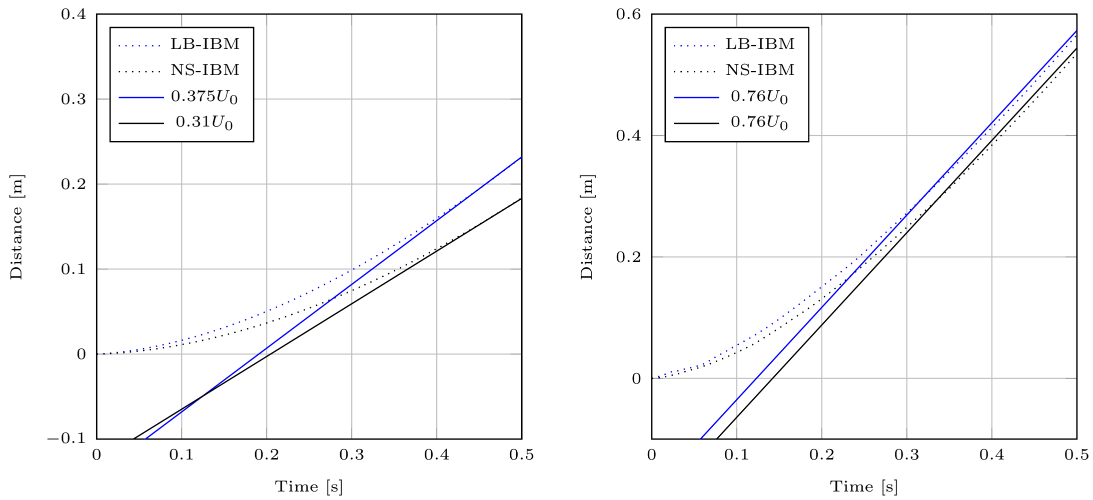
(b) Normal configuration

Figure 7.13: Computational domain and initial position of the rag for the study of its transport in a channel

The initial position of the rag is shown in Figure 7.13 for both configurations. The rags are centred in the channel. In the normal configuration, the structure is positioned at a distance L from the inlet face. In the parallel configuration, the distance between the inlet surface and the closest solid edge is $0.5L$. The density difference for the thin structure is fixed to $\Delta\rho = 1kg/m^2$, while air is considered for the carrier phase. The influence of gravity is neglected. The inlet velocity is set to $U_0 = 2m/s$, and a free-slip condition is applied at the four side walls, which ensures a uniform velocity everywhere in the domain. Given the uniform velocity, the structure only moves in the streamwise direction. In the case of a parallel rag, the structure remains perfectly aligned with the flow during its motion. In the case of a perpendicular rag, a light bending appears during the initial acceleration phase. However, the bending and tension forces are restraining sufficiently the deformation of the structure, which remains quasi-normal to the flow all along its transport. The time step is fixed to $\Delta t = 10^{-4}s$ for the two fluid solvers.

Figure 7.14 shows the streamwise position of the middle point of one rag's free edge, as a function of time, for both fluid solvers and for both parallel and perpendicular cases. As the rag initial velocity is zero, one can see the gradual acceleration of the structure, until it reaches a final constant velocity. These converged velocities are plotted as well in Figure 7.14 in function of the flow speed U_0. As expected, the structure is moving faster when it is normal to the flow i.e. when the pressure stresses are the driving force. Regarding the FSI models, the differences obtained between the two IBM couplings are relatively low. This is especially true considering the numerical nature of the IBM coupling. In the parallel configuration, the Lattice-Boltzmann solver predicts a higher final velocity. On the other hand, the final speed predicted by the two IBMs are similar in the normal configuration, but the rag modelled with LB-IBM reaches its terminal velocity slightly before. A very good agreement is found when the pressure force is dominant. This is a satisfactory result as in reality the pressure stresses are more likely to be the main driving force due to the folding of the rag.

This simple case allows us also to validate the assumption made regarding the influence of the hydrodynamic thickness of the rag arising from the translation of the volumetric body force to the the Lagrangian surface force (see Equation 7.3 in Section 7.2.2). This thickness ϵ can be seen as the ratio of forces from Equation 7.4 defined by



(a) Rag parallel to the flow

(b) Rag perpendicular to the flow

Figure 7.14: Evolution of the streamwise position of one edge middle point of the rag

$\epsilon = \frac{\|\sum_{\mathbf{x}} \mathbf{g}(\mathbf{x})(\Delta x)^3\|}{\|\sum_{\mathbf{X}} \mathbf{g}(\mathbf{X})\Delta s\|}$. This ratio has been computed during the transport of the rag in the parallel configuration for three different mesh resolutions ($\Delta x = 8mm$, $\Delta x = 4mm$ and $\Delta x = 2mm$). The resolution is chosen identical for both the Eulerian fluid mesh and the Lagrangian surface mesh. Figure 7.15 shows the evolution of $\epsilon/\Delta x$ for the three refinement levels. It appears firstly that the mean value of the ratio stabilizes relatively quickly at the start of the transport. Refining the grids leads to a decrease of both the mean value and the fluctuating component in time of the thickness. Using Richardson's method [108], it was found that the magnitude of the thickness converges with an order of 0.9 towards an extrapolated value around 0.75. Although slightly low, the order of convergence is consistent with the first-order accuracy of the method. The fact that the extrapolated value is smaller than 1, can be attributed to the molifier function and to the ratio edge length to surface area of the rag considered here. More recent results have shown that tuning ϵ to be equal to the ratio of forces for a given mesh resolution reduces the dynamic response of the flapping flag in Case 2 (in Section 7.3.2) and improves thus the results.

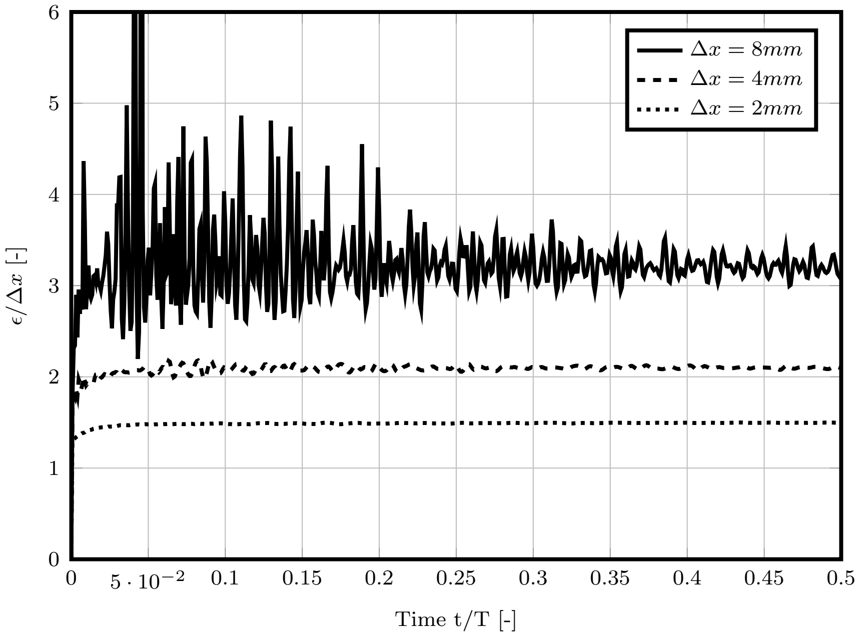


Figure 7.15: Evolution of the hydrodynamic thickness relatively to the mesh size during the transport of a rag in a parallel configuration.

7.3.4 Case 4: Rag release in a water tunnel

The release of a rag in a water tunnel is considered in this section. With liquids as carrier phase, fabrics or rags are close to the neutrally buoyant case, where the density difference between the two media is very small compared to the liquid density. A new issue may arise in this case, which is the instability of the solid solver. This problem is seen only with the Lattice-Boltzmann coupling. To overcome this issue, the density difference can be artificially increased for the inertial term in Equation 7.1, while it is kept at the desired value for the gravity term.

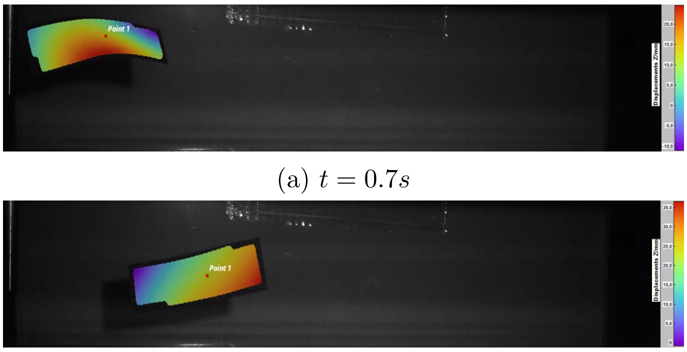
Regarding the experimental settings for the test case, the rag is initially clamped with a flagpole system at the center of the tunnel. Values of the rag elastic properties are available in Table 7.4, along with the numerical parameters of the computation. Such low density difference between the solid and the fluid corresponds to a material similar to business paper. The flow rate in the water tunnel is about $9.1m^3/h$. The numerical results of the present model are compared to experimental data obtained from Digital Image Correlation (DIC) tests, which have been performed to characterise the solid response. The DIC technique and the experimental rig are described in Appendix C. The size of the paper used for the experiments and the numerical simulations is $0.15m \times 0.048m$. Regarding the space discretization, a resolution of $\Delta x = 2mm$ is used for both the fluid and the solid grids.

Re	Fr	$\Delta\rho$	ϕ	ζ	Δt
5300	2.32	0.03	100	$1 \cdot 10^{-6}$	$1 \cdot 10^{-4}$

Table 7.4: Parameters for Case 4 (rag release).

DIC measurements are used to quantify the displacement of the rag in the three directions of space. Figure 7.16 shows the experimental data for the transport and the deformation of the rag in the tunnel at two different times after the release. During the first second, it appears that the rag is not moving much in the streamwise direction, but it is being pushed upwards by a stream-jet due to the presence of the shaft. The jet is visible in Figure 7.17, which is showing the numerical results obtained with the current Lattice-

Boltzmann model. Similarly, in the numerical simulation the rag is moving upward due to the jet, before settling down. One can also note a similar folding in the transverse direction, as illustrated in Figure 7.18. A maximum displacement of approximately $25mm$ is found numerically in the z-direction, which is very close to the displacement observed in the experiment (see at $t = 0.7s$ in Figure 7.16). This result shows that the rag behaviour and the flow characteristics are well reproduced numerically. Although, the deformation of the rag is comparable, noticeable differences can be seen between the experimental and the numerical results. The coupled LB-IBM simulates a rag that is moving more slowly in the streamwise direction, and that settles more slowly than reality. The numerical terminal velocity of the rag relatively to the average flow speed U_0 was found to be about $0.61U_0$, while experimentally a relative velocity around $0.85U_0$ is obtained independent of the flow rate. This difference in the transport can be explained by the artificial increase of the density difference in the inertial term.



(a) $t = 0.7s$

(b) $t = 1.4s$

Figure 7.16: Viewing of the transverse displacement of the rag in the water tunnel, obtained by DIC (experimental tests). The flagpole is visible on the left.

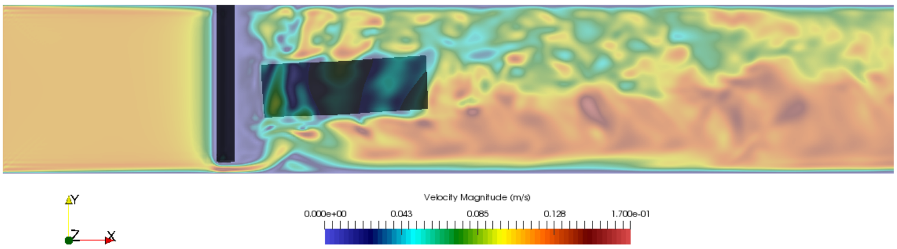


Figure 7.17: Snapshots of the rag at $t = 1.0s$, with contour of the velocity field in the background (obtained with numerical simulations).

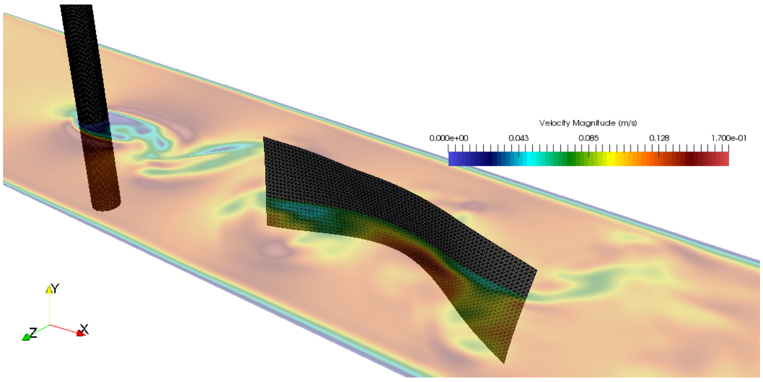


Figure 7.18: Snapshots of the rag at $t = 3.5s$ highlighting its folding.

In order to quantify the influence of the increased density in the inertial term, the current numerical case is simulated for three different values of this modified density $\tilde{\Delta\rho}$. The numerical parameters are identical to the previous simulation (see Table 7.4), except for the flow rate which is fixed to $36m^3/h$ leading to a Reynolds number around 21600. The displacement of the rag is followed during the simulation in each case. Figure 7.19 shows the streamwise displacement for the top right corner of the rag. When reducing the density difference, a convergence is visible for the rag terminal speed, which appears to be around $0.78U_0$. This corresponds to a 8% relative difference with the experimental rag velocity of $0.85U_0$. However, no clear convergence is obtained regarding the transient process. The artificially increased density delays the moment when the rag reaches its terminal speed. The discrepancy highlighted here shows that the neutrally buoyant case is not yet reached for $\tilde{\Delta\rho} = 10kg/m^2$ and can explain the slower motion obtain with the current Lattice-Boltzmann model due to an augmented inertia, in comparison to the experimental results.

The results of the current LB-IBM are compared also to the NS-IBM based on Huang's model [57] implemented in OpenFOAM®. In this comparison a high flow rate of $36m^3/h$ is again imposed for the tunnel inflow. A shorter rag is however considered here, with a size of $0.1m \times 0.048m$. All parameters are similar for both physical approaches, except for the inertial term. In the the Navier-Stokes method, a density difference of $\Delta\rho = 0.03kg/m^2$ mimicking neutrally buoyant conditions can successfully be considered, while this density is artificially increased to $10kg/m^2$ for the Lattice-Boltzmann method. The transport of the rag can be viewed in Figure 7.20 for both NS-IBM and LB-IBM. The figure shows snapshots of the rags at different times after the release. It is interesting to see that the combination of a high flow rate and a short rag generates a loss of alignment for the flexible structure. In this configuration, the pressure force is dominant and a higher velocity is expected, as seen in the previous section. The rag terminal speed has indeed found to be very close to U_0 with NS-IBM. With LB-IBM, a relative velocity of $0.82U_0$ is obtained. This indicates that the augmented inertia has a stronger impact when the rag is not aligned with the stream. Although unsynchronised, Figure 7.20 shows that the rag's motion and deformation are however in good agreement between both models. A similar interaction with the vortices generated by the shaft is indeed predicted, leading to a roughly 90° rotation of the rag around both axis Y and Z perpendicular to the streamwise direction.

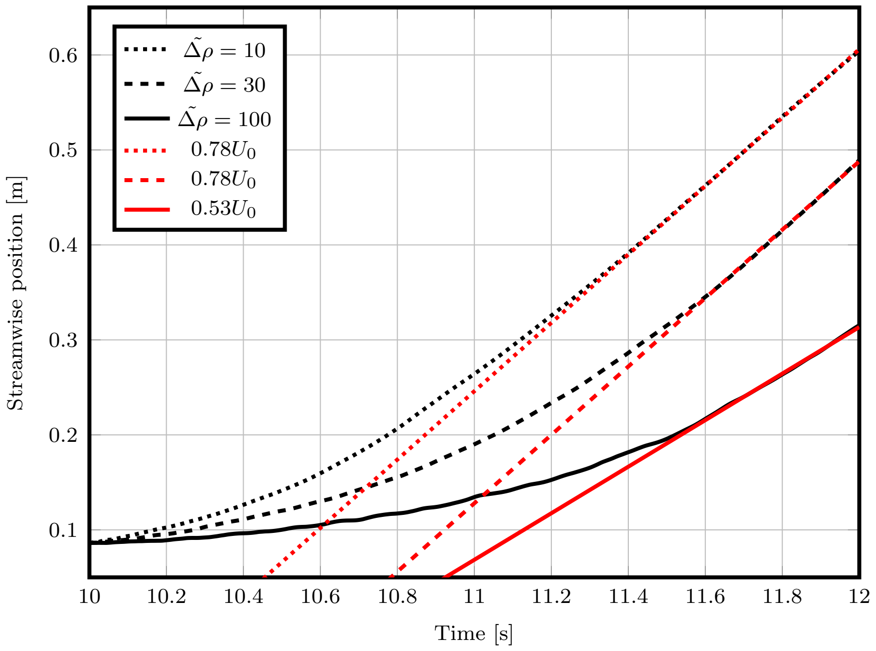


Figure 7.19: Time evolution of the streamwise displacement of the rag top right corner. The rag is released after $10s$ of a single phase simulation in order to obtain a fully developed flow.

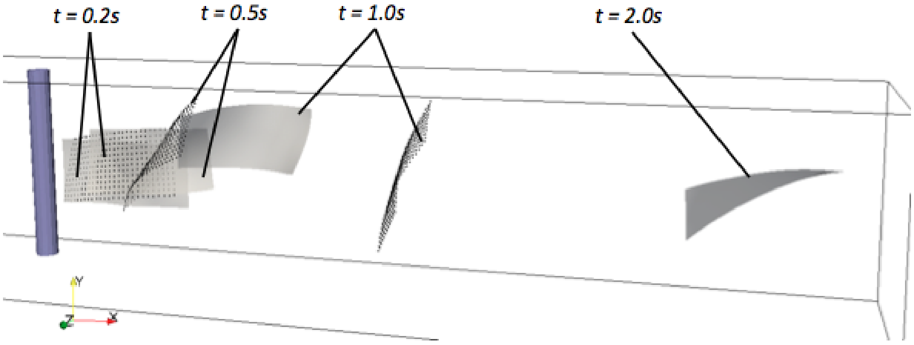


Figure 7.20: Snapshots of the rags at times $t = 0.2s$, $t = 0.5s$ and $t = 1.0s$ for both Lattice-Boltzmann solver (represented by grey surfaces) and Navier-Stokes solver (represented by black points), and at $t = 2.0s$ for the Lattice-Boltzmann solver only (as the rag compute with NS-IBM already exited the domain).

7.3.5 Case 5: Investigations on the modelling of collision

In this test case, the capabilities of the kinematic model and the force based model (described in Section 7.2.3) for the collision are assessed. More precisely, the results from the two collision models and the two numerical fluid solvers are compared. Due to the absence of published experiments regarding collision processes involving thin flexible structures, a quantitative validation is currently not possible.

The case considers the collision between a flexible 2D rag with a fixed 3D cylindrical obstacle. Gravity is not taken into account in this problem. The rag is transported in a rectangular channel flow and is initially placed perpendicularly to the stream. The initial configuration of the problem is shown in Figure 7.21. The dimension of the rag is $0.1m \times 0.1m$ and the domain is $0.4m \times 0.2m \times 0.2m$. The cylinder has a diameter of $0.05m$. An uniform velocity of $2m/s$ is imposed at the inlet, an outflow condition is defined for the outlet behind the cylinder, while all other boundaries are defined as no-slip walls. The grid shown in the Figure 7.21 involves 26 nodes in both direction for the rag, which corresponds to a spacing $\Delta s = 0.004m$. The same spacing is used to create the grid of the cylinder and the Eulerian grid for the fluid. This refinement corresponds to the medium grid. Two other grids are also considered in this problem, with a spacing of $\Delta s = 0.008m$ and $\Delta s = 0.002m$ for the coarse and the fine grid respectively. The three grids allow to study the influence of the resolution on the collision mechanism. The time step is set to $\Delta t = 0.1ms$. The kinematic viscosity is fixed to $\nu = 0.005m^2/s$, which leads to a laminar flow regime with a Reynolds number of 40.

General results and description of the collision process

The collision process is first examined with the kinematic model and the fluid considered is firstly air ($\rho_f = 1.2kg/m^3$). The density difference $\Delta\rho$ is set to $0.1kg/m^2$. The bending and tension coefficients are respectively fixed to $\phi = 100$ and $\zeta = 10^{-6}$. Each FSI computation is started from a fully developed flow in the channel obtained from a single-phase simulation (with the absence of the rag).

The collision of the rag is illustrated in Figure 7.22, which shows its position at

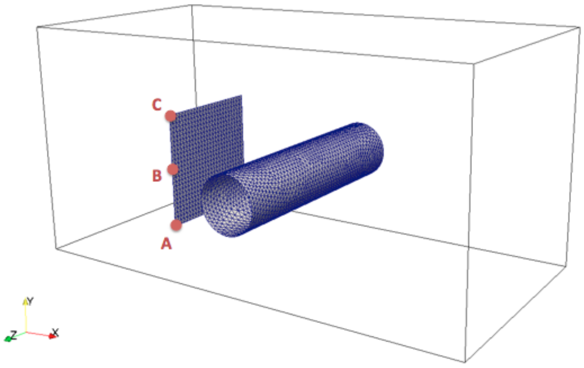


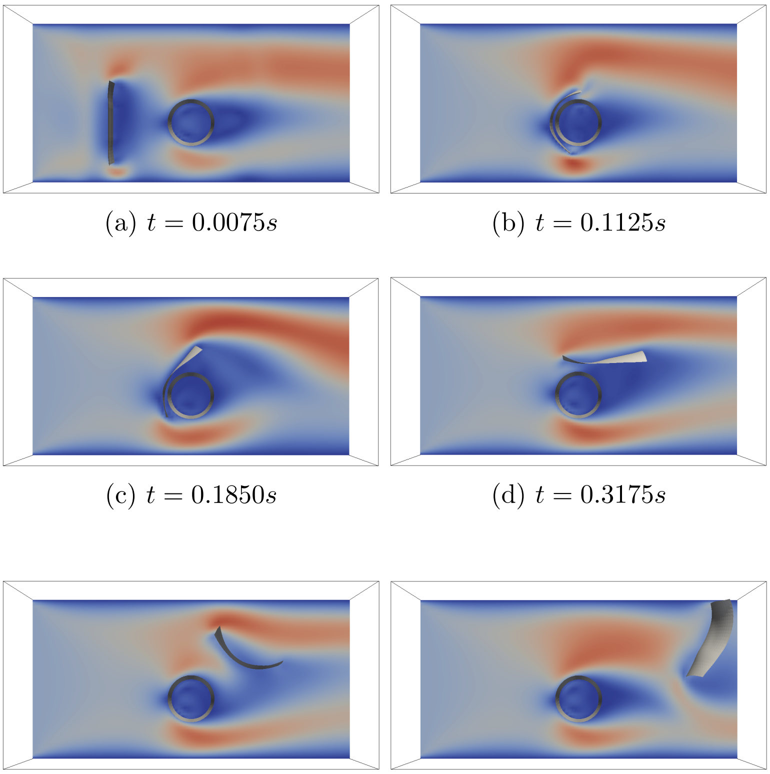
Figure 7.21: View of the computational domain and the grids (medium refinement) used for the cylindrical obstacle and for the rag (at its initial position). The trajectory of the rag is extracted for the points A, B and C.

different times. The introduction of the rag decelerates the flow around the flexible structure, because of its initial zero velocity. When the rag is getting closer to the obstacle, the kinematic collision allows the flexible structure to wrap itself around the cylinder. There is no direct contact between the two, and the collision distance is a user-defined parameter. The rag is eventually pulled from the upper part of the domain because of the higher velocity. Finally, for a relatively stiff material as it is simulated in this particular case, the rag is turning over itself once behind the cylinder. This is due to the difference of pressure between the rag downstream and upstream positions.

The displacements of the three points A, B and C on one edge of the rag (as illustrated in Figure 7.21) is examined in Figure 7.23. The collision process can be decomposed in four mechanisms:

1. The first part corresponds to the displacement of the rag before the collision with the cylinder. The trajectory is linear as the rag is released in an uniform flow.

2. The second part corresponds to the collision. The rag middle point (point B) remains behind the stagnation point, while the two tip points continue their streamwise route as the rag embraces the shape of the obstacle. Then, the inextensibility of the rag enforced by the internal tension force leads to the two tips to be pulled backward, until the rag reaches a quasi-stable position.



(a) $t = 0.0075s$

(b) $t = 0.1125s$

(c) $t = 0.1850s$

(d) $t = 0.3175s$

(e) $t = 0.3600s$

(f) $t = 0.4375s$

Figure 7.22: Snapshots of the rag at different times during the collision process. The contour of the velocity magnitude is shown in the background.

3. The third part corresponds to the slow sliding of the rag toward the upper part of the cylinder. In this part, the lower point (point A) is going upstream, while the two other points are slowly moving downstream.

4. The fourth and final part corresponds to the release of the rag in the wake of the cylindrical obstacle.

Sensitivity to numerical and physical parameters

The influence of the spatial resolution on the collision mechanism is analysed. The same default parameters as previously are considered here. Figures 7.24 and 7.25 shows the

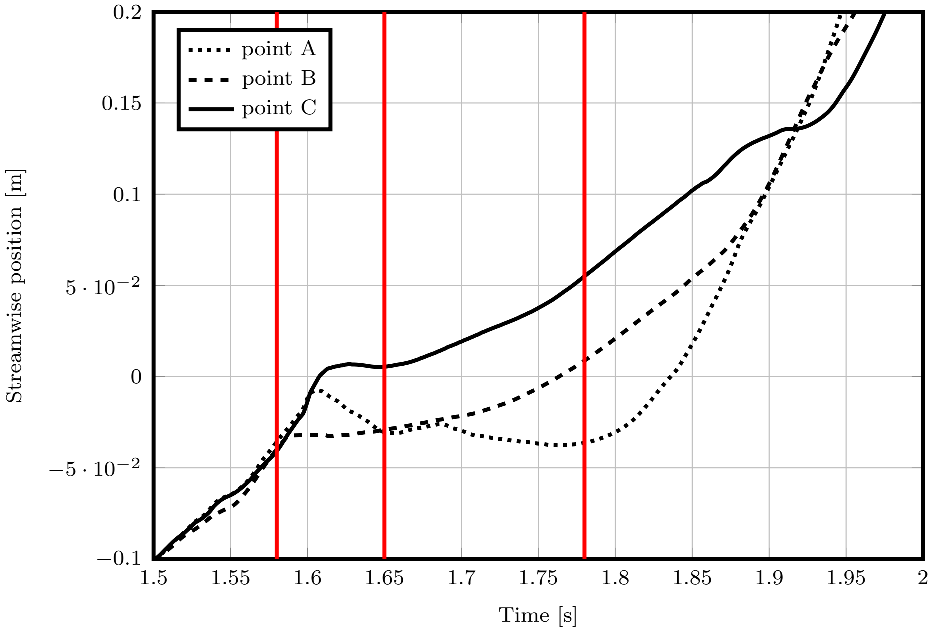


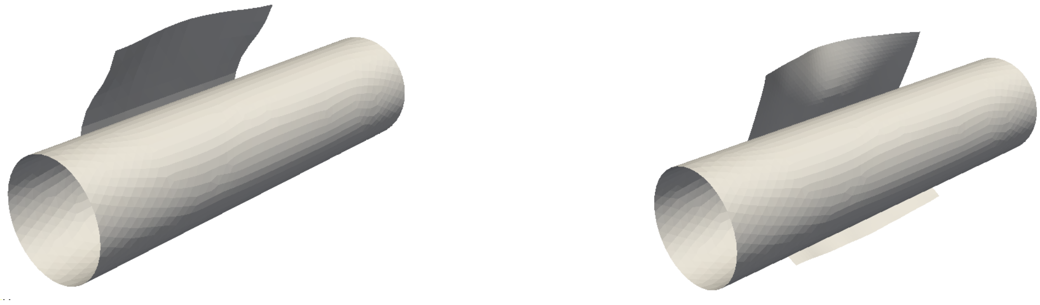
Figure 7.23: Time evolution of the rag streamwise displacement (points A B and C).

deformation of the rag when the latter is the closest to the cylinder, for two different grid resolutions. The coarse grid corresponds to a Lagrangian and an Eulerian spacing of $\Delta s = 0.008m$ while the spacing is $\Delta s = 0.002m$ for the fine grid. It appears that the deformation of the rag is very similar for both resolutions. The only noticeable difference in the shape is the higher folding of the upper and lower edges obtained with the fine grid. Regarding the collision process, the lower the resolution, the faster the release of the rag behind the obstacle is. This is merely due to the fact that, at low resolution, the collision is activated further away from the cylinder which results in higher fluid velocities. Finally, it was verified that the trajectory of the rag points A, B and C obtained with the three grid resolutions are showing a convergence behaviour for the rag displacement.

Figure 7.26 shows the deformation of rags during the collision process for different values of the bending and tension coefficients $(\zeta, \phi) = (B, T)$. Although the average path taken by the rag does not depend on its mechanical properties, noticeable differences are

(a) Coarse grid (b) Fine grid

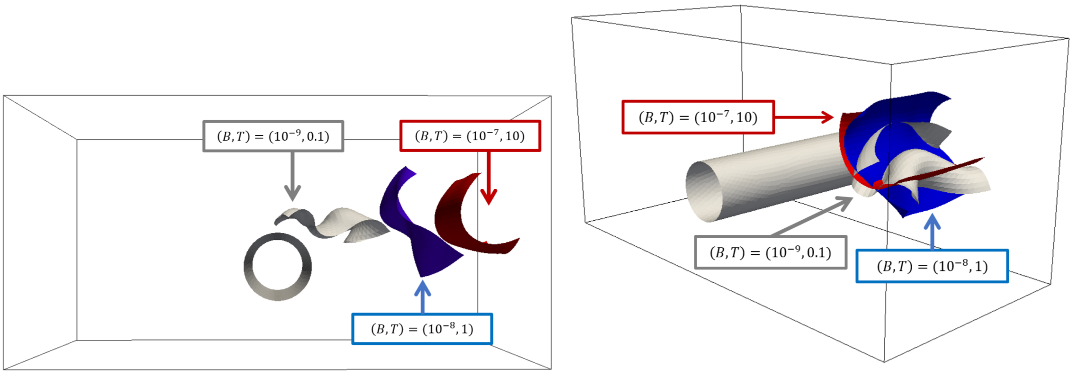
Figure 7.24: Front view of the rag colliding with the cylinder.



(a) Coarse grid (b) Fine grid

Figure 7.25: Side view of the rag colliding with the cylinder.

visible for the three values of the couple of coefficients. Two conclusions can be drawn from these results: (i) the collision with the cylinder is sufficient to trigger a folding of the rag, which is increased with lower coefficients as expected, and (ii) the deformation of the rag decelerates the collision mechanism. Hence, with lower values of the bending and tension forces, the inextensibility condition of the rag is not valid any more. This results in large stretching of the rag on both side of the cylinder as visible in Figure 7.27. In this configuration, the rag experiences large forces on both top and bottom edges, that give more balance to the rag around the cylinder and are thus slowing down the release of the flexible solid.



(a) At the same time $t = 0.2275s$

(b) At the same position

Figure 7.26: Influence of rag mechanical properties on the collision and transport processes.

(a) $(\zeta, \phi) = (10^{-6}, 100)$

(b) $(\zeta, \phi) = (10^{-8}, 1)$

Figure 7.27: Snapshot of the rag embracing the cylindrical obstacle for two values of the bending and tension coefficients.

Influence of friction

Assuming the existence of a direct solid - solid contact between the flexible slender structure and the rigid obstacle, accounting for a friction force can be of relevance. Simulations with the combination of the kinematic collision model and the friction model were performed for different values of the friction coefficients c_f. For these cases, the tension and bending coefficients are respectively fixed to 1 and 10^{-8}. The results are presented in Figure 7.28 and 7.29. As expected the friction force is decelerating the rag during its sliding against the cylinder wall, delaying its release. Above a threshold set by the friction coefficient, the friction force compensates for the acceleration force. This balance of forces prevents the release of the rag.

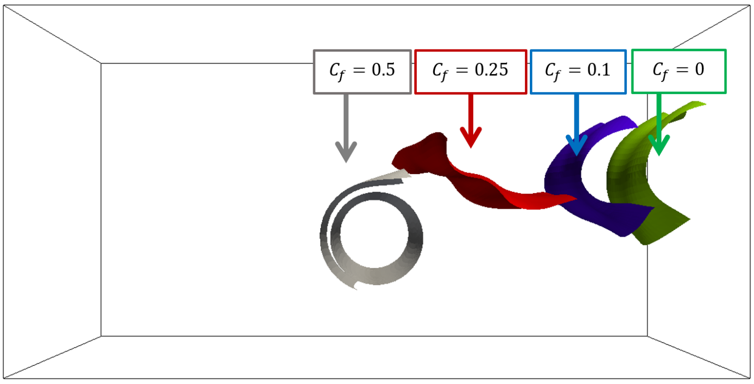


Figure 7.28: Snapshots of the rag at the same time for different friction coefficients.

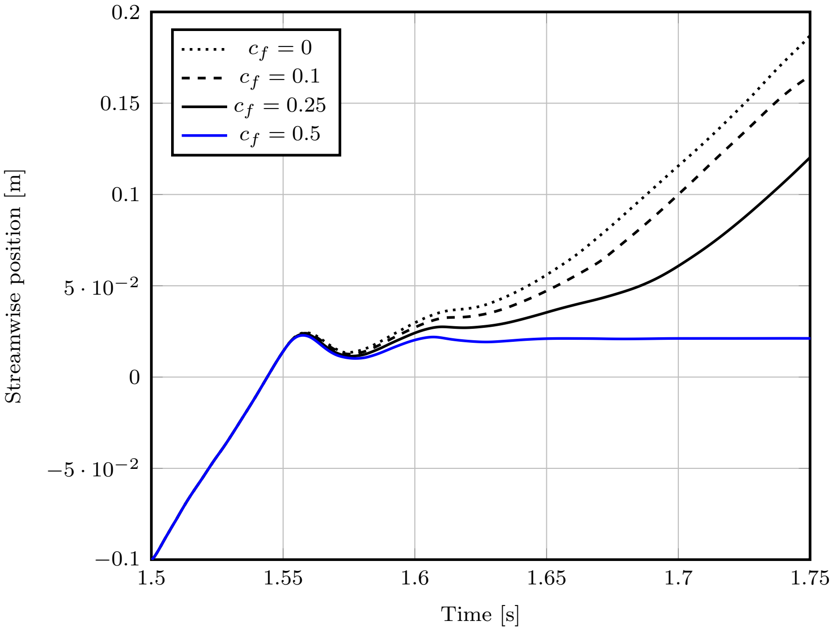
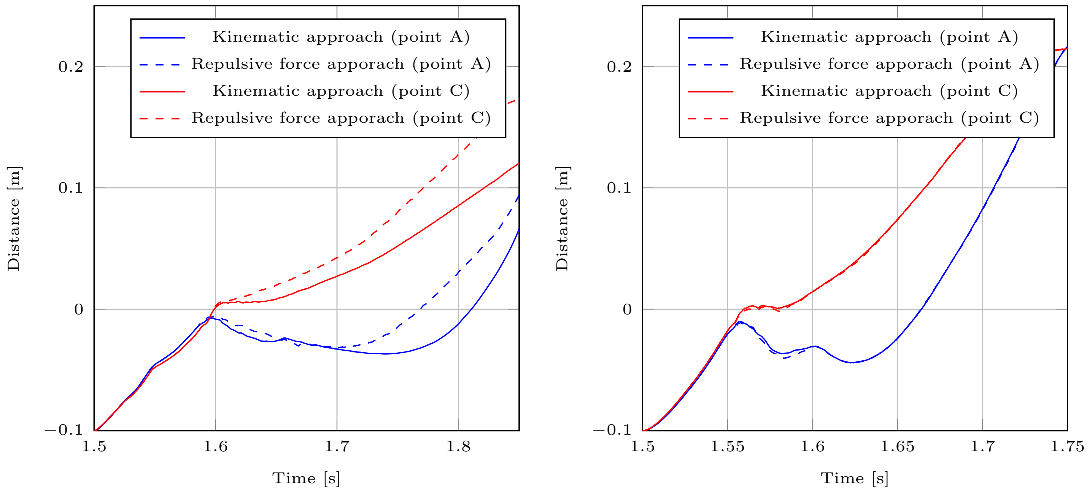


Figure 7.29: Streamwise displacement of point C for different friction coefficients c_f.

Comparison between the collision models

The rag's trajectories of the extremity points A and C are shown for the two collision models in Figure 7.30 and for the two fluid solvers (Lattice-Botlzmann in 7.30(a) and Navier-Stokes in 7.30(b)). A good similarity is found between the kinematic model and the force based model. The collision mechanism starts with a similar process as can be seen for the LB-IBM in Figure 7.31. With this method, the rag eventually shows a slightly more elastic behaviour with the kinematic approach (blue surface in the figure), which is slightly slowing down the release mechanism. The fact that this elastic rebound is not present with the repulsive force based model is due to its definition: the repulsive force acts as a force decelerating the rag proportionally to the acceleration provided by the IBM forcing. In the Navier-Stokes context, a perfect match is found between both collision models as highlighted in Figure 7.30(b). In this case, the kinematic approach does not generate a more elastic behaviour. However, the definition of the kinematic approach in the Navier-Stokes solver might lead to other issues due to the dependence of the correction on the local fluid velocity. If the rag manages to get close to the rigid body wall (due to a large time step or a small collision trigger h), the local fluid velocity can become small enough to produce a significant stretching of the flexible structure, then breaking the inextensibility criterion. This problem does not occur with the kinematic model implemented in the Lattice-Boltzmann context.

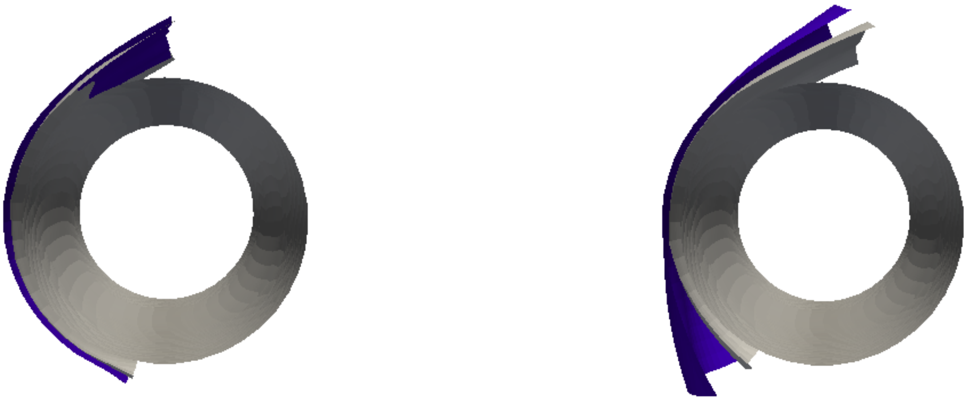
Regarding the force based model, it is important to note that this model is sensitive to the stiffness parameter k_w which is cancelling the external forces applied on the rag. Large values can lead to instabilities and low values do not prevent the rag from penetrating the cylinder. The proximity of the rag to the cylinder wall can be affected by the constant value and the collision process as well *a fortiori.* In this regard the force based model is less predictable than the kinematic model, for which the minimum distance is specified. Figure 7.32 illustrates the types of instabilities which can be encountered if the rag's flexibility is too high (left picture) or if the solid density is too low compared to the fluid density (right picture). The instabilities appear respectively in the form of very large deformations or spurious oscillations at the surface. A possible method



(a) LB-IBM & $\Delta\rho = 0.9$

(b) NS-IBM & $\Delta\rho = 0.1$

Figure 7.30: Comparison of the collision processes for the two types of collision model.

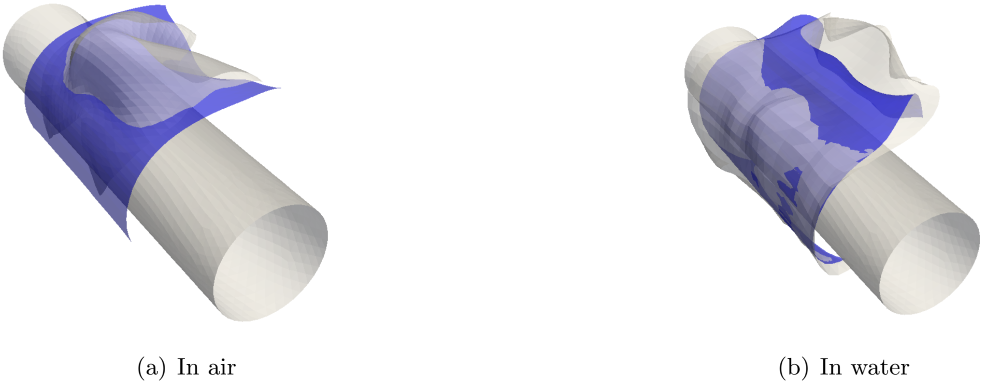


(a) $t = 0.1s$

(b) $t = 0.125s$

Figure 7.31: Comparison of the rag position for the two types of collision model with LB-IBM. The rag obtained with the Kinematic model is shown in blue, while the one obtained with the force model is shown in grey.

to avoid these oscillations could be to smooth the collision force, which is currently defined locally and which uses the minimum distance to the rigid body wall for each flexible solid points. This is another difference compared to the kinematic approach, where several points from the rigid solid may contribute to the correction of the flexible structure position. Finally, it is worth mentioning as well the last difference between both collision approaches: the kinematic method is a post-treatment method as the position of the flexible solid is corrected after its motion, while the force based method is a pre-treatment method as the repulsive force takes into account the information from the previous time step . The pre-treatment nature of the force based method can also impact the predictability of the distance where the collision will happen.


(a) In air

(b) In water

Figure 7.32: Instabilities arising with the repulsive force based collision model. The rag obtained with the Kinematic model is shown in blue, while the one obtained with the force model is shown in grey.

The agreement found between the two types of collision model is acceptable. This results is important as it justifies the use of the kinematic approach, which is not based on a physical correction, but was found to remain stable over a wider range of flows and mechanical properties.

Comparison between the fluid solvers

The present test case makes up an interesting case to compare also the two coupled solver LB-IBM and NS-IBM. Comparison between the two methods were already made for the dynamic response of a thin flexible structure (Case 2 in Section 7.3.2) or for the simple transport of a thin flexible structure (Case 3 in Section 7.3.3). Here, the focus is on the collision mechanism.

The trajectory of point C is shown in Figure 7.33 for both solvers. One can firstly note that the difference of times for the rag to reach the cylinder which is about 30 %. However, the rags mean speeds are very close as shown by the similar slopes obtained with the displacement curves. This result was already confirmed in Section 7.3.3, when comparing the transport of a rag for the two solvers. The difference in the physical times to reach the cylinder comes thus mostly from the initial acceleration process. The second point is the difference in the rag behaviour during the collision. The rebound of the rag shown by the backward displacement in Figure 7.33 for NS-IBM highlights an elastic behaviour with this solver, which is not obtained with LB-IBM. This can also be explained by the difference of magnitude of the IBM forcing between both solvers. Hence, a smaller IBM force means a more important dominance of the internal forces, which governs the rag's elasticity, and vice-versa. Furthermore, it can be seen that a lower density with NS-IBM is vanishing the elastic behaviour, as shown by the black dashed line in Figure 7.33. On the other hand, a higher density with LB-IBM does generate an elastic bounce after collision. The presence or the absence of this elastic bounce directly after the contact with the cylinder is illustrated in Figure 7.34.

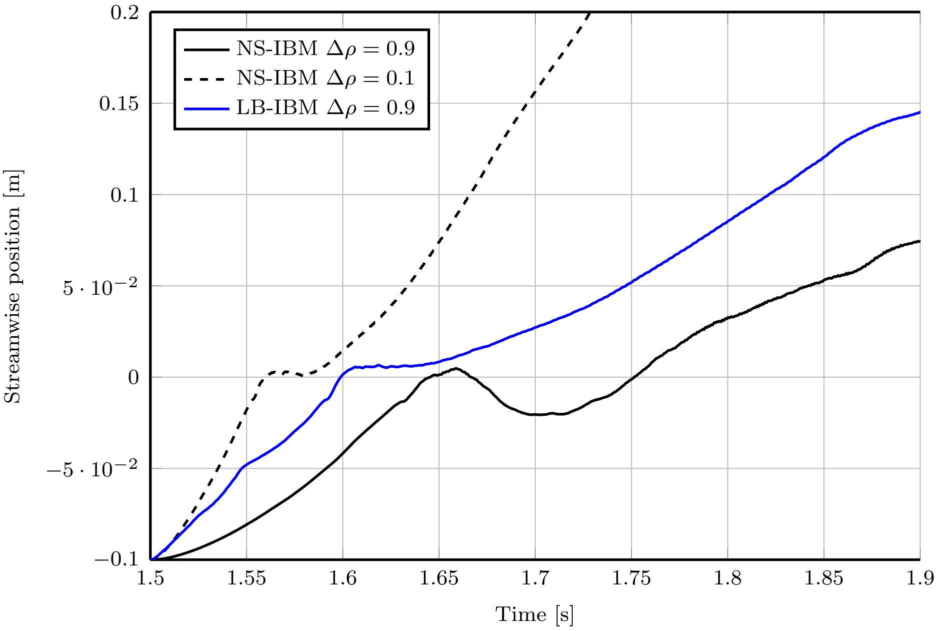


Figure 7.33: Comparison of NS-IBM and LB-IBM for the rag's transport and collision processes.

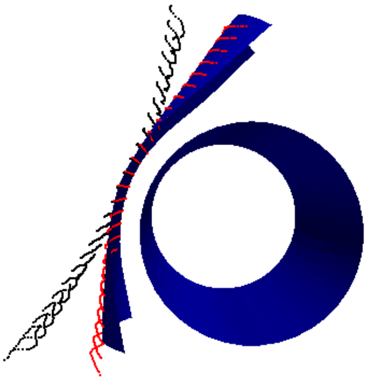


Figure 7.34: Illustration of the collision process for both fluid solvers. The blue surface denotes the rag computed with LB-IBM and $\Delta\rho = 0.9$. The black and red points represent the rags computed with NS-IBM respectively for $\Delta\rho = 0.9$ and $\Delta\rho = 0.1$

7.4 Conclusions

In this chapter, the formulation of a FSI model within a Lattice-Boltmzann solver for handling thin flexible structures was presented. The FSI model includes a solid solver based on the formulation of the deformation energy, which describes the dynamic of the structure. As one of the main aims of this project is to simulate the behaviour of rags in centrifugal pumps, particular attention was paid also to the modelling of the collision mechanism between the flexible solid and a rigid bluff body. Due to the scarcity of experiments dealing with these type of problems in literature, the FSI model developed by Huang et al. [56] has been implemented within a Navier-Stokes solver, for validation and comparison purposes. Both FSI models are based on an IBM in order to couple the solid equation with the fluid ones.

The LB-IBM produces results in reasonably good agreement with reference benchmark (see Case 2 in Section 7.3.2). Other cases (see Case 1 in Section 7.3.1 and Case 4 in Section 7.3.4) have shown the capabilities of this FSI model to generate deformations which are qualitatively consistent with experimental data and numerical results from literature. The model is also able to handle large deformations of relevance in engineering applications. From the analysis of the results, two shortcomings can be identified: (i) an issue regarding the stability of the solid solver for specific material properties, (ii) a potential variance in the accuracy of the results, which still needs to be quantified.

Regarding the first issue, the FSI model within the LB-IBM has shown its limitation in two cases: for materials with high stiffness and in the neutrally buoyant case. While very stiff materials are also problematic for the NS-IBM of Huang, this model can however handle materials with a density close to the fluid density. This difference of stability between both FSI models can be explained by the coupling mechanism. The penalty IBM of Huang [56] is indeed based on the use of a dual set of points for the modelling of the flexible structure: one set is solved with the solid equation, while the second one is following the streamlines. The coupling between both is performed with the momentum forcing, which can be regarded as the inertial force multiplied by a relaxation factor. This would suggest that the duality of the solid points enhance the stability of the overall FSI model. The same conclusion was already drawn for similar approaches as in [74] and in [133], also using twin sets of solid points but with a different coupling.

The difference of formulation for the momentum forcing in the NS-IBM [56] and in the LB-IBM [97] gives also different magnitudes for the action of the fluid. This discrepancy will influence both the mechanical behaviour of the slender structure and the transport of the latter. As seen in the test cases, the magnitude of the forcing obtained with the direct-forcing approach in LB-IBM is higher. Thereby, for a given set of parameters, the transport of the flexible solid is faster, while its behaviour appears less elastic. Case 4 in Section 7.3.4 which provides experimental data cannot help to conclude on which FSI model is the most accurate, as the case focuses on neutrally buoyant conditions, which cannot be achieved with LB-IBM. Additional experiment would be relevant to confirm this assertion. One could consider for instance a rag, heavier than the fluid, sinking in a tank or the same material rising in a vertical channel flow.

In regard to the collision modelling, a good agreement is found between the kinematic approach and the approach considering a repulsive forcing. As the kinematic model is stable in a wider range of applications, it is recommended to use the latter. Both approaches generate a realistic collision mechanism if one considers the presence of a fluid layer between the flexible and the rigid solids. For a quantitative validation of the numerical results, experiments dealing with collision processes must be carried out in future works. One can mention finally a similar limitation for both collision models, which lies in the fact that only local information is used to compute the repulsive force or the position enforcement. A more physical model could be based on the internal mechanical force rather than the action from the fluid. Due to the implicit treatment of the elastic force in the solid equation, such a model can apply a forcing locally taking into account the total stretching and bending effects in the flexible solid.

Chapter 8

Answers to the clogging issues and conclusions

8.1 Answers to the clogging issues

This first section has a dual aim:

- the comparison of the two FSI models developed with NS-IBM and LB-IBM for the transport of a rag in the single-blade pump.
- a preliminary study of the clogging mechanism with LB-IBM, which includes a sensitivity analysis to numerical and physical parameters.

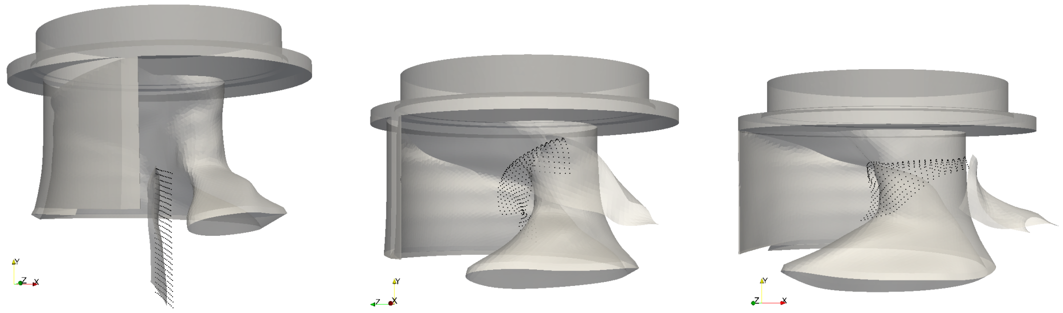
8.1.1 Comparison between both IBM-FSI models

The pump model is described in Section 5.1 and 6.6 for the sharp NS-IBM and the diffuse LB-IBM respectively. A flow rate of $55l/s$ is considered here, which corresponds to the BEP of single-blade pump. The diffuse LB-IBM is extended and coupled to a solid solver to model the rag (see previous chapter). In the Navier-Stokes framework, the diffuse NS-IBM of Huang [56] is used for the rag. The flexible solid is initially located below the impeller axis of rotation and vertically positioned. Its dimensions are $5cm \times 10cm$ and its width is aligned with the pump outlet pipe. The parameters of the two simulations are listed in Table 8.1. The mechanical properties have been non-dimensionalized using the impeller maximum radial velocity and the rag width. In this study, cylindrical coordinates are considered. The vertical (or axial) components correspond to the impeller axis of rotation, while the radial and tangential components are related to the impeller rotations.

Model	**Fluid grid size**	**Solid grid size**	ρ^*	ϕ^*	ζ^*	Δt
NS-IBM	$2mm$	$4mm$	$6 \cdot 10^{-4}$	20	$4 \cdot 10^{-7}$	$5 \cdot 10^{-5}$
LB-IBM	$4mm$	$4mm$	$6 \cdot 10^{-1}$	20	$4 \cdot 10^{-7}$	$5 \cdot 10^{-6}$

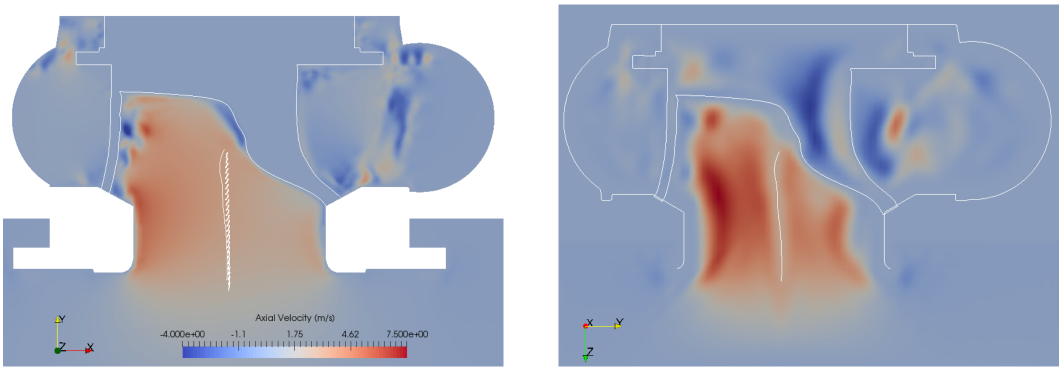
Table 8.1: Parameters for the simulation of a rag in the single-blade pump.

A good agreement in terms of velocity, trajectory and deformation is found between the two FSI models for the transport of the rag. Its position is shown at different times in Figure 8.1. In both cases, the rag is firstly rising along the axis of rotation at a similar speed, as it arrives at a similar position after $0.03s$. A difference of behaviour can be noticed, as the rag modelled with NS-IBM remains perfectly aligned with the axis of rotation, while the rag modelled with LB-IBM experiences small displacements in the transverse direction. The origin of this difference of behaviour lies in the difference of density. In NS-IBM, the flow is not affected at all by the presence of the rag, because a total neutrally buoyant condition is achieved. On the other hand, the FSI model within LB-IBM is only approaching this condition. This results in the rag generating a small blockage in the flow. As a consequence, its velocity is slightly lower than the average flow speed. These two configurations are shown in Figure 8.2. The rag modelled with LB-IBM is about 20 % slower that the average fluid velocity at the inlet, while its equivalent within NS-IBM is transported almost at the speed of the flow. Given that the two rags vertical velocities are similar with both models, it appears thus that the fluid velocity at the inlet is higher with LB-IBM than with NS-IBM. This difference is a result of both a coarser grid and the use of the bounce-back technique for the volute wall, which reduce the inlet pipe area. The combination of the transverse displacement of the rag modelled with LB-IBM and a slightly higher speed around the impeller eventually pushes the rag above the top part of the impeller leading edge before its Navier-Stokes counterpart. Despite this delay obtained with NS-IBM, the two trajectories and the rag's deformation are in good agreement as illustrated by Figure 8.1.



(a) $t = 0.03s$ (b) $t = 0.0625s$ (c) $t = 0.075ss$

Figure 8.1: Rag position during its passage through the pump. The rag modelled with LB-IBM is shown as a grey surface. The rag modelled with NS-IBM is shown as black dots.



(a) NS-IBM (b) LB-IBM

Figure 8.2: Vertical velocity in a cross-section of the pump cutting the center of rotation at $t = 0.03s$. The impeller interface is depicted by the white line for the two cases (as well as the interface of the volute and the cross-section of the rag for LB-IBM).

The difference in the inlet flow speed is negligible when a finer grid with a $2mm$ Lattice spacing is used for the LB-IBM. In this case however, the rag acceleration and velocity are significantly reduced, becoming also smaller than the one obtained with NS-IBM. This shows that neither the neutrally buoyant condition nor mesh independence are yet achieved with $\rho^* = 3 \cdot 10^{-1}$ and $4mm$ elements. Despite this issue, the results from the finer grid were showing that the rag followed the same path and experienced similar types of deformation.

8.1.2 Preliminary parametric sensitivity study for the occurrence of clogging

An important preliminary task in the physical study of the clogging mechanism is to eliminate or at least to quantify the influence of numerical parameters on the mechanism. By doing so, the study can then focus on the influence of physical parameters only (flow rate, rag's position, rag's mechanical properties, etc). The first numerical parameter to be investigated is the grid resolution. A grid convergence study is cumbersome in this pump case due the large mesh cell count involved. However, it was found that the mesh resolution has a significant impact on the initial acceleration and the final speed of the rag, mainly on the axial components. This can be due to two main reasons: (i) the LB-IBM is a first-order method only and (ii) perfectly neutrally buoyant conditions could not be achieved with the solid solver and the gravity force is also influencing the velocity of the flexible structure. Regarding solid-solid interactions in particular, the collision distance between the rag and the impeller is also a numerical parameter whose influence needs to be quantified. The low influence of the collision distance is demonstrated in Appendix D

Diverse clogging scenarios were investigated by varying the relevant physical parameters of the simulation: the dimension of the rag, the mechanical properties of the rag, the initial position of the rag in relation to the impeller center of rotation, and the flow rate. Five different initial positions were chosen as illustrated in Figure 8.3 for the same height. Another meaningful parameter which was not investigated in this research could be the initial axial position of the rag with respect to a fixed initial angle of the impeller. Varying the initial axial position of the rag is similar than varying the initial angle of the impeller. As previously, the non-dimensional density is set to $\rho^* = 6 \cdot 10^{-1}$. The mechanical properties have been non-dimensionalized using the impeller maximum radial velocity and the rag width. Regarding the numerical parameters, clogging simulations were performed with coarse ($4mm$ fluid and solid elements) and fine ($2mm$ fluid and solid elements) resolution. The same clogging scenario simulated with two different resolutions may lead to different conclusions as the velocity of the rag is different in each case, as highlighted previously. However similar trends have been found for both coarse and fine simulations, regarding the influence of the physical parameters considered here.

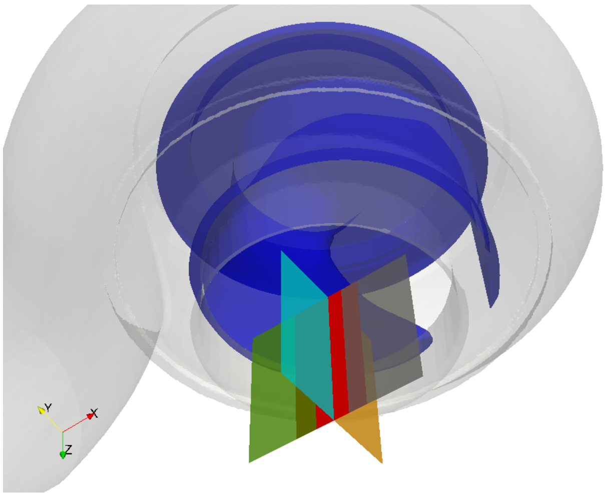


Figure 8.3: Initial position of the different $5cm \times 10cm$ rags used for the clogging scenario simulations. Position A in red, position B in green, position C in grey, position D in cyan, position E in orange.

The results of the different clogging studies are listed in Table 8.2 and Table 8.3. A real clogging case cannot be exactly reproduced by the IBM-FSI model presented here. However, in specific configurations the flexible solid eventually wraps itself around the leading edge, which is known from experience to lead to a physical clogging event. Interesting insights can be drawn from the parametric study. It appears firstly that the flow rate is the main parameter affecting the occurrence of clogging, as no configurations were found where the rag wraped itself around the impeller leading edge at $Q = 55l/s$ for more than half a cycle. Similar observations on the influence of the flow rate were made experimentally [23]. The fluid incident angle relative to the impeller leading edge is changing with the flow rate [48]. Increasing the flow generates an area of increased radial velocity and decreased tangential velocity, as was observed also in [37]. This helps the rag to move away from the impeller. Large separations observed with the singe-blade impeller at high flow rate (see Section 5.1) can also be an additional factor limiting the blockage. One can note however that experiments carried out in [23] have shown that rags are more likely to wrap around the upper part of the leading edge, while in most of the present simulations the wrapping occurred at the bottom part of the leading edge.

Finally, the length of the rag also increases the likelihood of clogging. For two configurations without onset of clogging with $5cm \times 10cm$, it was found that extending the length of the rag to $15cm$ led to the wrapping of the latter around the leading edge. The change in initial position has also an effect on the solution. It is however difficult to conclude on the predictability of clogging regarding this parameter, given the small number of simulations. On the contrary, the rag's flexibility does not seem to influence significantly the behaviour of the slender solid in the pump.

Rag	**Flow rate**	**Wrapping ?**	**How long ?**	**Comments**
$5cm \times 10cm$ Position A	$20l/s$	yes	$>$ 1 cycle	
$5cm \times 10cm$ Position A	$55l/s$	no	-	directly to the eye
$5cm \times 10cm$ Position B	$20l/s$	yes	1/2 cycle	
$5cm \times 10cm$ Position B	$55l/s$	no	-	directly to the eye

Table 8.2: Parameters and results of the different clogging scenarios with coarse grids ($4mm$ elements). LE stands for leading edge. The values of the bending and tension coefficient are chosen as $(\phi^*, \zeta^*) = (200, 4 \cdot 10^{-5})$. The eye defines the central region below the impeller upper plate.

Rag	**Flow rate**	**Wrapping ?**	**How long ?**	**Comments**
$5cm \times 10cm$ Position A	$20l/s$	no	-	below LE pushed to center
$5cm \times 10cm$ Position B	$20l/s$	no	-	below LE pushed to center
$5cm \times 15cm$ Position B	$20l/s$	yes	1 cycle	
$5cm \times 15cm$ Position B	$55l/s$	no	-	below LE pushed to center
$5cm \times 10cm$ Position C	$20l/s$	no	-	directly to the eye "bounce" on LE
$5cm \times 10cm$ Position C flexible	$20l/s$	no	-	directly to the eye "bounce" on LE
$5cm \times 15cm$ Position C	$20l/s$	yes	$>$ 1 cycle	
$5cm \times 15cm$ Position C flexible	$20l/s$	yes	$<$ 1 cycle	
$5cm \times 15cm$ Position C	$55l/s$	slightly	1/2 cycle	around top part of LE
$5cm \times 10cm$ Position D	$20l/s$	no	-	directly to the eye
$5cm \times 10cm$ Position E	$20l/s$	slightly	1/2 cycle	around top part of LE
$5cm \times 10cm$ Position E	$55l/s$	slightly	$<$ 1/2 cycle	around top part of LE

Table 8.3: Parameters and results of the different clogging scenarios with fine grids ($2mm$ elements). The values of the bending and tension coefficient are chosen as $(\phi^*, \zeta^*) = (200, 4 \cdot 10^{-5})$ for the standard cases, and as $(\phi^*, \zeta^*) = (20, 4 \cdot 10^{-6})$ for flexible scenarios.

8.2 General Conclusions

8.2.1 Conclusions and contributions

This last section summarizes conclusions on the specific Immersed Boundary Methods considered in this research and their application to Fluid-Structure Interaction modelling, with a particular focus on waste water pump design and clogging issues. The conclusion is divided into two parts: (i) validation and assessment of IBMs for single-phase FSI problems, and (ii) outcomes of diffuse IBMs for fluid - flexible solid interactions modelling.

A new versatile sharp interface type of IBM was developed within a Navier-Stokes solver and validated against laminar and turbulent cases involving fixed and moving boundaries. This sharp NS-IBM is a continuous IBM, based on a Penalty approach, where the momentum forcing is applied before the discretization. The method includes reconstruction schemes for the physical variables at the immersed interface to enforce more accurately the boundary conditions, and a coupling with the DES turbulence model with consistent definition of the turbulent variables at the same interface. A diffuse interface type of IBM implemented within a Lattice-Boltmzann solver was assessed and compared to the sharp NS-IBM for similar applications. The conclusions from these numerical investigations are presented below:

- The correction schemes applied to the velocity and the pressure to impose, respectively, a Dirichlet and a Neumann boundary condition at the immersed interface to achieve a no-slip condition, have shown to improve the order of accuracy of the solution. These improvements are visible for both local variables near the interface and integrated quantities. The order of accuracy of the improved NS-IBM, including the correction schemes, is at least 1.5, while a simple NS-IBM without correction achieves only first order.
- Different reconstruction schemes were investigated for the correction of velocity and pressure:

- Regarding the velocity correction, very little difference was found from a comparison between the types of schemes proposed and considered in this research (explicit or implicit formulation, based on interpolation or based on gradient reconstruction). However, it was found that imposing the velocity condition either internally or externally did have an impact on the solution. Both types of schemes have shown to improve the velocity profiles locally and the estimation of integrated quantities. Shortcomings were however identified. An external velocity correction scheme tends to thicken the size of the immersed object, due to possible discrepancies between the reconstructed linear profile and the theoretical profile of a laminar boundary layer. An internal velocity correction scheme deteriorates locally the pressure solution near the immersed interface due to the mirrored velocities from the extrapolation.

- Regarding the pressure correction, the decoupling scheme has been shown not to impact the solution. On the other hand, imposing a zero gradient for pressure through a penalization approach leads to significant improvements in the solution for both velocity and pressure. These improvements are more significant with the internal velocity correction scheme, as the pressure is deteriorated with this specific scheme. The combination of both velocity and pressure penalizations has been shown to provide results in very good agreement with equivalent body-fitted simulations for the laminar cases considered in this research. Furthermore, it was shown that the treatment of the pressure boundary condition at the immersed interface, which is usually not handled for continuous IBM, is improving the accuracy of the results. A shortcoming was identified for this explicit pressure penalization scheme: spurious pressure time fluctuations and an increase of the harmonics amplitude appear upstream of the immersed solid when standard wall boundaries are in the vicinity of the immersed boundary. However, these fluctuations are not strongly affecting the time-averaged pressure solution, and results suggest that their influence on the pressure distribution over the surface of the immersed solid is negligible. These spurious fluctuations were removed with an implicit pressure penalization scheme.

- The conclusions on the assessment of NS-IBM for turbulent rotating flows are the following:

  - Spurious pressure fluctuations from the explicit penalization scheme were found to be negligible for the mixer case, but were observed for the two pump cases, and especially for the single-blade pump due to the close proximity of wall boundaries (volute wall). The implementation of the implicit penalization scheme was however not successful for these engineering cases.
  - The IBM formulation of the Spalart-Allmaras turbulence model (pSA), based on a penalization of the turbulent viscosity, is necessary in order to obtain meaningful results. For instance, the pSA method leads to an improvement of respectively 20% and 50% in head and torque estimates for the single-blade pump (with a flow rate of $40l/s$), in comparison to the standard SA formulation. A linear reconstruction scheme of the turbulent viscosity gives additional improvements of 6% and 1% in comparison to the simple pSA for the same case.
  - The typically poor resolution of the boundary layer with IBM was found to enhance the mixing, in comparison to body-fitted simulations with boundary layer resolution. However, this has a negligible influence on integrated quantities, such as the force and the torque on the immersed solid, as well as time and phase averaged variables in the turbulent cases considered in this research. This argument is supported by the investigations made on the influence of the inflation layer on body-conforming simulations for this type of application. The velocity correction schemes within NS-IBM have been shown to reduce the overestimation of the mixing. A correction scheme based on a power-law model, which is more consistent with the logarithmic profile of an attached turbulent boundary layer, was tested for the mixer case. Although reducing

the turbulent intensity, this reconstruction scheme did not provide better predictions of averaged or instantaneous flow variables.

  - For the single-blade pump, the head estimates obtained with NS-IBM were found to be as good as body-fitted results with equivalent mesh resolution, except for the highest flow rates ($Q = 70l/s$). In this case, the use of a dynamic mesh is significantly improving the solution. For the two-blade pump, the maximum relative difference between the head and the efficiency obtained with the simple NS-IBM and experimental data is about 13% and 33% respectively (for a mesh using $5mm$ cells). The corrections schemes are capable of improving these estimates.

- The diffuse interface type of IBM has been shown to overestimate instantaneous variables, such as the pressure stress over the immersed solid, the maxima of velocity as well as the intensity of mixing. Despite these discrepancies, this method was shown to be able to give predictions of averaged quantities, similar to NS-IBM in terms of accuracy, at equivalent maximum resolution.

To conclude on the application of IBMs to pump design, the outcomes of this thesis shows that both sharp and diffuse types of approaches are suitable to accurately predict the quantities of interest to hydraulic design, where accuracy within 10% is appropriate. In specific configurations, involving large flow separation, adaptive meshing may be required to achieve similar level of precision than standard body-fitted simulations.

The diffuse LB-IBM used for single-phase flow applications was extended and coupled to a dynamic solid solver to model the interactions between fluids and two-dimensional flexible structures. A diffuse NS-IBM based on Huang's model [56] was assessed and compared to the LB-IBM for laminar and turbulent cases dealing with the transport and the deformation of flags/rags. Comparisons with experiments were also performed. Collision models were implemented and assessed to handle possible interactions between the flexible structure and rigid solids, within both approaches. The two IBMs were finally applied to the study of rag transport in a waste water pump and clogging scenarios were investigated with the LB-IBM. The conclusions for this part of the research can be summarized as follows:

- The coupled LB-IBM FSI model has been shown to be able to correctly reproduce the behaviour of slender flexible solids in fluids, when under the combined influence of elastic internal forces, gravity and the action of the fluid. A quantitative comparison against a benchmark case have also shown that correct agreement with previously published results can be achieved.

- When compared to the FSI model within NS-IBM, a significant difference was found regarding the magnitude of the IBM momentum forcing. The IBM force computed with LB-IBM, and based on iterative direct-forcing, was shown to be considerably higher than its NS counterpart based on penalization, when the position of the flexible structure is constrained. Despite this difference, the results from LB-IBM and NS-IBM are in good agreement regarding the deformation and the transport of the structure, in similar conditions. This outcome supports the correctness of both FSI models.

- An advantage of direct-forcing approach (used in LB-IBM) in comparison to the penalty approach (used in NS-IBM) is the absence of user-defined parameter like the penalization coefficient. However, the definition and adjustment of the hydrodynamic thickness of the slender solid within the LB-IBM coupling counterbalances this advantage.

- The difference of magnitude of the IBM forcing term explains the difference of stability between the two solid solvers under neutrally buoyant conditions. The NS-IBM is capable of achieving perfectly stable solutions under such conditions, while the solid solver within LB-IBM becomes unstable when the solid density is significantly decreased.

- The results of the two types of collision models (kinematic and force based) are providing similar dynamic simulations. The kinematic approach has been found to remain stable over a wider range of flow conditions and mechanical properties.

- Regarding the transport of a rag in the centrifugal waste water single-blade pump, the main conclusions are the following:
  - Although the two IBMs cannot realistically reproduce the typical blockage of waste water pump, both models predict rags wrapping around the impeller leading edge in specific conditions, which is known to lead to a physical clogging event. The failure to reproduce such an event is probably due to the approximate collision and friction models, combination of which is not capable of modelling real structure-wall friction.
  - Similar results were obtained with the two types of IBM for the rag path and behaviour in the pump. However, differences were found regarding the rag velocity, because of (i) the different coupling methods and mainly because of (ii) the fact that neutrally buoyant conditions are not achieved with LB-IBM.
  - Mesh independency is not achieved for the configuration considered here. However, similar conclusions were drawn for both resolutions regarding the influence of specific physical parameters. The main parameters prone to increase the occurrence of clogging are the flow rate and the rag length.

8.2.2 Future work

As stated in the previous section, when applied to turbulent rotating flows, IBMs appear to break the uniformity of the flow in specific areas and to increase the velocity fluctuations in space. These effects are more noticeable when no correction is used for the physical variables at the immersed interface within sharp IBMs, and they are again more noticeable for diffuse IBMs in comparison to sharp IBMs. Although these shortcomings do not deteriorate integrated quantities, their influence on the transport of a secondary component should be assessed. This influence is expected to be small for a relatively stiff material, but it could have a non-negligible influence when the flexibility of the slender solid is increased or when its density is decreased. A way to quantify the influence of the enhanced mixing would be to compare the behaviour of rags when rotating equipments (mixer or pump) are modelled by IBMs or by standard body-fitted techniques. The remainder of the recommendations for future work is divided into two parts, in accordance with the two main objectives of this research.

(i) The future works advised for IBMs applied to three-dimensional rigid bodies with prescribed velocity are the following:

- Two tasks are identified regarding the pressure correction through a penalization scheme:
  - A solution to overcome the stability issue encountered with the implicit penalization scheme when applied to 3D complex geometries could be to implement a symmetrical (and artificial) Neumann boundary condition in the internal solid domain.
  - In most of the cases considered in this research, the Neumann zero pressure gradient condition is consistent with the projection of the Navier-Stokes equation on the immersed interface, because fixed or rotating boundaries are involved. However, this projection leads to $\frac{\partial p}{\partial \mathbf{n}}|_{\Gamma} = \rho \frac{D\mathbf{u}}{Dt}|_{\Gamma} \cdot \mathbf{n}$, where the normal acceleration is not necessarily zero for instance with the oscillating cylinder case. For a more generic version of the model, such a pressure boundary condition should be implemented. Although the pressure penalization scheme imposing a zero gradient has shown to improve the results in the oscillating cylinder case, discrepancies are still visible in comparison to the body-fitted results, and their presence can be explained by not taking into account the normal projection of the acceleration term.
- Regarding turbulence modelling, more accurate wall models could be considered for the reconstruction of the velocity with a relatively coarse grid. For instance, models based on the Turbulent Boundary Layer (TBL) equation are usually performing

better than the power-law model [19], which was implemented in this research. The main difficulty in this case is to estimate the shear stress at the immersed wall.

- Body-fitted simulations for the two-blade impeller pump would provide interesting insight, particularly regarding the improvements due to the velocity and pressure correction schemes onto the torque predictions.
- For a complete comparison of IBMs and the two types of fluid solver, it would be interesting to propose a diffuse IBM within the Navier-Stokes solver, and a sharp IBM within the Lattice-Boltzmann. The comparison of the outcomes of these additional approaches would help to better distinguish the differences coming from the modelling of the interface (sharp or diffuse) with the differences lying in the solution of the fluid equations.

(ii) The future work recommended for diffuse IBMs applied to fluid - flexible structure interaction can be divided as follow:

- A complete characterization of the mechanical properties of typical resilient solids found in waste water needs to be performed. This would allow to choose relevant tension and bending coefficients for the solid solver, and to set the non-dimensional density, which is showing to what extent the neutrally buoyant condition is achieved, if it is. In addition, this characterization would help define the most accurate model between the two types of coupling, penalization in NS-IBM or iterative direct-forcing in LB-IBM.
- The low tension and bending force necessary to reproduce realistic large deformation may break the inextensibility criterion for the current dynamic solver. In this case, the criterion must be implemented implicitly in the solver as an additional constraint to the solid equation, as is proposed in [55] and [35].
- The main limitation of the current FSI models (for both NS-IBM and LB-IBM) lies in the fluid-solid coupling, which is obtained from the IBM forcing term. A more

accurate model would be to estimate the action of the fluid on the flexible solid from the integration of pressure and viscous stress on the immersed surface. This type of approach would also significantly reduce the differences between NS-IBM and LB-IBM.

- Investigations on the interactions between rigid and flexible solids immersed in fluids are relatively recent and scarce. In this context, experimental characterization of the collision between a rag and a bluff body in a flow for instance, would allow a quantitative validation of the collision models implemented during this research. Regarding numerical models, both kinematic and force based approaches are only using local information in order to correct the rag's position. This limitation can be overcome by deriving a repulsive force from the internal mechanical force. As this force is handled semi-implicitly in the solid solver, such a model would account for the global stretching and bending effects of the flexible structure.
- Additional investigations on the clogging scenarios with the single-blade pump are necessary to fully understand the mechanism leading to blockage. The use of friction or suction models could lead to more realistic blockage. However, these models would also need to be quantitatively validated.

Appendix A

Assessment of xIBM for multiphase flows

The influence of high swirl mixing on the instantaneous position of a gas bubble in a mixer has been studied. The geometry of the mixer considered here is similar to the one described in Section 5.3. The same procedure is used for the body-fitted and IBM grids. The average cell size was however fixed to $3mm$ in the core of the domain. For the body-fitted grid, three levels of refinement were then achieved by dividing cells into uniform octants, as illustrated in Figure A.1. This meshing led to a minimum cell size around $0.4mm$ at the impeller wall. For the IBM case, two grids were considered: (i) the same resolution as the body-fitted case and (ii) a mesh resolution without the last level of refinement (coarser mesh). The results are not compared to experimental data due to the scarcity of this type of experiments in the literature.

The gas phase is air, while the carrier phase is water. A Volume of Fluid (VOF) solver (interFOAM) is used to compute the incompressible fluid equations for these two phases. The interface capturing advection equation allows one to track the transport of the secondary phase within the primary one. The IBM is used to model the presence of the impeller. Similar to a single-phase solver, this is achieved with a forcing term in the momentum equation of the considered phase. No corrections for pressure or velocity

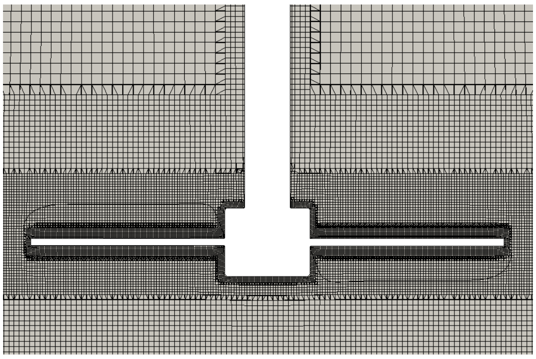
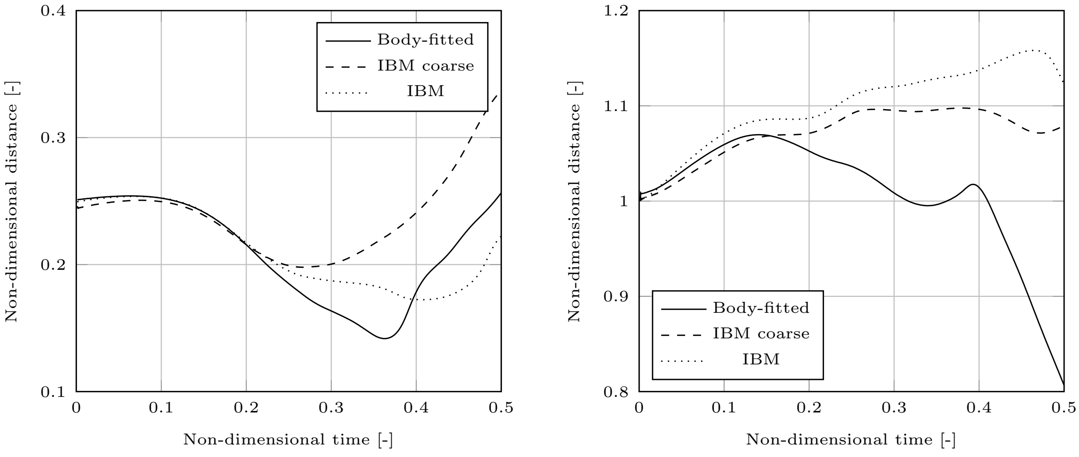


Figure A.1: Cross section of the body-fitted mesh around the impeller

near the IB interface have been considered initially. The penalization coefficient is as usual the subject of a convergence study to confirm independence of the results with regards to the latter coefficient.

The air bubble is released close to the lower surface of the mixer. The evolution of its center of gravity has been examined. Figure A.2 displays the results obtained with both IBM grids compared to the body-fitted approach. The time is non-dimensionalized by the rotational speed of the impeller and covers half a revolution. In addition, the distance traveled by the bubble is non-dimensionalized by the diameter of the mixer and the initial vertical position of the bubble for the radial and the axial distance respectively. From these trajectories, it appears that the IBM gives a fairly good reproduction of the impeller radial force. The last layer of refinement shows in addition an improvement in the estimation of the radial trajectory. However, a significant difference is noticed when comparing the axial component of bubble's trajectory. The coarse mesh fails to reproduce the exact downward force which compensates the buoyancy force and pushes the bubble towards the bottom of the tank. This is mainly due to the large thrust force generated around the blade with the body-fitted approach. Although under-estimated, the finer IBM grid seems to better evaluate this force, as visible in Figure A.2(b) when the bubble starts to move downward again. During the beginning of the computation, the underestimation of the impeller thrust force allows the dominance of the buoyancy force which explains the high position of the bubble.



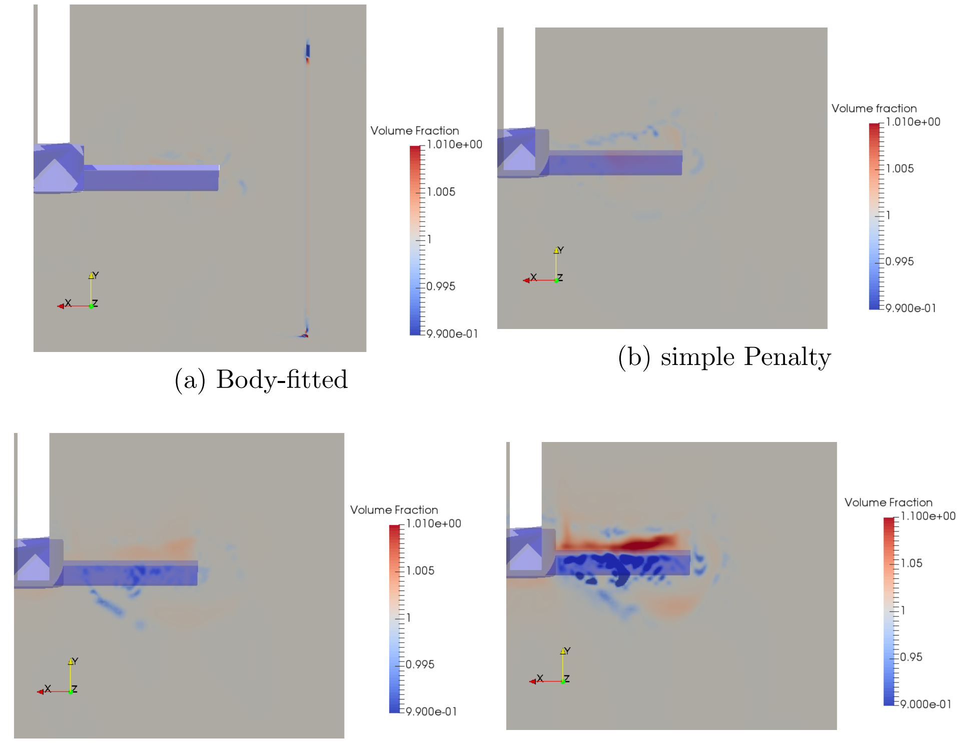
(a) Radial position

(b) Axial position

Figure A.2: Instantaneous evolution of the bubble position inside the mixer for body fitted and immersed boundary solutions

Figure A.4 shows the bubble interface after half a revolution of the impeller. The IBM grid based on the same resolution as the body-fitted grid allows to capture the bubble separation in two distinct parts. Although more elongated the bubble shape is in fairly good agreement compared to the reference results from the body-fitted simulation. On the contrary, the IBM on the coarse grid fails to capture the bubble separation. Regarding the thrust force, represented by the downward velocity vectors visible above the impeller, it appears also that the last grid refinement level improves the results for the IBM.

The influence of the IBM on the transport of the volume fraction is also investigated. Figure A.3 shows the contours of the volume fraction α for the body-fitted case and different penalty approaches on a section of the mixer lying in the fluid domain (far from the bubble). In the ideal case, the volume fraction should be equal to the fraction of fluid ($\alpha = 1$) everywhere on the section. However, one can see that this is not achieved for any of the numerical methods. The simple penalty approach and the penalty with outer velocity correction lead to deviation of the volume fraction of the order of 1% near the impeller. The latter method exhibits a slightly higher and broader deviation than the former one, due to the possible inconsistency between the linear correction scheme and the mass conservation. The mirrored velocities obtained with the inner velocity correction are as expected even more inconsistent with the mass conservation equation. In this case, the deviation from the ideal value exceed 10%. It is interesting to note that the standard body-fitted approach is also leading to mass conservation issues. Volume fractions above and below 1.0 are visible near the impeller and especially in the close vicinity of the interface between the moving and the fixed meshing parts of the GGI method, where deviations exceed 1% (higher than IBM).



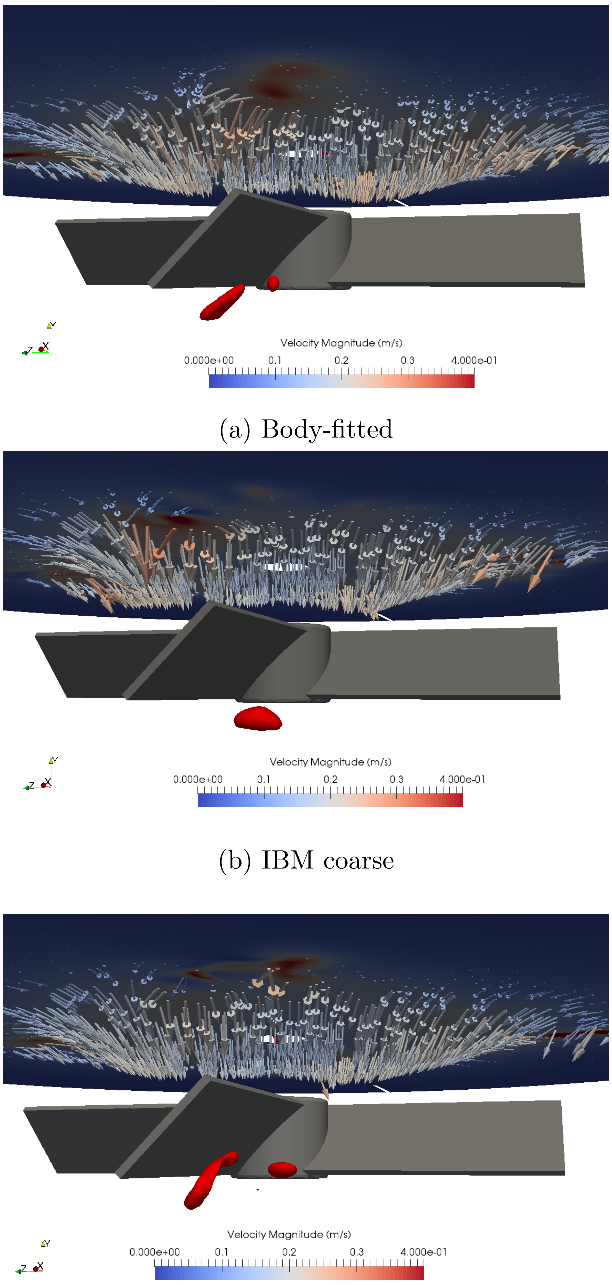
(a) Body-fitted

(b) simple Penalty

(c) Penalty with outer velocity correction

(d) Penalty with inner velocity correction

Figure A.3: Contour of α for different numerical approaches.



(a) Body-fitted

(b) IBM coarse

(c) IBM

Figure A.4: Contour of the air-water interface (in red) after half a revolution of the impeller. The velocity vectors are displayed on the horizontal cross section above the impeller.

Appendix B

On the modelling of insect flight

This section presents the investigation carried out with a 2D model of a symmetric flapping insect flight. The model is based on the previous work of Ota et al. [97], who developed a coupled LB-IBM identical to the numerical method presented in Section 6.2. While in [97] the insect wings are defined as a rigid 1D filament, in the current approach the solver for the solid dynamics introduced in Section 7.2.1 is coupled to the LB-IBM in order to model the deformation of the wings.

The influence of the wings flexibility on the flight mechanics, and in particular the generation of lift, has already been studied for a similar 2D model by De Rosis et al. [27] and for 3D models as well [95], [134]. When the structural deformation is not prescribed, unlike in [129] for instance, most of the studies employ Finite Element Methods (FEM) to solve the structural dynamics as opposed to the Finite Difference Method considered in the current study. The next section describes the 2D flapping wing model and shows how the wing flexibility is taken into account within the model. The last section summarizes the main results obtained with this model implemented in the Lattice-Botlzmann solver.

B.1 A 2D symmetric flexible flapping wing model

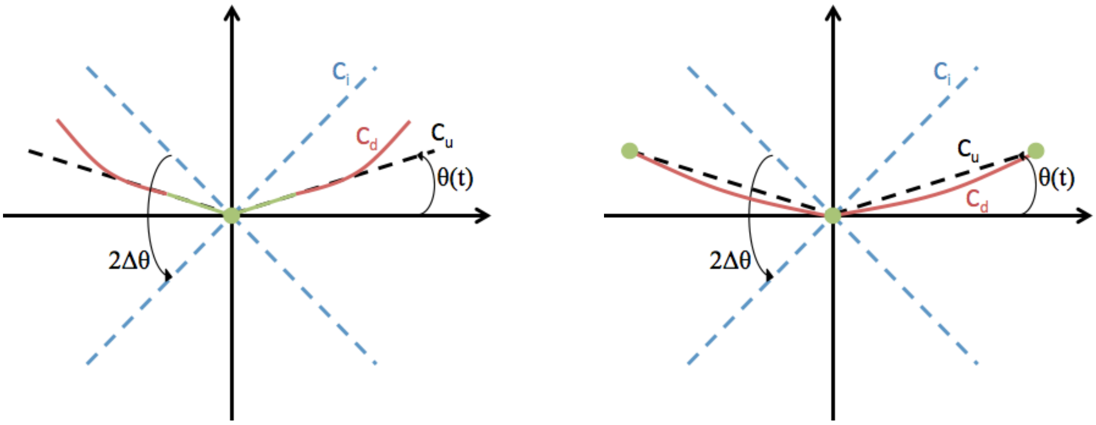
The insect is made of the two wings of length L connected to a hinge of mass M. The wings flap symmetrically with respect to the vertical and horizontal axis. For rigid wings, the equation of their angular position reads:

$$\theta(t) = \Delta\theta cos(\frac{2\pi}{T}t + \frac{\phi}{180}\pi) \tag{B.1}$$

where $\Delta\theta$, T and ϕ are respectively the amplitude, the period and the initial phase of the flapping oscillations.

Particular attention should be paid to the combination of the wing rigid body motion and its deformation. In this context, two different models were considered for the treatment of the deformation of the wings. Those two models are illustrated in Figure B.1. In Model A, a specific number of points along each wing are following a rigid motion, without any deformation taken into account. For all the remaining points, Equation 7.1 is solved for the structural dynamics of the wings. Regarding the boundary conditions, the free end condition is used at the wing extremities. At the interface between the rigid and flexible parts of the wing, the simply-supported condition introduced in Section 7.2.1 is however not suitable in this case, as it allows rotation. A clamped condition is then implemented, which satisfies $\mathbf{X} = constant$ and $\frac{\partial \mathbf{X}}{\partial s} = (cos\theta, sin\theta)$ and $\frac{\partial \mathbf{X}}{\partial s} = (-cos\theta, sin\theta)$ for the right and left wings respectively. In Model B on the other hand, only the hinge and the extremity points of each wing are governed by Equation B.1, while for all the other points, the wings deformation is solved thanks to the solid solver. In this case, the simply-supported condition can be used for all boundary points.

A decoupling behaviour is observed between the rigid motion zone and the deformable zone of the wings with Model A. Three possible sources for this issue were identified: (i) an error in the implementation of the clamped boundary condition, (ii) the necessity to use non zero initial tension through the wing and (iii) the necessity to use large values of the tension coefficient, making it impossible to obtain stable solutions of the solid model. It was confirmed that the use of non-homogeneous tension and bending coefficients to strengthen the connection between the two zones are not sufficient either to achieve the deformations shown in [134] for instance. As for Model B, deformations



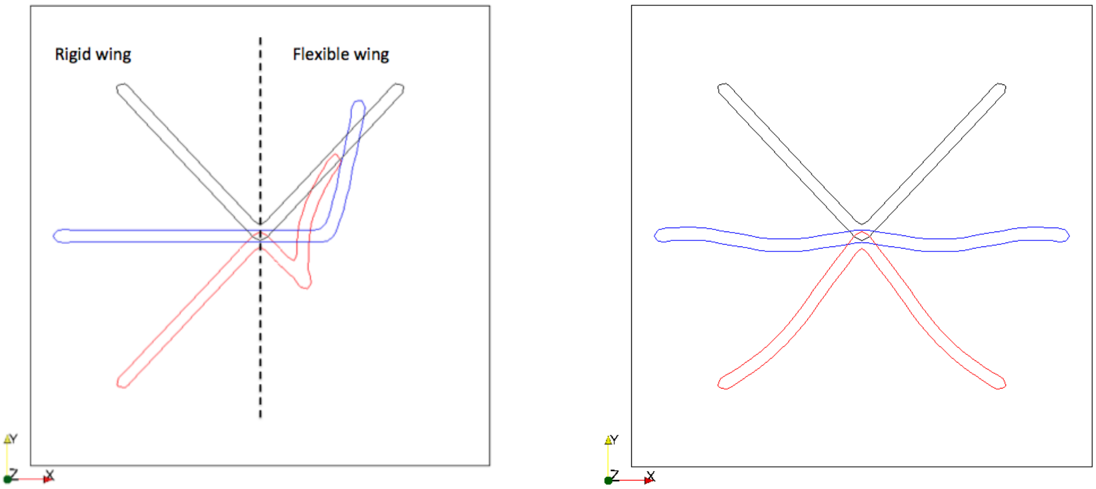
(a) Model A

(b) Model B

Figure B.1: Sketch of the two models considered for the 2D symmetric flapping with flexible wings. C_i, C_u and C_d stand respectively for the initial, current undeformed and current deformed configuration of the wings. The green parts of the wings represent the points whose position is following the rigid body motion, while the red parts represent the parts that are allowed to deform.

close to what can be seen in reality are obtained (see [134]). The main but significant difference with the Model A is that the tip wing position is prescribed, which influences the overall position of the wing during its stroke. The different deformations generated with both models for the same physical parameters are shown in Figure B.2 for half a stroke. Based on the same 2D model, De Rosis et al. [27] used a different approach to solve the wings deformation. In the latter, the overall body motion is split into a rigid motion component and a pure deformation component with respect to the first one. At each time step, the wing is firstly moved according to the rigid motion, and then the iterative solid solver is used to compute the deformation. The deformable solid modelling is based on the co-rotational formulation of Finite Elements [36]. This iterative model allows larger amplitude of the wings oscillations for a given imposed rigid motion. Large amplitude can strongly affect the results regarding the lift generation.

In the current 2D model, the rotation of the insect is not considered. Hence, the motion equation for the center of mass $\mathbf{X_c}$ of the insect is governed by Newton's law as



(a) Model A

(b) Model B

Figure B.2: Iso-contours of the distance function to the IB solid representing the wing. The black, blue and red contours stand respectively for the initial configuration, the wing configuration after quarter of a stroke, and the wing configuration after half a stroke. For sake of simplicity, only the right wing is considered as deformable for the model A.

follow:

$$\frac{d\mathbf{U_c}}{dt} = \frac{1}{M}\mathbf{F} + \mathbf{G} \tag{B.2}$$

$$\frac{d\mathbf{X_c}}{dt} = \mathbf{U_c} \tag{B.3}$$

where **F** stands for the fluid force acting on the wing. The latter is obtained from the IBM force as it is described in the previous chapter (see Equation 6.6). Equations B.2 and B.3 are solved using a second-order Adams-Bashforth scheme.

B.2 Results and discussion

The computational conditions are based on the configuration of Ota et al. [97], except the size of the domain which is smaller in the current case with a width of $6L$ and a height of $12L$. The bounce-back boundary condition is used to model the no-slip walls at the fours edges of the domain. The latter is discretized with 360×720 lattice grid points and the Lagrangian grid has an equivalent spacing, giving 60 points per wings. The influence of gravity is not taken into account in this preliminary study. All the physical variables are non-dimensonalized following the definition of Ota et al. [97], using the wing length L, the time-averaged wing tip speed $\bar{u_{tip}}$, and a reference fluid density ρ_0. For the following simulations, the amplitude of the oscillations is set equal to $\Delta\theta = 2\pi \times 0.13 = 46.8^\circ$ and the initial phase is set to 0, while the non-dimensional mass M^* is fixed to 9.05. Finally, the internal bending force has no influence on the wing deformation, as the Model B introduced earlier is prescribing the positions of the wings extremities.

The two remaining parameters, the Reynolds number Re and the non-dimensional tension coefficient ϕ^*, are listed in Table B.1 for the different simulations. Smaller values of the tension coefficient are leading to non-physical behaviour or numerical instabilities, while larger values generate rigid-like wings. The difference of deformation between the two values of the coefficient ϕ^* is illustrated in Figure B.3. The trajectory of the insect's center of mass is first validated against the results of Ota et al. [97] for a rigid wing and with $Re = 200$, which corresponds to Simulation 2. The vertical position of the hinge is shown in Figure B.4 for the 10 first strokes. The results are in reasonable agreement with those of Ota et al. A similar level of discrepancy has been found for instance by De Rosis et al. [27]. It is important to note that the insect trajectory is very sensitive to the numerical parameters, as a small difference in the initial strokes due to a small difference of fluid force estimation, can have a significant impact on the next strokes.

The non-dimensional mass in the motion equation (Equation B.2) has to be multiplied by approximately a factor 10 to obtain a non-dimensional density ratio ρ^* (see Equation

	Sim. 1	Sim. 2	Sim. 3	Sim. 4	Sim. 5
Re	100	200	100	200	100
ϕ^*	rigid	rigid	0.01	0.01	0.003

Table B.1: Parameters of the fluid and solid solvers.

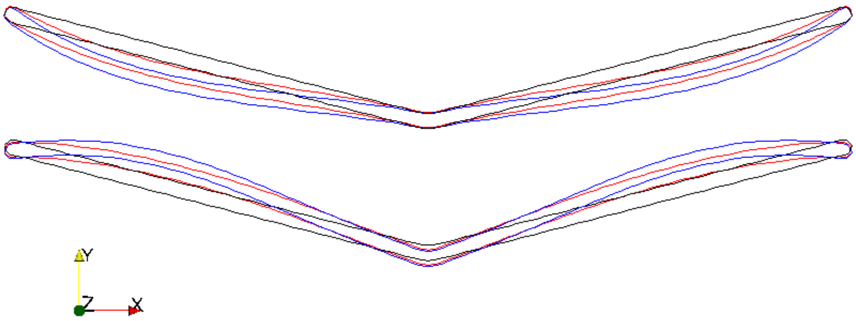


Figure B.3: Iso-contours of the distance function to the IB solid representing the wing, after $0.2T$ (top) and after $0.8T$ (bottom). The black contours represent the rigid case, while the red and blue contours display respectively a flexible wing with a non-dimensional bending coefficient of 0.01 and of 0.003.

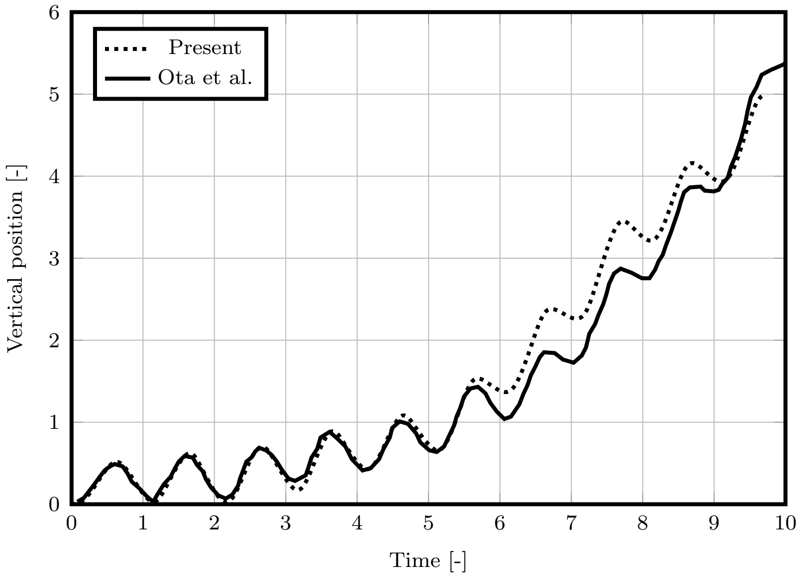
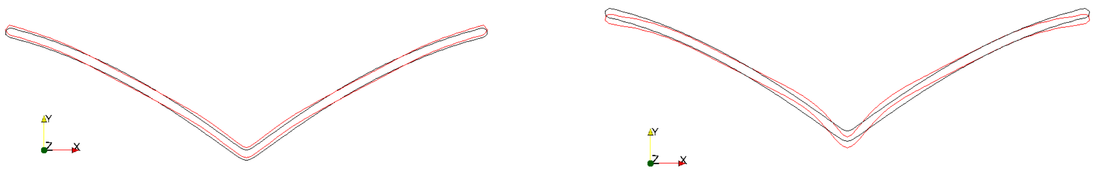


Figure B.4: Vertical trajectory of the hinge (center of mass) for a rigid wing at $Re = 200$.

7.1) that allows a stable behaviour of the solid solver. For the same reasons, a damping effect is considered with a non-dimensional coefficient $\lambda^* = 0.2$. Tuning these two coefficients increases the wings inertia, reduces spurious oscillations and avoids their resonance, for a stable wing motion. Figure B.5 shows the deformation of the wings obtained with and without a damping effect. Without damping, spurious oscillations are amplified through time, eventually leading to the iterative solution failing to converge. The arbitrary choice of ρ^* and λ^* is not an issue here, as the main interest is to obtain a physical deformation of the insect wing during the flapping mechanism.



(a) after $1.15T$

(b) after $2.15T$

Figure B.5: Iso-contours of the distance function to the IB solid representing the wing. The black contour represents the flexible wing position computed with a damping coefficient. The red contour represents the flexible wing position computed without a damping effect.

The trajectories of the wing for the 5 simulations listed in Table B.1 are presented in Figure B.6. It shows that the wing's flexibility, modelled as it is here, is an asset in specific condition thanks to an improvement of the average lift, while it appears as a drawback for other case. Hence, at $Re = 100$, a slight flexibility improves the generation of lift, as the body moves up faster than for a rigid wing after the 5 first strokes. However, increasing again the flexible nature of the wing will vanish this advantage as it shown for the Simulation 5 (dashed black line). Similarly at higher Reynolds, the deformation of the wing seems to reduce the production of the upward force. Looking at the flow fields allows one to understand why the flexibility is not increasing the lift force. For both high Reynolds and high flexibility cases, the symmetric nature of the flow, with respect to the vertical line, is eventually broken after a certain number of strokes. As a result, the vortices provide less support to the lift generation. The symmetry breaking process is visible in Figure B.7, which displays the vortical structures for both rigid and flexible wings. While the symmetry is still preserved after 3 strokes, tiny differences are already present after 5 strokes. The emergence of asymmetry is pushing the vortices apart reducing the support to the lift as it can been clearly seen after 9 strokes.

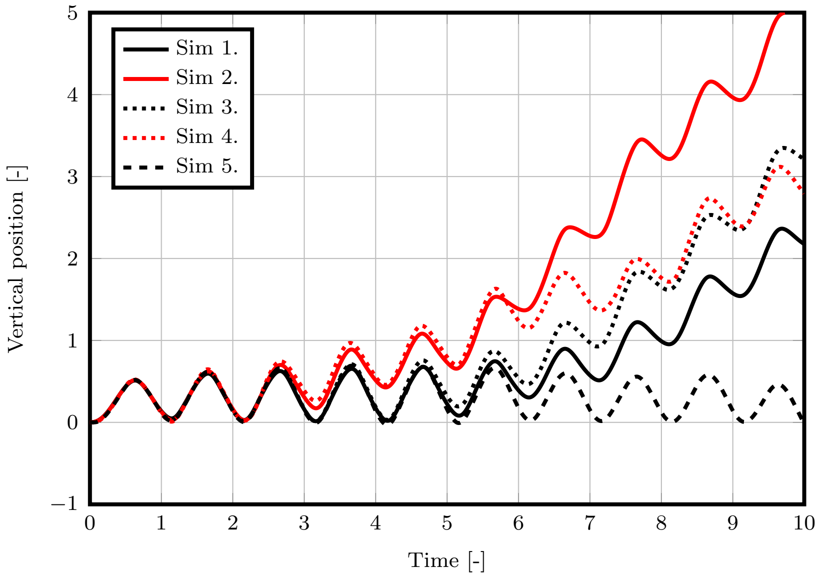
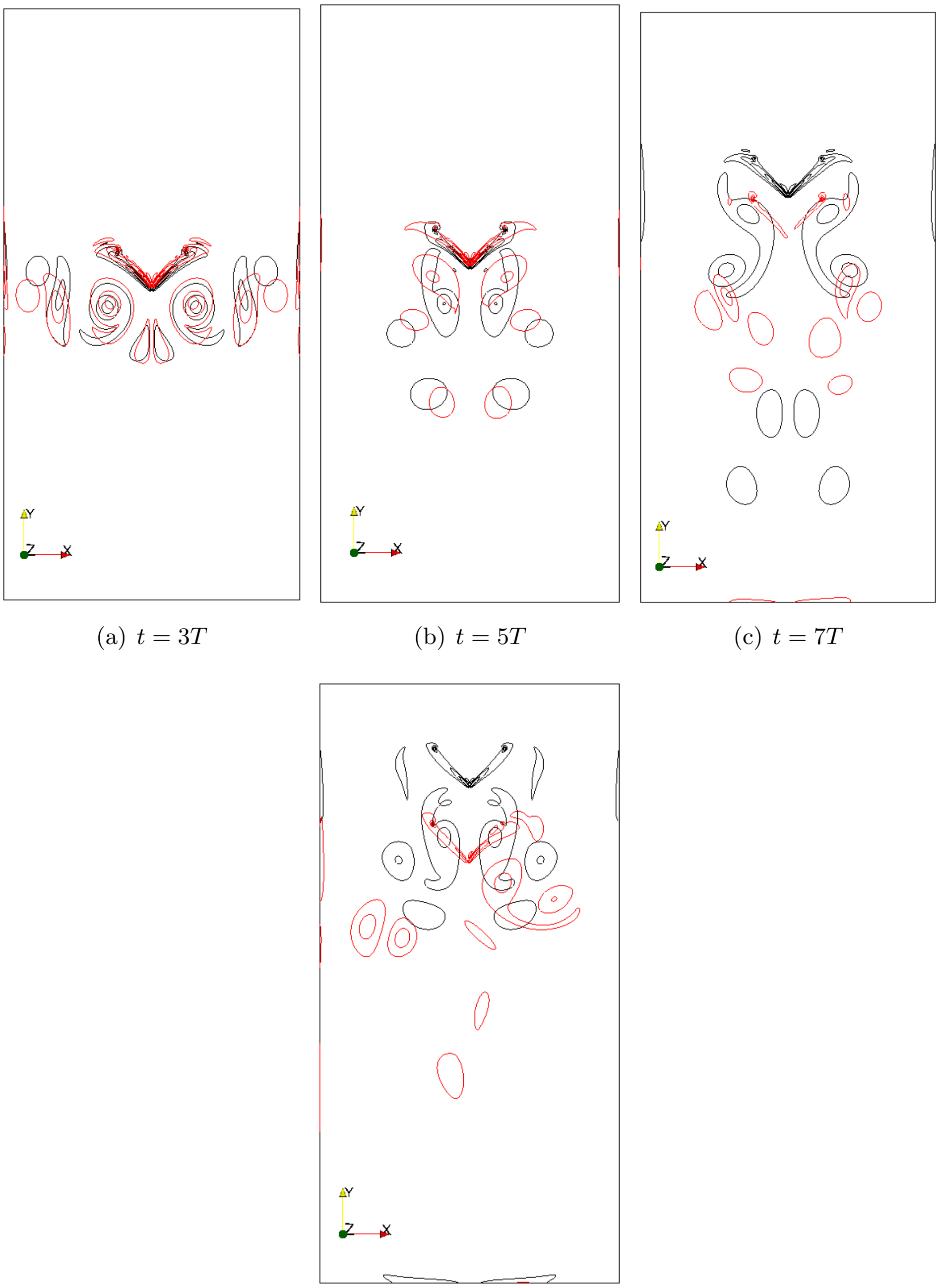


Figure B.6: Vertical trajectory of the hinge for all simulations.

Regarding the case where the deformability of the wing represents an asset for the lift generation (Simulation 3), the symmetry of the flow field is preserved as one can see in Figure B.8. In the rigid case, a vortex is present around the tip of each wing at the end of the upstroke. This vortex is acting against the lift force, whereas the vortices below the wing which were generated during the downstroke contribute to the lift force. The slight bending of the wing in the flexible case prevents the generation of these vortices at the wings extremities, and thus enhances the lift force. One can also notice the presence of the two secondary small vortices in the flexible case that are helping to support the lift generation.

The high sensitivity of the insect trajectory to the numerical parameters makes it difficult to draw a conclusion on the benefits of wing flexibility. Investigations on the sensitivity to the amplitude or period of the flapping oscillations can also be performed. However the main limitation of the model considered in this study for wing deformation lies in the prescription of the tips positions. In reality, a flexible wing might exhibit larger motion at the tips in comparison to an equivalent rigid wing with the same amplitude of oscillation. With a model able to do so, De Rosis et al. [27] shows that the flexibiltity of the wing can increase up to a factor 3 the maximum vertical position reached by the insect after 8 strokes for instance.

This test case suggests that taking into account the flexibility of the insect wings can have a significant impact on lift generation and thus on the flight capabilities of the insect. However, this type of problem has shown to be very sensitive to both numerical and physical parameters, making it hard to draw a conclusion on the benefit of the wing's deformation. Furthermore, the restrictive boundary condition used for the wing displacement are not necessarily realistic. For a more physical reproduction of the insect flight, its is recommended to use a full 3D model, without prescription of the displacement at boundaries, and with non-uniform stiffness over the flexible surface, as it is done in [95] and [134]. With this type of modelling, the structural dynamics is however solved with FEM for better accuracy and stability.



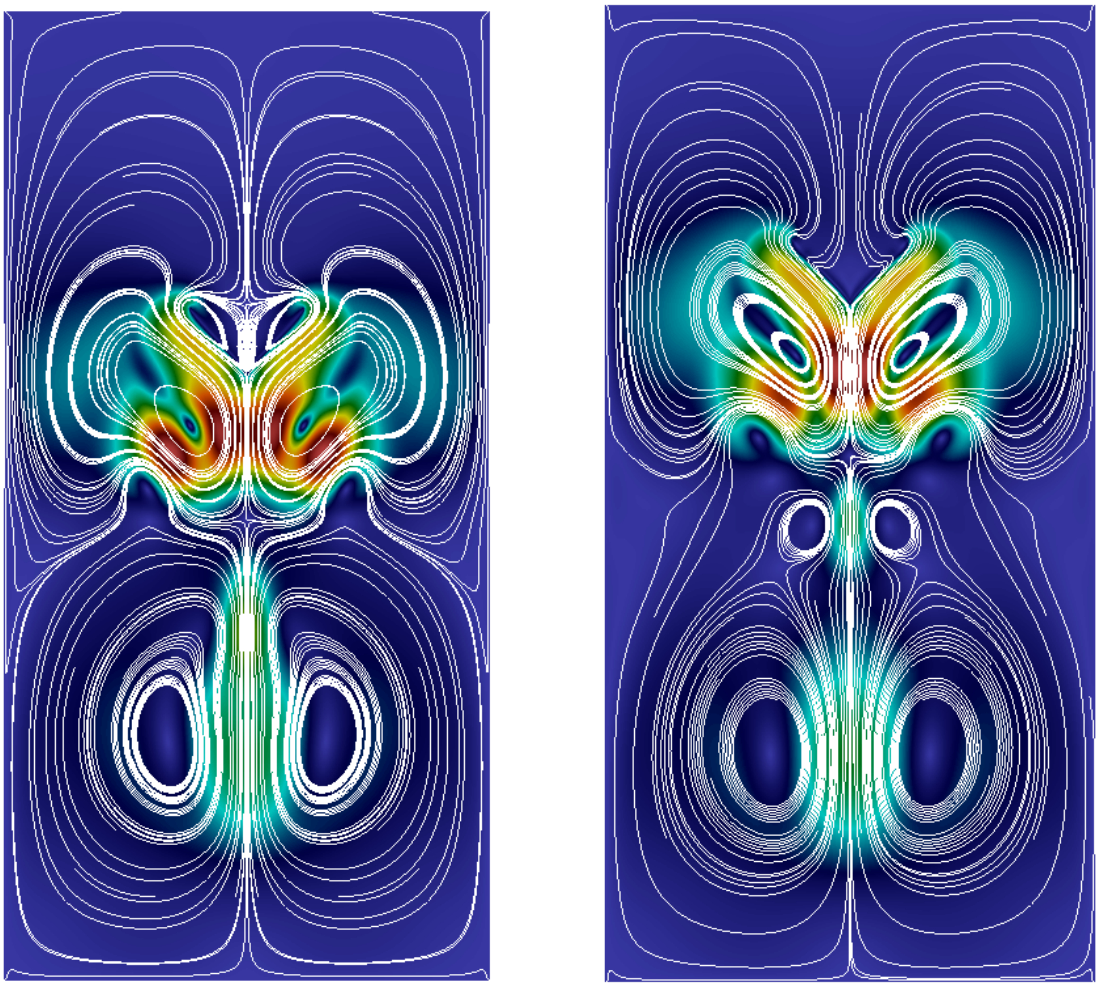
(a) $t = 3T$

(b) $t = 5T$

(c) $t = 7T$

(d) $t = 9T$

Figure B.7: Vorticity contours of the flow produced by a rigid (in black) and a flexible (in red) wing at different times at $Re = 200$.



(a) rigid wing

(b) flexible wing

Figure B.8: Streamlines and velocity contours of the flow produced by the flapping wings after 9 strokes at $Re = 100$.

Appendix C

Design and assembly of a closed loop water tunnel and visualisation system

A closed loop water tunnel for visualisation purposes has been designed, manufactured, and assembled by third parties. The design of the water tunnel and the piping system is shown in Figure C.1. The tunnel is composed of the following parts as labelled in Figure C.1: (1) Inlet section, (2) Dye injection section, (3) Test section, (4) Laser protection table, (5) Discharge section, (6) Discharge pipe, (7) Discharge reservoir, (8) Pump, (9) Piping connections, (10) Laser table, Power unit, and Laser Head, (11) Support table, (12) Flowmeter, (13) Electrical valve, (14) Inlet pipe. The tunnel allows for studying rag oscillations under the effect of several flow rates, as well as rag release. In order to study the release, the rag is clamped to a flagpole in the test section as shown in Figure C.2.

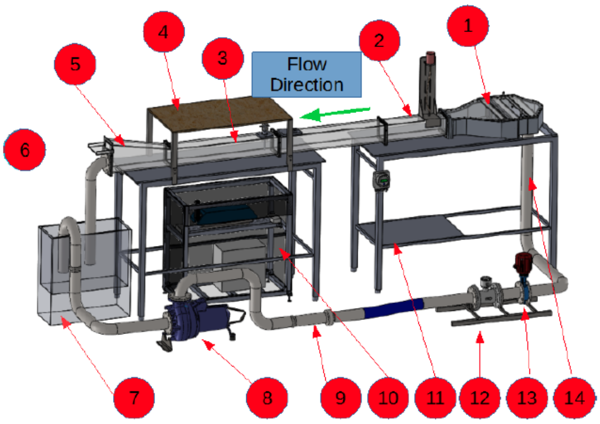


Figure C.1: Design of the closed loop water tunnel with piping system.

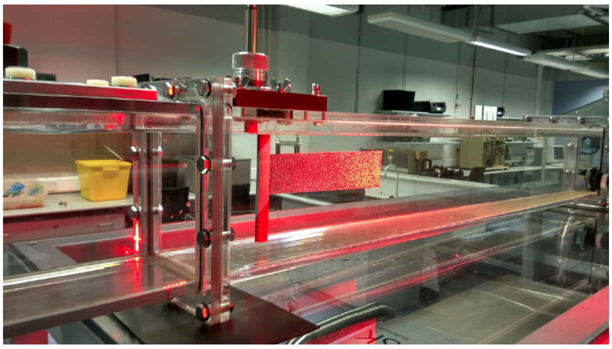


Figure C.2: The flagpole system for the clamping mechanism of the rag inside the testing section.

Figure C.3 shows a schematic diagram of the measurement tools and their position regarding the testing section. The setup illustrated in this figure allows for capturing the deformation of the rag during its oscillation under the effect of the fluid flow. This is achieved using the Digital Image Correlation (DIC) measurement tool. The Particle Image Velocimetry (PIV) system, on the other hand, allows for analysing the characteristics of the fluid flow in the wake region behind the rag.

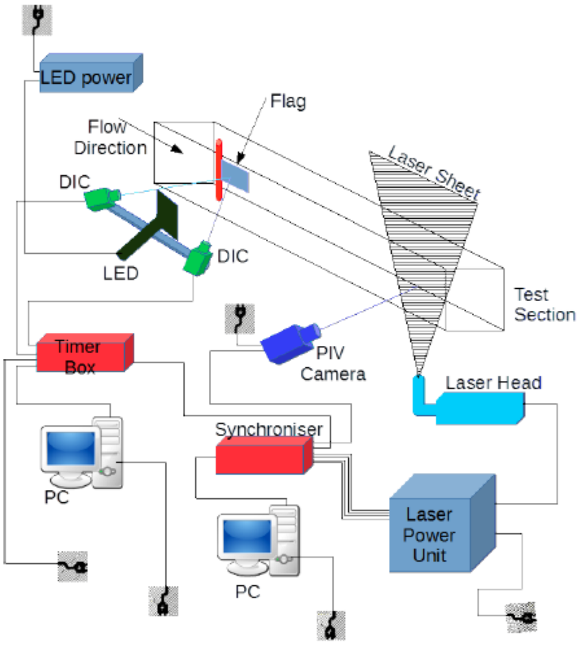


Figure C.3: Schematic Diagram of the Water tunnel incorporated with the measurement tools (PIV - DIC).

Appendix D

Influence of the collision distance for the clogging simulation

A collision distance of $2h$ was compared to $1.25h$, where h is the lattice Spacing. It is found that a smaller collision distance is only slightly slowing the collision mechanism and delaying the rag release. Very few changes can be identified however in the rag's behaviour as illustrated in Figure D.1. Figure D.2 shows the radial and axial positions of the rag in the volute during its transport and its collision with the impeller. The similarities for the two cases are highlighted by the concordance between the trajectories, especially for the axial position of the rags. The small delay in the rag release can be seen for both coordinates of the point C. These results demonstrate the low sensitivity of the numerical collision distance on the collision mechanism, if the parameter is chosen small enough. The same conclusion was drawn for different simulations, when varying the rag's initial position and length.

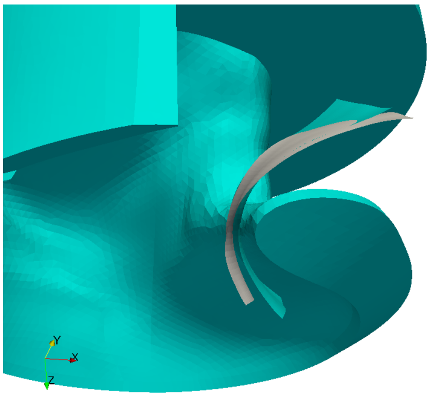
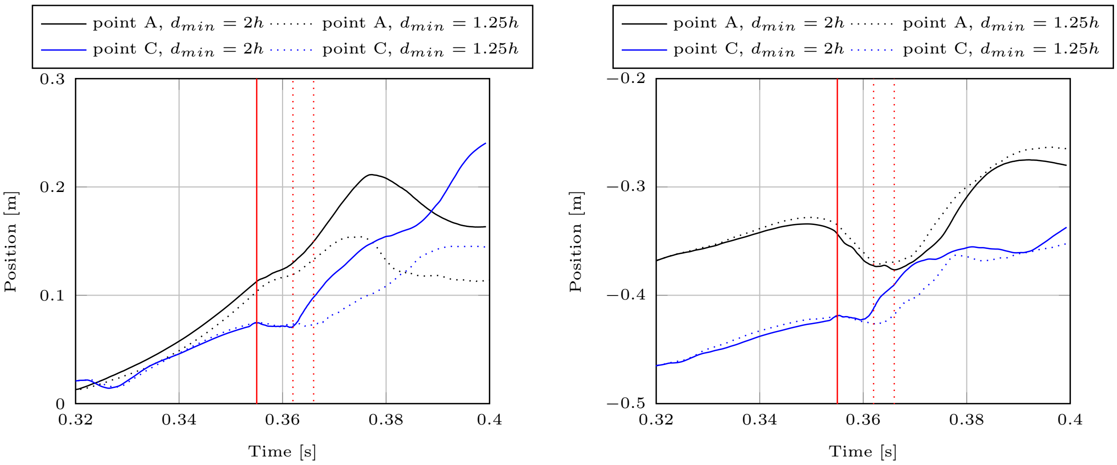


Figure D.1: Position of the rag during the collision process for two values of the collision distance threshold. The grey surface is obtained for a collision with a threshold of $2h$, while the light blue surface for a threshold of $1.25h$.



(a) Radial distance

(b) Axial distance

Figure D.2: Influence of the collision distance threshold on the rag transport in the pump. The vertical red lines denote approximately the start and the ends (dotted lines) of the collision processes for the two cases.

Bibliography

[1] OpenFOAM v2.0.1: Numerical Methods. pages 119

[2] R. Abgrall, H. Beaugendre, and C. Dobrzynski. An immersed boundary method using unstructured anisotropic mesh adaptation combined with level-sets and penalization techniques. *Journal of Computational Physics*, 257(PA):83–101, 2014. pages 36

[3] D T Akcabay. *Physics Based Washing Machine Simulations.* PhD thesis, University of Michigan, 2007. pages 8

[4] Steven R Allmaras, Forrester T Johnson, and Philippe R. Spalart. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. *Seventh International Conference on Computational Fluid Dynamics*, (ICCFD7-1902):1–11, 2012. pages 73

[5] Dionysios Angelidis, Saurabh Chawdhary, and Fotis Sotiropoulos. Unstructured Cartesian refinement with sharp interface immersed boundary method for 3D unsteady incompressible flows. *Journal of Computational Physics*, 325(August):272–300, 2016. pages 44

[6] Philippe Angot. *Contribution à l'étude des transferts thermiques dans des systèmes complexes aux composants électroniques.* PhD thesis, Université Bordeaux I, 1989. pages 32

[7] Philippe Angot, Charles-henri Bruneau, and Pierre Fabrie. Numerische Mathematik A penalization method to take into account obstacles in incompressible viscous flows. *Numerische Mathematik*, 81:497–520, 1999. pages xvi, 29, 32, 55

[8] E. Arquis and J. P. Caltagirone. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieu poreux: application àla convection naturelle. *Comptes Rendus de l'Académie des Sciences, Paris 2*, 299:1–4, 1984. pages 31, 54

[9] Elias Balaras. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Computers & Fluids*, 33(3):375–404, mar 2004. pages 34, 44, 57, 74, 112

[10] M. Bergmann and A. Iollo. Modeling and simulation of fish-like swimming. *Journal of Computational Physics*, 230(2):329–348, jan 2011. pages 28, 36, 40, 44

[11] Matteo Bernardini, Davide Modesti, and Sergio Pirozzoli. On the suitability of the immersed boundary method for the simulation of high-Reynolds-number separated turbulent flows. *Computers and Fluids*, 130:84–93, 2016. pages 43, 44

[12] R. P. Beyer and R. J. Leveque. Analysis of a One-Dimensional Model for the Immersed Boundary Method. *SIAM Journal on Numerical Analysis*, 29(2):332–364, 1992. pages 27

[13] Iman Borazjani, Liang Ge, and Fotis Sotiropoulos. Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies. *Journal of Computational Physics*, 227(16):7587–7620, aug 2008. pages 29, 37, 38, 61

[14] M'hamed Bouzidi, Mouaouia Firdaouss, and Pierre Lallemand. Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Physics of Fluids*, 13(11):3452–3459, 2001. pages 22

[15] Christopher Brennen. *Hydrodynamics of Pumps.* Cambridge University Press, Cambridge, 2011. pages 3

[16] J. M. Buick and C. A. Greated. Gravity in a lattice Boltzmann model. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 61(5A):5307–20, 2000. pages 24, 47

[17] W Cabot and P Moin. Approximate wall boundary conditions in the large-eddy simulation of high Reynolds number flow. *Flow, Turbulence and Combustion*, 63:269–291, 2000. pages 43

[18] Antoni Calderer, Seokkoo Kang, and Fotis Sotiropoulos. Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures. *Journal of Computational Physics*, 277:201–227, 2014. pages 38, 42, 44, 61

[19] Po Hua Chang, Chuan Chieh Liao, Hsin Wei Hsu, Shih Huang Liu, and Chao An Lin. Simulations of laminar and turbulent flows over periodic hills with immersed boundary method. *Computers and Fluids*, 92:233–243, 2014. pages 42, 43, 76, 258

[20] Frédéric Chantalat, Charles Henri Bruneau, Cédric Galusinski, and Angelo Iollo. Level-set, penalization and cartesian meshes: A paradigm for inverse problems and optimal design. *Journal of Computational Physics*, 228(17):6291–6315, 2009. pages 36, 55

[21] Jung-Il Choi, Roshan C. Oberoi, Jack R. Edwards, and Jacky A. Rosati. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224(2):757–784, jun 2007. pages 42, 76, 160

[22] Benjamin S H Connell. *Numerical investigation of the flow-body interaction of thin flexible foils and ambient flow.* PhD thesis, Massachusetts Institute Of Technology, 2006. pages 193

[23] Robert Connolly. *An Experimental and Numerical Investigation into Flow Phenomena Leading to Wastewater Centrifugal Pump Blockage.* PhD thesis, Dublin City University, 2017. pages 247

[24] Madeleine Coutanceau and Roger Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation . Part 1 . Steady flow. *Fluid Mechanics*, 79:231–256, 1977. pages 89

[25] Antonio Cristallo and Roberto Verzicco. Combined Immersed Boundary/Large-Eddy-Simulations of Incompressible Three Dimensional Complex Flows. *Flow, Turbulence and Combustion*, 77(1-4):3–26, 2006. pages 42, 43

[26] Daily Mail Australia. Pictured: The 500 tonnes of putrid WET WIPES choking our cities' sewers... and comfort-loving MEN are to blame because they like the 'softer feel' in the bathroom, 2015. pages 1

[27] Alessandro De Rosis, Giacomo Falcucci, Stefano Ubertini, and Francesco Ubertini. Aeroelastic study of flexible flapping wings by a coupled lattice Boltzmann-finite element approach with immersed boundary method. *Journal of Fluids and Structures*, 49:516–533, 2014. pages 48, 267, 269, 271, 275

[28] M.D. de Tullio and G. Pascazio. A moving-least-squares immersed boundary method for simulating the fluid-structure interaction of elastic bodies with arbitrary thickness. *Journal of Computational Physics*, 325(August):201–225, 2016. pages 41, 42, 195, 212, 216

[29] J. J. Derksen, M. S. Doelman, and H. E A van den Akker. Three-dimensional measurements in the baffle region of a turbulently stirred tank. *Experiments in Fluids*, 27:522–532, 1999. pages 150, 156

[30] D. D'Humières. Generalized lattice Boltzmann Equations, Rarefied Gas Dynamics: Theory and Simulations. *Progress in Astronautics and Aeronautics*, 159:450–458, 1992. pages 49

[31] Alexandre Dupuis, Philippe Chatelain, and Petros Koumoutsakos. An immersed boundary-lattice-Boltzmann method for the simulation of the flow past an impulsively started cylinder. *Journal of Computational Physics*, 227(9):4486–4498, 2008. pages 47

[32] H Dütsch, F Durst, S Becker, and H Lienhart. Low-Reynolds-number flow around an oscillating circular cylinder at low KeuleganCarpenter numbers. *Fluid Mechanics*, 360:249–271, 1998. pages 95, 175

[33] Thomas Engels, Dmitry Kolomenskiy, Kai Schneider, and Jörn Sesterhenn. Numerical simulation of fluid-structure interaction with the volume penalization method. *Journal of Computational Physics*, 281:96–115, 2015. pages 41

[34] E.a. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *Journal of Computational Physics*, 161(1):35–60, jun 2000. pages 32, 34, 46, 47, 57

[35] Julien Favier, Alistair Revell, and Alfredo Pinelli. A Lattice Boltzmann-Immersed Boundary method to simulate the fluid interaction with moving and slender flexible objects. *Journal of Computational Physics*, 261:145–161, 2014. pages 48, 49, 193, 195, 196, 197, 258

[36] Carlos A. Felippa and Bjorn Haugen. A unified formulation of small-strain corotational finite elements: I. Theory. *Computer Methods in Applied Mechanics and Engineering*, 194(21-24 SPEC. ISS.):2285–2335, 2005. pages 269

[37] J. Feng, F. K. Benra, and H. J. Dohmen. Unsteady flow visualization at part-load conditions of a radial diffuser pump: By PIV and CFD. *Journal of Visualization*, 12(1):65–72, 2009. pages 247

[38] Zhi Gang Feng and Efstathios E. Michaelides. The immersed boundary-lattice Boltzmann method for solving fluid-particles interaction problems. *Journal of Computational Physics*, 195(2):602–628, 2004. pages 45, 46

[39] Zhi Gang Feng and Efstathios E. Michaelides. Proteus: A direct forcing method in the simulations of particulate flows. *Journal of Computational Physics*, 202(1):20–51, 2005. pages 46, 47

[40] Mattia Gazzola, Philippe Chatelain, Wim M. van Rees, and Petros Koumoutsakos. Simulations of single and multiple swimmers with non-divergence free deforming geometries. *Journal of Computational Physics*, 230(19):7093–7114, aug 2011. pages xvi, 27, 29, 30, 32, 40

[41] Chun Yan Ge, Jia Jun Wang, Xue Ping Gu, and Lian Fang Feng. CFD simulation and PIV measurement of the flow field generated by modified pitched blade turbine impellers. *Chemical Engineering Research and Design*, 92(6):1027–1036, 2014. pages 150, 157

[42] Seyed Amin Ghaffari, Stéphane Viazzo, Kai Schneider, and Patrick Bontoux. Simulation of forced deformable bodies interacting with two-dimensional incompressible flows: Application to fish-like swimming. *International Journal of Heat and Fluid Flow*, 51:88–109, 2015. pages 40

[43] a. Gilmanov, F. Sotiropoulos, and E. Balaras. A general reconstruction algorithm for simulating flows with complex 3D immersed boundaries on Cartesian grids. *Journal of Computational Physics*, 191(2):660–669, nov 2003. pages 34, 35, 112

[44] Anvar Gilmanov and Sumanta Acharya. A hybrid immersed boundary and material point method for simulating 3D fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 56(August 2007):2151–2177, 2008. pages 40

[45] Anvar Gilmanov and Fotis Sotiropoulos. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics*, 207(2):457–492, aug 2005. pages 34, 35, 57

[46] R. Glowinski, T. W. Pan, T. I. Hesla, and D. D. Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999. pages 198

[47] D. Goldstein, R. Handler, and L. Sirovich. Modeling a No-Slip Flow Boundary with an External Force Field. *Journal of Computational Physics*, 105(2):354–366, apr 1993. pages 31

[48] Johann Friedrich Gülich. *Centrifugal Pumps.* Springer, 2008. pages 3, 247

[49] Zhaoli Guo, Chuguang Zheng, and Baochang Shi. An extrapolation method for boundary conditions in lattice Boltzmann method. *Physics of Fluids*, 14(6):2007–2010, 2002. pages 22, 23, 169, 170

[50] Zhaoli Guo, Chuguang Zheng, and Baochang Shi. Discrete lattice effects on the forcing term in the lattice Boltzmann method. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 65(4):1–6, 2002. pages 23, 24, 46, 47, 48, 49

[51] S. Haeri and J.S. Shrimpton. On the application of immersed boundary, fictitious domain and body-conformal mesh methods to many particle multiphase flows. *International Journal of Multiphase Flow*, 40:38–55, apr 2012. pages 15

[52] H. Hartmann, J.J. Derksen, C. Montavon, J. Pearson, I.S. Hamill, and H.E.A. van den Akker. Assessment of large eddy and RANS stirred tank simulations by means of LDA. *Chemical Engineering Science*, 59(12):2419–2432, jun 2004. pages 156, 157

[53] Yang Hu, Haizhuan Yuan, Shi Shu, Xiaodong Niu, and Mingjun Li. An improved momentum exchanged-based immersed boundary-lattice Boltzmann method by using an iterative technique. *Computers and Mathematics with Applications*, 68(3):140–155, 2014. pages 49

[54] Wei Xi Huang, Cheong Bong Chang, and Hyung Jin Sung. An improved penalty immersed boundary method for fluid-flexible body interaction. *Journal of Computational Physics*, 230(12):5061–5079, 2011. pages 41, 46

[55] Wei-Xi Huang, Soo Jai Shin, and Hyung Jin Sung. Simulation of flexible filaments in a uniform flow by the immersed boundary method. *Journal of Computational Physics*, 226(2):2206–2228, oct 2007. pages 8, 195, 197, 198, 258

[56] Wei-Xi Huang and Hyung Jin Sung. An immersed boundary method for fluidflexible structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2650–2661, 2009. pages 41, 192, 193, 195, 197, 217, 240, 241, 243, 254

[57] Wei-Xi Huang and Hyung Jin Sung. Three-dimensional simulation of a flapping flag in a uniform flow. *Journal of Fluid Mechanics*, 653:301–336, 2010. pages xxiv, 197, 212, 213, 214, 216, 226

[58] Gianluca Iaccarino, Roberto Verzicco, Politecnico Bari, and Via Re David. Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Reviews*, 56(3):331, 2003. pages 34

[59] Tsutomu Ikeno and Takeo Kajishima. Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations. *Journal of Computational Physics*, 226(2):1485–1508, oct 2007. pages 56

[60] C. Introïni, M. Belliard, and C. Fournier. A second order penalized direct forcing for hybrid Cartesian/immersed boundary flow simulations. *Computers & Fluids*, 90:21–41, feb 2014. pages 27, 36, 55, 57, 112

[61] Hrvoje Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows.* PhD thesis, Imperial College London, 1996. pages 64, 65, 80

[62] Anna Lyhne Jensen, Henrik Sørensen, and Lasse Rosendahl. Towards Simulation of Clogging Effects in Wastewater Pumps : Modelling of Fluid Forces on a Fiber of Bonded Particles using a Coupled CFD-DEM Approach. In *ISROMAC 2016 - International Symposium on Transport Phenomena and Dynamics of Rotating Machinery*, pages 1–6, 2016. pages 8

[63] Anna Lyhne Jensen, Henrik Sørensen, Lasse Rosendahl, Per Adamsen, and Flemming Lykholt-ustrup. Investigation of Drag Force on Fibres of Bonded Spherical Elements using a Coupled CFD-DEM Approach. In *ICMF 2016 - 9th International Conference on Multiphase Flow*, number 1922, 2016. pages 8

[64] C. Ji, A. Munjiza, and J. J R Williams. A novel iterative direct-forcing immersed boundary method and its finite volume applications. *Journal of Computational Physics*, 231(4):1797–1821, 2012. pages 42, 43

[65] Georgi Kalitzin and Iaccarino Gianluca. Toward Immersed Boundary Simulation of High Reynolds Number Flows. *Center for Turbulence Research*, pages 369–376, 2003. pages 43

[66] Seongwon Kang, Gianluca Iaccarino, Frank Ham, and Parviz Moin. Prediction of wall-pressure fluctuation in turbulent flows with an immersed boundary method. *Journal of Computational Physics*, 228(18):6753–6772, oct 2009. pages 35, 36, 67

[67] Shin K. Kang and Yassin A. Hassan. A direct-forcing immersed boundary method for the thermal lattice Boltzmann method. *Computers & Fluids*, 49(1):36–45, oct 2011. pages 45

[68] Tobias Kempe and Jochen Fröhlich. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *Journal of Computational Physics*, 231(9):3663–3684, 2012. pages 39

[69] Tobias Kempe, Matthias Lennartz, Stephan Schwarz, and Jochen Fröhlich. Imposing the free-slip condition with a continuous forcing immersed boundary method. *Journal of Computational Physics*, 282:183–209, 2015. pages 39

[70] Nicholas K Kevlahan and Jean-michel Ghidaglia. Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization. *European Journal of Mechanics, B/Fluids*, 20:333–350, 2001. pages 28, 32

[71] Khodor Khadra, Philippe Angot, Sacha Parneix, and Jean-paul Caltagirone. Fictitious domain approach for numerical modelling of Navier Stokes equations. 34(April 1998):651–684, 2000. pages 31, 36

[72] Rooh A. Khurram, Yue Zhang, and Wagdi G. Habashi. Multiscale finite element method applied to the Spalart-Allmaras turbulence model for 3D detached-eddy simulation. *Computer Methods in Applied Mechanics and Engineering*, 233-236:180–193, 2012. pages 73

[73] Jungwoo Kim, Dongjoo Kim, and Haecheon Choi. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *Journal of Computational Physics*, 171(1):132–150, jul 2001. pages 34, 35, 36, 39, 67

[74] Yongsam Kim and Charles S. Peskin. Penalty immersed boundary method for an elastic boundary with mass. *Physics of Fluids*, 19(5), 2007. pages 46, 208, 210, 241

[75] Manish Kumar and Somnath Roy. A sharp interface immersed boundary method for moving geometries with mass conservation and smooth pressure variation. *Computers and Fluids*, 137:15–35, 2016. pages 36

[76] Uis Lacis, Kunihiko Taira, and Shervin Bagheri. A stable fluid-structure-interaction solver for low-density rigid bodies using the immersed boundary projection method. *Journal of Computational Physics*, 305:300–318, 2016. pages 38

[77] A. J C Ladd and R. Verberg. Lattice-Boltzmann simulations of particle-fluid suspensions. *Journal of Statistical Physics*, 104(5-6):1191–1251, 2001. pages 24

[78] Ming-Chih Lai and Charles S. Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity. *Journal of Computational Physics*, 160(2):705–719, may 2000. pages 27, 32, 37, 167, 196, 198

[79] Pierre Lallemand and Li S. Luo. Lattice Boltzmann method for moving boundaries. *Journal of Computational Physics*, 184(2):406–421, 2003. pages 22

[80] Jonas Latt, Bastien Chopard, Orestis Malaspinas, Michel Deville, and Andreas Michler. Straight velocity boundaries in the lattice Boltzmann method. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 77(5):1–16, 2008. pages 22

[81] D. V. Le, J. White, J. Peraire, K. M. Lim, and B. C. Khoo. An implicit immersed boundary method for three-dimensional fluid-membrane interactions. *Journal of Computational Physics*, 228(22):8427–8445, 2009. pages 41

[82] Injae Lee and Haecheon Choi. A discrete-forcing immersed boundary method for the fluid-structure interaction of an elastic slender body. *Journal of Computational Physics*, 280:529–546, 2015. pages 40, 41, 212, 213, 216

[83] Ru-Yang Li, Chun-Mei Xie, Wei-Xi Huang, and Chun-Xiao Xu. An efficient immersed boundary projection method for flow over complex/moving boundaries. *Computers & Fluids*, 140:122–135, nov 2016. pages 95, 96

[84] Zhilin Li. *The Immersed Interface Method - A Numerical Approach for Partial Differential Equations with Interfaces*. PhD thesis, University of Washington, 1994. pages 15

[85] Chuan Chieh Liao, Yu Wei Chang, Chao An Lin, and J. M. McDonough. Simulating flows with moving rigid boundary using immersed-boundary method. *Computers and Fluids*, 39(1):152–167, 2010. pages 33, 35, 95, 96, 112, 174

[86] Kun Luo, Zeli Wang, and Jianren Fan. A modified immersed boundary method for simulations of fluidparticle interactions. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4):36–46, dec 2007. pages 63

[87] Andreas Mark and Berend G.M. van Wachem. Derivation and validation of a novel implicit second-order accurate immersed boundary method. *Journal of Computational Physics*, 227(13):6660–6680, jun 2008. pages 57, 67

[88] F.R. Menter, M. Kuntz, and R. Langtry. Ten Years of Industrial Experience with the SST Turbulence Model. In *Turbulence, Heant and Mass Transfer 4*. Begell House, Inc, oct 2003. pages 72

[89] C. Mimeau, G.-H. Cottet, and I. Mortazavi. Direct numerical simulations of three-dimensional flows past obstacles with a vortex penalization method. *Computers & Fluids*, 136:331–347, 2016. pages 36

[90] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825–4852, 2008. pages 35

[91] Rajat Mittal and Gianluca Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, jan 2005. pages 25

[92] Rajat Mittal and Gianluca Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, jan 2005. pages 67

[93] J. Mohd-Yusof. Combined immersed boundary/B-spline methods for simulation of flow in complex geometries. *Center for Turbulence Research Annual Research Briefs1*, pages 317–328, 1997. pages xvi, 29, 46, 47

[94] New York Times. Wet Wipes Box Says Flush. New York's Sewer System Says Don't., 2015. pages 1

[95] T.T. Nguyen, Dhanabalan Shyam Sundar, Khoon Seng Yeo, and Tee Tai Lim. Modeling and analysis of insect-like flexible wings at low Reynolds number. *Journal of Fluids and Structures*, 62:294–317, apr 2016. pages 267, 275

[96] X. D. Niu, C. Shu, Y. T. Chew, and Y. Peng. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. *Physics Letters, Section A: General, Atomic and Solid State Physics*, 354(3):173–182, 2006. pages 49

[97] Keigo Ota, Kosuke Suzuki, and Takaji Inamuro. Lift generation by a two-dimensional symmetric flapping wing: immersed boundary-lattice Boltzmann simulations. *Fluid Dynamics Research*, 44(4):045504, 2012. pages 5, 8, 47, 48, 166, 241, 267, 271

[98] Allmaras P R, Spalart, W-H, Jou, M, Strelets, S R. Comments on the feasibility of LES for wings and on a hybrid RANS/LES approach. *Proceedins of the first AFOSR International Conference on DNS/LES*, 1997. pages 72

[99] Dingyi Pan, Xueming Shao, Jian Deng, and Zhaosheng Yu. Simulations of passive oscillation of a flexible plate in the wake of a cylinder by immersed boundary method. *European Journal of Mechanics, B/Fluids*, 46:17–27, 2014. pages 41, 195

[100] Lucia Parussini and Valentino Pediroda. Fictitious Domain approach with hp -finite element approximation for incompressible fluid flow. *Journal of Computational Physics*, 228(10):3891–3910, 2009. pages 89

[101] Charles S. Peskin. Flow Patterns Around Heart Valves: A Numerical Method. *Journal of Computational Physics*, 10:252–271, 1972. pages xvi, 25, 29, 30, 45

[102] A. Pinelli, I. Z. Naqavi, U. Piomelli, and J. Favier. Immersed-boundary methods for general finite-difference and finite-volume Navier-Stokes solvers. *Journal of Computational Physics*, 229(24):9073–9091, 2010. pages 48

[103] A. Piquet, O. Roussel, and A. Hadjadj. A comparative study of Brinkman penalization and direct-forcing immersed boundary methods for compressible viscous flows. *Computers and Fluids*, 136:272–284, 2016. pages 50

[104] Antonio Posa, Antonio Lippolis, and Elias Balaras. Investigation of Separation Phenomena in a Radial Pump at Reduced Flow Rate by Large-Eddy Simulation. *Journal of Fluids Engineering*, 138(12):121101, aug 2016. pages 8

[105] Antonio Posa, Antonio Lippolis, Roberto Verzicco, and Elias Balaras. Large-eddy simulations in mixed-flow pumps using an immersed-boundary method. *Computers and Fluids*, 47(1):33–43, 2011. pages 8, 44

[106] Narsimha R. Rapaka and Sutanu Sarkar. An immersed boundary method for direct and large eddy simulation of stratified flows in complex geometry. *Journal of Computational Physics*, 322:511–534, 2016. pages 43

[107] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flowpast an airfoil with trailing edge separation. *The American Institute of Aeronautics and Astronautics*, 21:1525–1532, 1983. pages 65, 67

[108] L. F. Richardson. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses I a Masonry Dam. *Transactions of the Royal Society of London Ser. A*, 210:307–357, 1910. pages 83, 93, 221

[109] F. Roman, V. Armenio, and J. Fröhlich. A simple wall-layer model for large eddy simulation with immersed boundary method. *Physics of Fluids*, 21(10):10–14, 2009. pages 42

[110] D. H. Rothman and S. Zaleski. *Lattice-Gas Cellular Automata.* Cambridge University Press, 1997. pages 20

[111] Arthur Sarthou, Stéphane Vincent, Philippe Angot, and Jean-Paul Caltagirone. The Sub-Mesh Penalty Method. In *5th International Symposium on Finite*

Volumes for Complex Applications, page 633, 2009. pages xvi, 29, 30, 36, 55, 57, 59, 60, 80, 112

[112] Arthur Sarthou, Stéphane Vincent, J. P. Caltagirone, and Philippe Angot. Eulerian-Lagrangian grid coupling and penalty methods for the simulation of multiphase flows interacting with complex objects. *International Journal for Numerical Methods in Fluids*, 56(8):1093–1099, mar 2008. pages 57

[113] S. Schmidt and M. Breuer. Hybrid LES-URANS methodology for the prediction of non-equilibrium wall-bounded internal and external flows. *Computers and Fluids*, 96:226–252, 2014. pages 73

[114] S. Schwarz, T. Kempe, and J. Fröhlich. A temporal discretization scheme to compute the motion of light particles in viscous flows by an immersed boundary method. *Journal of Computational Physics*, 281:591–613, 2015. pages 38

[115] Stephan Schwarz, Tobias Kempe, and Jochen Fröhlich. An immersed boundary method for the simulation of bubbles with varying shape. *Journal of Computational Physics*, 315:124–149, 2016. pages 39, 40

[116] Jung Hee Seo and Rajat Mittal. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of Computational Physics*, 230(19):7347–7363, 2011. pages 95

[117] Mikhail L. Shur, Philippe R. Spalart, Mikhail Kh Strelets, and Andrey K. Travin. A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities. *International Journal of Heat and Fluid Flow*, 29(6):1638–1649, dec 2008. pages 72

[118] J. Sigüenza, S. Mendez, D. Ambard, F. Dubois, F. Jourdan, R. Mozul, and F. Nicoud. Validation of an immersed thick boundary method for simulating fluid-structure interactions of deformable membranes. *Journal of Computational Physics*, 322:723–746, 2016. pages 41

[119] Fotis Sotiropoulos and Xiaolei Yang. Immersed boundary methods for simulating fluidstructure interaction. *Progress in Aerospace Sciences*, 65:1–21, feb 2014. pages 14, 25, 37

[120] P. SPALART and S. ALLMARAS. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, number 1, pages 5–21, Reston, Virigina, jan 1992. American Institute of Aeronautics and Astronautics. pages 74

[121] P. R. Spalart, S. Deck, M. L. Shur, K. D. Squires, M. Kh Strelets, and A. Travin. A new version of detached-eddy simulation, resistant to ambiguous grid densities. *Theoretical and Computational Fluid Dynamics*, 20(3):181–195, 2006. pages 153

[122] D. B. Spalding. A Single Formula for the Law of the Wall. *Journal of Applied Mechanics*, 28(3):455, 1961. pages 75, 124

[123] Shen Wei Su, Ming Chih Lai, and Chao An Lin. An immersed boundary technique for simulating complex flows with rigid boundary. *Computers and Fluids*, 36(2):313–324, 2007. pages xvi, 28, 29

[124] M. C. Sukop and T. D. Thorne. *Lattice Boltzmann Modeling.* Springer, 2006. pages 20

[125] Kosuke Suzuki and Takaji Inamuro. Effect of internal mass in the simulation of a moving body by the immersed boundary method. *Computers and Fluids*, 49(1):173–187, 2011. pages 5, 8, 47, 48, 166, 168

[126] Kosuke Suzuki and Takaji Inamuro. A higher-order immersed boundary-lattice Boltzmann method using a smooth velocity field near boundaries. *Computers and Fluids*, 76:105–115, 2013. pages 47, 48

[127] Sydney Water Corporation. Keep wipes out of the pipes!, 2015. pages xv, 1, 2

[128] Kunihiko Taira and Tim Colonius. The immersed boundary method: A projection approach. *Journal of Computational Physics*, 225(2):2118–2137, aug 2007. pages 88, 89

[129] W. B. Tay, B. W. van Oudheusden, and H. Bijl. Numerical simulation of a flapping four-wing micro-aerial vehicle. *Journal of Fluids and Structures*, 55:237–261, 2015. pages 267

[130] Lionel Temmerman, Michael A. Leschziner, Christopher P. Mellen, and Jochen Fröhlich. Investigation of wall-function approximations and subgrid-scale models in large eddy simulation of separated flow in a channel with streamwise periodic constrictions. *International Journal of Heat and Fluid Flow*, 24(2):157–180, 2003. pages 76

[131] F. Tessicini, G. Iaccarino, M. Fatica, M. Wang, and R. Verzicco. Wall modeling for large-eddy simulation using an immersed boundary method. *Center of Turbulence Research, Annual Research Briefs*, (1975):181–187, 2002. pages 43

[132] Fang-Bao Tian, Hu Dai, Haoxiang Luo, James F. Doyle, and Bernard Rousseau. Fluid-structure interaction involving large deformations: 3D simulations and applications to biological systems. *Journal of Computational Physics*, 258:451–469, 2014. pages 212, 216

[133] Fang Bao Tian, Haoxiang Luo, Luoding Zhu, James C. Liao, and Xi Yun Lu. An efficient immersed boundary-lattice Boltzmann method for the hydrodynamic interaction of elastic filaments. *Journal of Computational Physics*, 230(19):7266–7283, 2011. pages 46, 195, 241

[134] S. Tobing, J. Young, and J. C.S. Lai. Effects of wing flexibility on bumblebee propulsion. *Journal of Fluids and Structures*, 68(October 2016):141–157, 2017. pages 267, 268, 269, 275

[135] By D J Tritton. Experiments on the flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 6:547–567, 1959. pages 89

[136] Yu-Heng Tseng and Joel H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, dec 2003. pages xvi, 28, 29, 34, 35, 57, 59

[137] Mayank Tyagi and Sumanta Acharya. Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method. *Journal for Numerical Methods in Fluids*, 48(December 2004):691–722, 2005. pages 34, 44

[138] Markus Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, nov 2005. pages xvi, 29, 33, 36, 39

[139] United States EPA. Wastewater Management Fact Sheet: Energy Conservation, 2006. pages 1

[140] S. P. van der Pijl, A Segal, C Vuik, and P. Wesseling. A mass-conserving Level-Set method for modelling of multi-phase flows. *International Journal for Numerical Methods in Fluids*, 47(March 2004):339–361, 2005. pages 62

[141] H. A. van der Vorst. Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, mar 1992. pages 194

[142] R. Verzicco, M. Fatica, G. Iaccarino, and P. Orlandi. Flow in an impeller-stirred tank using an immersed-boundary method. *AIChE Journal*, 50(6):1109–1118, jun 2004. pages 156

[143] Stéphane Vincent, Jorge César Brändle de Motta, Arthur Sarthou, Jean-Luc Estivalezes, Olivier Simonin, and Eric Climent. A Lagrangian VOF tensorial penalty method for the DNS of resolved particle-laden flows. *Journal of Computational Physics*, 256:582–614, jan 2014. pages 29, 32, 39, 113, 171

[144] Stéphane Vincent, Arthur Sarthou, Jean-Paul Caltagirone, Fabien Sonilhac, Pierre Février, Christian Mignot, and Grégoire Pianet. Augmented Lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. Application to hydroplaning flows interacting with real tire tread patterns. *Journal of Computational Physics*, 230(4):956–983, feb 2011. pages 44

[145] Chengjie Wang and Jeff D. Eldredge. Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method. *Journal of Computational Physics*, 295:87–113, 2015. pages 39

[146] Meng Wang and Parviz Moin. Dynamic wall modeling for large-eddy simulation of complex turbulent flows. *Physics of Fluids*, 14(7):2043, 2002. pages 43

[147] Zeli Wang, Jianren Fan, and Kun Luo. Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles. *International Journal of Multiphase Flow*, 34(3):283–302, mar 2008. pages 8, 33, 39, 47, 166

[148] G H Wannier. A contribution to hydrodynamics of lubrication. *Quarterly of Applied Mathematics*, 8:1–32, 1950. pages 82

[149] H Werner and H Wengle. Large-Eddy Simulation of Turbulent Flow Over and Around a Cube in a Plate Channel. In *Turbulent Shear Flows 8*, pages 155–168. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993. pages 76

[150] J. Wu and C. Shu. Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications. *Journal of Computational Physics*, 228(6):1963–1979, 2009. pages 48

[151] J. Wu and C. Shu. An improved immersed boundary-lattice Boltzmann method for simulating three-dimensional incompressible flows. *Journal of Computational Physics*, 229(13):5022–5042, 2010. pages 49

[152] Heng Xiao and Patrick Jenny. A consistent dual-mesh framework for hybrid LES/RANS modeling. *Journal of Computational Physics*, 231(4):1848–1865, 2012. pages 73

[153] Min Xu, Mingjun Wei, Tao Yang, and Young S. Lee. An embedded boundary approach for the simulation of a flexible flapping wing at different density ratio. *European Journal of Mechanics, B/Fluids*, 55:146–156, 2016. pages 42

[154] Sheng Xu and Z. Jane Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 216(2):454–493, aug 2006. pages 41, 88, 89

[155] Jianming Yang and Elias Balaras. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics*, 215(1):12–40, jun 2006. pages 35, 44

[156] Jianming Yang and Frederick Stern. A simple and efficient direct forcing immersed boundary framework for fluid-structure interactions. *Journal of Computational Physics*, 231(15):5029–5061, 2012. pages 39, 95

[157] Jianming Yang and Frederick Stern. A non-iterative direct forcing immersed boundary method for strongly-coupled fluid-solid interactions. *Journal of Computational Physics*, 295:779–804, 2015. pages 39

[158] T Ye, R. Mittal, H.S. Udaykumar, and W. Shyy. An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *Journal of Computational Physics*, 156:209–240, 1999. pages xvi, 16, 17

[159] Hai Zhuan Yuan, Xiao Dong Niu, Shi Shu, Mingjun Li, and Hiroshi Yamaguchi. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating a flexible filament in an incompressible flow. *Computers and Mathematics with Applications*, 67(5):1039–1056, 2014. pages 49, 195

[160] X.Q. Zhang, P. Theissen, and J.U. Schlüter. Towards simulation of flapping wings using immersed boundary method. *International Journal for Numerical Methods in Fluids*, 71(April 2012):522–536, 2013. pages 35, 37, 44

[161] Luoding Zhu, Guowei He, Shizhao Wang, Laura Miller, Xing Zhang, Qian You, and Shiaofen Fang. An immersed boundary method based on the lattice Boltzmann approach in three dimensions, with application. *Computers and Mathematics with Applications*, 61(12):3506–3518, 2011. pages 46

[162] Xiaojue Zhu, Guowei He, and Xing Zhang. An Improved Direct-Forcing Immersed Boundary Method for Fluid-Structure Interaction Simulations 1. *Journal of Fluids Engineering*, 136(4):040903, 2014. pages 41