LIÈGE université
Sciences Appliquées

Mesh adaptation techniques for the Particle Finite Element Method (PFEM) in 2D and application to flows around bodies crossing a free surface

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy (Ph.D) in Engineering Science

by

Romain FALLA

Supervisor: Vincent E. TERRAPON

Co-supervisor: Jean-Philippe PONTHOT

DOCTORAL COLLEGE IN AEROSPACE AND MECHANICS

JUNE 2024

Doctoral Thesis

Mesh adaptation techniques for the Particle Finite Element Method (PFEM) in 2D and application to flows around bodies crossing a free surface

Author:

Romain FALLA

Supervisors:

Pr. Vincent E. TERRAPON
Pr. Jean-Philippe PONTHOT

Members of the Jury:

Dr. Romain BOMAN (President)	from University of Liège
Pr. Vincent E. TERRAPON	from University of Liège
Pr. Jean-Philippe PONTHOT	from University of Liège
Dr. Stéphane DORBOLO	from University of Liège
Pr. Benoit SCHEID	from Université Libre de Bruxelles
Pr. Jean-François REMACLE	from Université Catholique de Louvain
Pr. Marcella CRUCHAGA	from University of Santiago, Chile

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

July 8, 2024

Acknowledgment

It was challenging to bring this work to completion, and it would not have been possible without the help of different people, that I would like to acknowledge. I first thank the WOLFLOW team, in the person of Benoît Scheid, Stéphane Dorbolo, Lionel Vincent, and Vincent Terrapon. The collaboration inside the WOLFLOW team was very fruitful and stimulating and led to the development and implementation of novel ideas, leveraging numerous synergies between simulations, experiments and mathematical developments. On a more practical extent, I also thank Romain Boman and Luc Papeleux for their support regarding code management and general technical support. For some PFEM-related aspects, I also thank the specially dedicated team in our lab, in the person of Jean-Philippe Ponthot, Billy-Joe Bobach, Romain Boman, Simon Février, Martin Lacroix, and Eduardo Fernandez. Through discussions and shared papers, they all have contributed, at different levels, to a better understanding of some PFEM challenges involved in this work. I also take this opportunity to thank Marco Lucio Cerquaglia for having developed the PFEM code I started with. Most importantly, I would like to address a huge thank to my main advisor Vincent Terrapon. Not only he provided a consequent help for the writing of both the PFEM article and the Ph.D. manuscript, but also pushed me forward in the different parts of my work. While being very demanding on the form, he also made relevant comments on the substance in multiple occasions, which enabled me to improve many parts of the work. Eventually, I thank my family for their support and patience. Finally, both the financial support of the Belgian Fund for Scientific Research under research project WOLFLOW (F.R.S.-FNRS, PDR T.0021.18) and of the School of Engineering of the University of Liège through the "Subside FSA fin de thèse" are gratefully acknowledged.

Abstract

Two-dimensional flows around cylinders and spheres (axisymmetric) rising at constant velocity and crossing a free surface, are analyzed numerically using the Particle Finite Element Method (PFEM). The overall transient regime is investigated, ranging from the free-surface deformation and the wake dynamics when the cylinder is below the initial free-surface level, to the interface crossing and the film drainage when it is above.

The PFEM is well suited to describe these different flow features. First, it is a Lagrangian method and it therefore naturally captures the surface deformation thanks to the nodal displacements. Moreover, it can handle topological changes through the combination of the Delaunay triangulation and the α-shape technique, respectively used to re-triangulate the cloud of fluid particles and to identify the fluid boundaries. In particular, the interface crossing and the subsequent drainage of the thin film above the cylinder/sphere can be robustly simulated because of these particular features of the method. Nonetheless, the method has limitations stemming from the α-shape technique used to identify the domain boundary. First, the traditional implementation of this α-shape technique in the PFEM is limited to uniform meshes, such that the use of the PFEM can be very expensive in terms of computing time when a fine mesh is required. Moreover, it introduces mass conservation errors due to the deletion of existing, or addition of new, fluid elements.

To overcome the limitations of the PFEM, a novel mesh adaptation algorithm is proposed. A local target mesh size is prescribed according to geometric and/or physics-based criteria and particles are added or removed to approximately enforce this target mesh size. Additionally, a new boundary recognition algorithm relies on the tagging of boundary nodes and a local α-shape criterion that depends on the target mesh size. The method allows thereby reducing mass conservation errors at free surfaces and improving the local accuracy through local mesh refinement, and simultaneously offers a new boundary tracking algorithm. The possible extension of the proposed algorithm to three-dimensional tetrahedral meshes is then considered theoretically. In particular, the problematic case of slivers, i.e., very flat tetrahedra that are not removed by the 3D Delaunay triangulation and that can strongly deteriorate the local accuracy of the solution, is discussed.

The novel mesh adaptation algorithm is tested on six two-dimensional validation cases. The first three cases, i.e., the flow around a (static, rotating, or oscillating) cylinder at Reynolds numbers below or equal to 200, the lid-driven cavity flow at Reynolds numbers of 100 and 400, and the flow around an impulsively started cylinder at a Reynolds numbers of 9500, do not feature a free surface and mainly illustrates the mesh refinement capability. The last three test cases consist in the sloshing problem in a reservoir subjected to forced oscillations, the fall of a 2D liquid drop into a tank filled with the same viscous fluid, and the rise of an impulsively started cylinder toward the free surface at constant velocity. These last three cases demonstrate the more accurate representation of the free surface and a corresponding reduction of the error in mass conservation.

Then, the novel algorithm is applied to the main case of interest: a 2D cylinder or a 3D axisymmetric sphere pulled out of a liquid bath at constant velocity and crossing the free surface. Different aspects of the physics are investigated, including the free-surface elevation and the total drag, as well as the boundary layer, wake, and film drainage dynamics. In particular, the dependencies of these flow features on the different flow parameters, i.e., the Reynolds and the Froude numbers, and geometrical parameters, i.e., the pool width and the release depth, are investigated in details. Comparisons with the literature, as well as with in-house experiments of a rising cylinder in oil, are performed. In particular, the latter highlights the limitation of the present two-dimensional approach to represent real three-dimensional cases, despite rather good agreement for sufficiently high cylinder aspect ratios.

Finally, a mathematical model is developed for the description of the drainage dynamics in the thin film at the cylinder/sphere apex. The model relies on the observation that the film thickness during interface crossing is almost uniform around the apex. Combining the radially integrated Navier-Stokes equations and an assumed velocity profile, the film model enables to describe the transition from the inertia-to-gravity to the viscous-to-gravity dominated regime. The model involves only one calibrated parameter. It is validated using different PFEM simulations of a cylinder or a sphere crossing the free

surface at constant velocity. In particular, it is found to predict qualitatively very well, despite small quantitative discrepancies, the variation of the film thickness for a given range of Froude and Reynolds numbers.

This work concludes with several perspectives for future work.

Contents

List of Figures

List of Tables

Nomenclature

Physical properties

g	acceleration of gravity	$[\mathrm{L\,T^{-2}}]$
ν	kinematic viscosity of the fluid	$[\mathrm{L^2\,T^{-1}}]$
μ	dynamic viscosity of the fluid	$[\mathrm{M\,L^{-1}\,T^{-1}}]$
ρ	density of the fluid	$[\mathrm{M\,L^{-3}}]$
σ	surface tension	$[\mathrm{M\,T^{-2}}]$
l_ν	viscous-gravity length scale	[L]

Flow variables

$\mathbf{u}$	velocity vector field	$[\mathrm{L\,T^{-1}}]$
$\mathbf{a}$	acceleration vector field	$[\mathrm{L\,T^{-2}}]$
p	scalar pressure field	$[\mathrm{M\,L^{-1}\,T^{-2}}]$
δ	boundary layer thickness	[L]
U	scaling velocity/ cylinder-sphere Velocity	$[\mathrm{L\,T^{-1}}]$
Fr	Froude number	[-]
Re	Reynolds number	[-]
We	Weber number	[-]
h	distance between im. body apex and highest point of the free surface	[L]
y_s	free-surface maximal elevation with respect to the free surface at rest	[L]
f	argument of the exponential in the definition of h	[-]
W	thickness of the wake after the interface crossing	[L]
C_D	Drag coefficient	[-]
C_L	Lift coefficient	[-]
f_e, f_S	excitation/shedding frequency (Von Karman vortex street)	$[\mathrm{T^{-1}}]$
St	Strouhal number (non-dimensional shedding frequency)	[-]
C_f	friction coefficient	[-]

Geometrical parameters

a	characteristic radius of the immersed body	[L]
D	characteristic diameter of the immersed body	[L]
d	initial immersed body depth	[L]

y_c	vertical position of the immersed body center	[L]
L	characteristic length or mesh size	[L]
W	pool half width	[L]
H	pool depth	[L]

Mesh parameters

α	threshold parameter for the α-shape technique	[L] or [-]
A	mesh element area	[L^2]
L^*	target mesh size	[L]
V	mesh element volume	[L^3]
γ	threshold for the removal of nodes	[-]
$N_{\mathrm{nodes,g}}$	number of vertices in the Euler characteristics	[-]
$N_{\mathrm{edges,g}}$	number of edges in the Euler characteristics	[-]
$N_{\mathrm{faces,g}}$	number of faces in the Euler characteristics	[-]
$N_{\mathrm{cells,g}}$	number of cells in the Euler characteristics	[-]
N_{nodes}	number of mesh nodes	[-]
$\hat{N}$	shape function	[-]
σ	nodal density	[L^{-2}] or [L^{-3}]
f_N	nodal fraction	[-]
λ	mesh distortion factor	[-]

Other symbols

ϵ_T	truncation error of partial differential equations	[case dep.]
AR	Aspect ratio	[-]
$\mathbf{M}$	mass matrix	[M] or [$\mathrm{M\,L}^{-1}$]
$\mathbf{K}$	viscous shearing matrix	[$\mathrm{M\,T}^{-1}$] or [$\mathrm{M\,L}^{-1}\,\mathrm{T}^{-1}$]
$\mathbf{D}$	discretized divergence-gradient operator matrix	[L^2] or [L]
$\mathbf{f}$	vector of body and surface forces	[$\mathrm{M\,L\,T}^{-2}$] or [$\mathrm{M\,T}^{-2}$]
$\mathbf{C}$	dynamical stabilization matrix	[$\mathrm{L}^2\,\mathrm{T}$] or [$\mathrm{L\,T}$]
$\mathbf{L}_\tau$	discrete stabilizing Laplacian operator matrix	[$\mathrm{M}^{-1}\,\mathrm{L}^4\,\mathrm{T}$] or [$\mathrm{M}^{-1}\,\mathrm{L}^3\,\mathrm{T}$]
$\mathbf{h}$	vector of stabilizing projected forces	[$\mathrm{L}^3\,\mathrm{T}^{-1}$] or [$\mathrm{L}^2\,\mathrm{T}^{-1}$]
τ^e_{PSPG}	elementary stabilizing time	[T]

Chapter 1

Introduction

The overall goal of this thesis is two-fold: 1) the development and implementation of a two-dimensional (2D) mesh adaptation algorithm for the particle finite element method (PFEM), and 2) the use of 2D numerical simulations based on the PFEM to investigate the flow physics of bodies exiting a liquid bath and crossing a free surface. This work has been performed within the frame of the research project WOLFLOW, funded by the Belgian FNRS, in collaboration with the teams of Dr. Dorbolo at the University of Liège and of Prof. Scheid at the Université Libre de Bruxelles. The novel algorithm has been implemented in an existing in-house 2D PFEM solver that was initially developed by Cerquaglia [1].

1.1 Flow around a body rising toward and crossing a free surface

The crossing of an object through a liquid-air interface and the subsequent drainage of the fluid are of particular importance, not only from a fundamental point of view, but also for many applications. In zoology, it could help to understand the physics associated with animals jumping out of the water [2] (e.g., dolphins, whales, etc.). Similarly, other less peaceful applications can also be thought of. From an industrial point of view, the problematic of pulling a body out of a liquid bath applies directly to the coating process. On a lighter note, a better understanding of the underlying physics could also help choosing the optimal bread size and extraction velocity when eating a good Swiss cheese fondue.

Main physical aspects

The problem of a body crossing a liquid-air interface includes several interesting aspects and can be divided into two different dynamical phases:

- The first phase consists in the body approaching the free surface. One interesting aspect during this phase is the surge dynamics of the fluid above the body, and the free-surface deformation it provokes. If the body is initially at rest, this first phase also includes the development of a wake behind the body and the associated drag force. Depending on the initial distance to the free surface and acceleration of the body, the wake might still be developing when the body reaches the free surface.

- The second phase consists in the actual crossing of the free surface. The start of this phase can be for instance defined as the moment when the body apex reaches the initial position of the free surface. During this phase, liquid from the bath is entrained by the body. Subsequently, the film of liquid around the body thins by drainage. A schematic illustration of these two distinct phases and the associated physical processes is shown in Fig. 1.1.

If surface tension can be neglected (for instance when the free-surface curvature is not strongly varying along the free surface, or simply when the surface tension is low), the physics of the problem is mostly controlled by the Reynolds number (ratio of inertia to viscous forces) and Froude number (ratio of inertia to gravity). Their respective influence on the aforementioned physical processes is now briefly discussed.

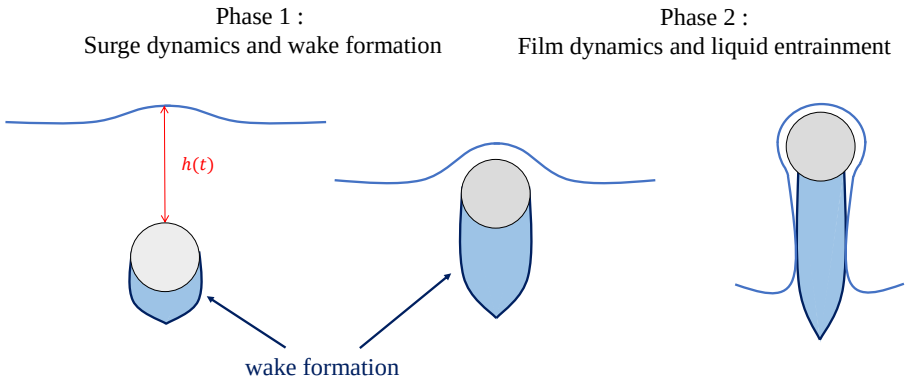


Figure 1.1: Schematic of the two phases of the free-surface crossing of a body (a cylinder here) before (left), during (middle) and after (right) the interface crossing. The free surface is represented by the blue line above the body and the shaded blue region indicates the wake.

Surge dynamics above the body

The first feature that can be easily observed during the initial phase is the free-surface deformation. In the present context, a key question is therefore how the deformation of the free surface is influenced by different parameters, such as the initial depth of the body, its velocity, gravity, the viscosity of the fluid, etc. Because of pressure forces, the free surface above a body moving up towards it rises. Intuitively, it can be expected that the amplitude of the deformation increases with the body velocity, as the pressure also increases. On the other hand, gravity plays the opposite role, and tends to flatten the free surface. These competing effects can be quantified by the Froude number

$$Fr = \frac{U^2}{gL}\,, \tag{1.1}$$

where U is the body velocity, g the gravitational acceleration and L a characteristic length representative of the body size. As shown later Fr is the main non-dimensional parameter controlling the surge dynamics. In particular, increasing the Froude number leads to a larger vertical displacement of the free surface. The influence of the other non-dimensional parameter, i.e., the Reynolds number, on the surge dynamics, is expected to be less significant and more indirect. It mostly controls the wake which can be viewed as some extension of the body shape and has thus some limited impact on the amplitude of the free-surface deformation.

Overall drag and wake dynamics behind the body

The literature on free-surface flows is mainly based on potential flow theory, which assumes an inviscid fluid. The interactions between viscous boundary layers, or wakes behind submerged bodies, and a free surface, are not well understood. In particular, one challenge is to identify how the combined effects of viscosity, the free surface at constant atmospheric pressure, and gravity influence the overall drag on the rising body. It will be shown that both inviscid and viscous contributions to the drag come into play, and that the inviscid contribution mainly depends on the Froude number. Moreover, the largest contribution to the drag results from friction on the body surface (direct viscous effect) and the depression in its wake (indirect viscous effect). This viscous drag is related to the Reynolds number,

$$Re = \frac{UL}{\nu}\,, \tag{1.2}$$

where ν is the kinematic viscosity.

Film drainage above the body

When the body is very close to the (deformed) free surface, the liquid between the free surface and the body surface starts to take the form of a thick film that becomes thinner as the body rises. At

small Reynolds number, the viscosity combined with the no-slip boundary condition at the body surface impacts the dynamics of the film, which is expected to thin more slowly (slower drainage) at smaller Reynolds number. At larger Reynolds number, however, the effect of Re on the film dynamics remains small until the film becomes very thin. The dependency on the Froude number is more complex. At some times, the velocity of fluid above the body is expected to be of the same order as that of the body, such that the velocity difference is mainly due to gravitational drainage, and the local Froude number ($\frac{U_g^2}{gL}$ where $U_g = \sqrt{gL}$) is one in all cases. However, the Froude number still matters insofar as the fluid around the body (other than that contained in the film) and the history of the previous surge dynamics has still an impact on the film dynamics, when the body is crossing the free surface. One key objective of this work is thus to provide a more quantitative view on how the Reynolds and the Froude numbers influence the film dynamics above the body.

Liquid entrainment below the body after the interface crossing

Lastly, when the body has crossed the free surface, it is followed by its wake and its associated momentum. The quantity of fluid entrained is expectedly a function of the Reynolds number, but also of the release depth, as the wake dynamics evolve during the rise of the body. The Froude number has an impact on the amount of entrained liquid, insofar as the acceleration of gravity has more or less time to decrease the momentum gained in the body's wake.

1.2 Simplified models for free-surface flows

The common characteristic of all free-surface flows is that they involve deformable boundaries or interfaces, leading to unsteady nonlinear boundary conditions. The most well-known resulting feature of this is the propagation of gravity waves, for which several researchers, dating back as far as Newton, Laplace and Lagrange, have attempted to build a theory [3]. Although an exhaustive review of the developments in that field is not given here, a few landmark works can briefly be cited. Note that a summary of the different models and their domain of validity can be found in the book of Bernard Méhauté [4].

- The standard linear theory was proposed by Airy in 1841, as explained in Craik [3]. It is based on the approximation that the free-surface elevation is negligible with respect to the water depth H, such that the free-surface boundary condition is imposed at the initial position $z = 0$ (i.e., the height of the free surface at rest), eliminating the nonlinearities. The linear approximation gives accurate results for $\frac{A}{H} << 1$ and $H/\lambda << 1$ (shallow water) or for $\frac{A}{\lambda} << 1$ and $H/\lambda >> 1$ (deep water), where A and λ are the amplitude and wavelength of the free-surface motion.

- Later, Stokes introduced a Taylor expansion [5] of the equations around the initial free surface (i.e., at $z = 0$) to deal with the nonlinearities inherent to the free-surface motion. Depending on the order of the expansion, the range of validity of the method can be extended, but the method is always better suited for deep water waves ($H/\lambda >> 1$).

- Finally, Boussinesq also developed nonlinear wave equations to describe shallow water waves, followed by Korteweg and De vries [5], who have rediscovered an approximation of Boussinesq equations. The equations are also derived from a Taylor expansion of the nonlinear equations, but starting from the bottom position $z = -H$.

These works are limited to water wave propagation and do not enable the description of the interaction between a free surface and a body. As discussed in Chapter 5, several works have attempted to describe this interaction, but they are mostly based on potential flow theory and thus neglect viscous effects. However, the presence of viscous terms are required to describe phenomena such as the free surface deformation of a viscous fluid, the interaction between the wake of a moving body and the free surface, etc. The inclusion of these additional physical flow characteristics renders the derivation of analytical solutions almost impossible, and the use of simulations is therefore unavoidable to study in detail such applications.

1.3 Numerical methods for free-surface flows and moving bodies

From a numerical point of view, simulating this type of flows involves several challenges. First, it needs to account for the displacement of the deformable interfaces (for instance, the free surface in the present case). Particularly, the identification of these interfaces is often subjected to mass conservation errors, which depend on the considered numerical method. Another challenge is the description of bodies moving within the computational domain and the corresponding fluid-structure interactions. At the simplest level, this might be a rigid body whose displacement is imposed. Alternatively, there might be a two-way coupling between the flow and the motion of the rigid body, which is governed in this case by Newton's second law. At the other hand of the spectrum, the body might even deform in response to the stress at its surface, so that a structural model is also needed to determine this deformation. Note that the present work only considers the first case, i.e., the imposed motion of a rigid body, so that only the fluid phase needs to be calculated. Finally, the last challenge resides in the potential large range of scales present in such flows. For instance, the fluid is expected to barely move away from the body so that the solution is rather smooth in large regions of the domain. On the other hand, the boundary layers around, the wake below, and the thin film above the rising body involve very small scales that need to be captured. A continuous mesh adaptation capability is thus required to efficiently simulate such flows.

Within this context, existing methods can be divided into two groups:

- In **Eulerian methods** the flow variables are defined at points that are not moving or that are moving in arbitrary manner. For these methods, the difficulty mostly lies in the fact that moving physical boundaries do not coincide with the initial mesh boundaries, and special techniques must be used to take this into account [6].
- In **Lagrangian methods** the flow variables are defined at moving fluid particles. These methods have the advantage of naturally taking into account the displacements of the moving boundaries, as they depend on the solution or the boundary conditions. They are however subjected to other limitations, as explained hereafter.

Volume of fluid method (VOF)

In the Eulerian framework, the method of choice for simulating multiphase and free-surface flows is the Volume of Fluid (VOF) method, which consists in accounting for the fraction of fluid in interface cells [7]. This is often coupled with a boundary tracking algorithm, such as the Piecewise linear interface-capturing (PLIC) method [8] or the level-set method [9], which are used to track the boundary on top of the background Eulerian mesh. One advantage on this method is that it can easily handle topological changes. Another advantage is that the mesh does not change, unless some adaptive mesh refinement is used, such that the mesh quality can be optimized in a way that the space discretization errors due to mesh irregularities are minimized. However, the use of a boundary tracking algorithm also involves approximations and, therefore, leads to numerical errors near the free surface, and in particular to errors related to mass conservation. Furthermore, the regions where there is initially no fluid (e.g., when the gas phase over the free surface of a liquid is neglected) need also to be meshed if fluid is expected to be present at later stages of the simulation. Lastly, accounting for moving bodies with large displacements requires special algorithms, such as the immersed boundary [10] or the overset mesh [11] techniques, which are not easy to implement in combination with the VOF method and introduce new challenges.

Arbitrary Lagrangian-Eulerian (ALE) method

In the Eulerian context, another approach relies on the arbitrary Lagrangian - Eulerian (ALE) description [12]. Specifically, the displacement of the domain boundaries, due to a moving body or a deforming free surface, is accommodated through the deformation of the mesh. The boundary displacement can either be imposed or result from the solution. Based on the boundary displacement, the internal mesh is then deformed using different possible techniques [13] (e.g., linear spring analogy, linear elasticity, Laplace smoothing, interpolation techniques). The main advantage of this approach is that

the mesh remains conformal with the boundaries. However, only relatively small mesh deformations can be considered to avoid a rapid deterioration of the mesh quality. For larger deformations, a costly remeshing is required. Furthermore, the method precludes any topological change of the computational domain, such as merging of fluid regions, fluid detachment, interface folding, etc. The ALE formulation with mesh deformation is a typically used in FSI problems.

EXtreme Mesh deformation approach (X-MESH)

The extreme mesh deformation approach [14] can be seen as a special case of ALE. It enables the mesh to follow sharp interfaces of the fluid that are possibly subjected to topological changes (fluid merging, spitting,...), without changing the mesh topology (i.e., without the need to remesh). This is possible by allowing some mesh elements to locally reach zero measure at the level of the interfaces. The main idea is to keep the mesh nodes coincident with the physical interface (using some boundary tracking algorithm) until some adjacent elements become crushed, in which case other mesh nodes take over to coincide with the physical interface. The difficulty here is to keep a stable time integration scheme and a good spatial accuracy while using very distorted elements, which requires some special treatment.

Smoothed Particle Hydrodynamics (SPH)

Unlike in the Eulerian formulation, in Lagrangian approaches the displacement of boundaries is directly obtained from the solution. The main challenge stems from the fact that the particle motion is not arbitrary but depends on the solution everywhere in the domain. If a mesh is used, it can be subjected to strong mesh distortions so that regular time-consuming remeshing might be needed. To avoid this, Lagrangian mesh-less methods can be considered, such as the Smoothed Particle Hydrodynamics (SPH). In SPH the equations are locally averaged at position $\mathbf{x}$ over a stencil of arbitrary size h based on the variables defined at the particles. For mathematical details on the method, the reader is referred to Monaghan et al. [15] and Lucy et al. [16], who pioneered this method initially introduced in the context of astrophysics. A more exhaustive review of the method and its recent challenges can be found in Vacondio et al. [17]. A major challenge in SPH is the difficulty to impose boundary conditions, which arises from the meshless nature of the method. A second challenge concerns the consistency, stability and convergence behavior of SPH schemes, which are not as robust as in the finite element method. For instance, stability and convergence depend strongly on the particle distribution [17]. Another challenge is related to the low adaptivity of the method (i.e., the capacity of the method to use spatial discretisation of different resolutions at different locations of the computational domain). Most SPH codes are based on a uniform resolution and are therefore not efficiently applicable to many multi-scale engineering problems, even if some recent efforts have been devoted to enabling the use of non-uniform particle distributions [18–21].

The particle Finite Element Method (PFEM)

Unlike the SPH, the PFEM is a Lagrangian method that relies on a mesh. In this case, frequent remeshing following large mesh distortion is not avoided but efficiently performed through a fast Delaunay triangulation [22, 23]. The imposition of boundary conditions is thus much easier than with SPH, and similar to what is done in a classical finite element approach (with the exception of the free-slip boundary condition, in particular on curved boundaries [24]). Another strength of the PFEM is that it takes advantage of the remeshing to allow topology changes so as to accomodate fluid separation and merging. This feature results from the boundary recognition algorithm, based on the α-shape technique [25].

The properties of the PFEM, and in particular its natural treatment of changes in the topology and boundaries of the fluid domain and the robustness of the underlying finite element method, make it an ideal candidate to simulate the flow around bodies crossing a free surface. Consequently, the numerical simulations and analysis presented in this work rely on the PFEM (note that other methods could have also been considered). However, as discussed in Chapter 2 where the method is described in detail, the

classical PFEM also suffers from several limitations. These shortcomings mostly stem from the α-shape technique used for boundary recognition. First, and similarly to many other methods for multiphase flows, the conservation of mass is not strictly enforced. Second, the α-shape technique as classically used in the PFEM presupposes an approximately uniform mesh size, which makes it inefficient in cases where some regions of the flow require a higher resolution than others.

1.4 Objectives of the thesis

The overall goal of this work is to investigate numerically the underlying physics of the flow around a body pulled out of a liquid bath. For the sake of simplicity, the scope is here restricted to a rigid 2D cylinder or an axisymmetric sphere. Because of the deforming free surface and potential changes in topology, the PFEM is chosen as the method of choice. Nevertheless, in such flows, the mesh size requirements can greatly vary from one region of the domain to another. For large tanks for instance, the fluid close to the tank walls moves very little and the solution is smooth so that a coarse mesh can be used. On the other hand, the viscous boundary layer along the moving body surface, the wake behind it, the free surface in the crossing region or the thin liquid film around the body after crossing the interface, are regions characterized by very small scales and require thereby a high mesh resolution. Using the same grid resolution everywhere, including in a large region of the computational domain where it is not needed, would induce an excessive computational cost. It is thus necessary to adapt the classical PFEM for strongly non-uniform meshes.

This leads to the two specific objectives of this work:

1) Develop and implement a 2D mesh adaptation algorithm in an existing in-house PFEM solver to improve the efficiency of the method for applications of practical use and reduce the error in mass conservation.

2) Simulate the flow around a cylinder/sphere pulled out of a liquid bath and investigate the underlying physics of the process. In particular, the following key questions should be adressed:

- How is the free-surface deformation influenced by the main flow parameters? How does its time evolution vary with the Froude number?
- What is the effect of the free surface and gravity forces on the total drag of the rising body, and how does it compare with similar flows without free surface?
- After the interface crossing, which key flow parameters govern the drainage of the film above the cylinder/sphere? How can this be predicted based on the main flow parameters?
- What is the overall effect of viscosity on the surface elevation? What are the interactions between the wake and the free-surface deformations?

The main contributions of this work thus consist in both the further development of the PFEM method through a new mesh adaptation capability and the better understanding of the flow physics for bodies crossing a free surface. The main idea behind the novel mesh adaptation algorithm relies on

- the definition of a local target mesh size L^* based on different geometrical and solution-based quantities,
- the addition and removal of nodes, so as to locally approximately enforce the target mesh size L^*,
- the boundary tracking and a local version of the α-shape technique.

For the investigation of the underlying physics, numerical simulations leveraging the new mesh adaptation algorithm are performed in order to identify the different flow features previously discussed in Section 1.1. In particular, the following results, among others, are shown and discussed in details:

- The key parameter controlling the surface deformation is the Froude number, and its impact is quantified.
- The deviation of the total drag from that obtained in similar flows without free surface, is also found to mainly depend on the Froude number. In particular, the drag with free surface is larger at small Froude number, but lower at large Froude number.
- The wake is shown to have a secondary impact on the free-surface deformation, suggesting only an indirect effects of the Reynolds number and release depth.

Finally, a mathematical model, whose derivation is based on radially averaging the flow equations, is proposed to predict the film thickness during the drainage phase. This model is calibrated and validated using different PFEM simulations, i.e., using different combinations of the flow parameters Re and Fr.

1.5 Outline of the thesis

This manuscript is divided into seven chapters.

- After this introduction, Chapter 2 first presents in details the Particle Finite Element Method and its formulation implemented in the in-house code used in the present work [1]. The boundary recognition algorithm classically used in the PFEM, i.e., the α-shape technique, is then described and the associated problem of mass conservation inherent to the PFEM is discussed and illustrated.
- The novel 2D mesh adaptation algorithm (definition of the target mesh size, addition and removal of particles, boundary tagging) is extensively explained in Chapter 3. This chapter also highlights how the algorithm can both enable the use of a non-uniform mesh and reduce the error in mass conservation. A new theoretical framework for extending the 2D mesh adaptation and boundary recognition techniques to three dimensions is then introduced. More specifically, a generalization of the notion of (target) mesh size to a three-dimensional (target) nodal density for use in three-dimensional unstructured tetrahedral meshes is proposed.
- The implementation of the new mesh adaptation algorithm is then tested in Chapter 4 through six different validation cases. The first three cases consist in flows inside a bounded domain, and mainly address the mesh refinement/coarsening capabilities. These are the flow around a (static, vertically oscillating or rotating) cylinder at $Re_D \leq 200$, the flow around an impulsively started cylinder at $Re_D = 9500$, and the flow in a lid-driven cavity. The last three cases consist in free-surface flows and assess both the refinement/coarsening capabilities and the improved boundary recognition algorithm. The main focus of these test cases is the accurate representation of the free surface and the reduction of mass conservation errors. Specifically, these are the sloshing of water in a rotating reservoir harmonically excited, the fall of a drop into a bath of the same viscous fluid, and the rise of a cylinder toward a free surface. For this last case, the performance in terms of CPU time of the new algorithm is evaluated, demonstrating the applicability of the method to simulate flows in large domains.
- Chapter 5 focuses on the application of the PFEM and the new mesh adaptation algorithm to the case of a flow around a rigid cylinder/sphere pulled out of a bath at constant velocity. First, the literature on free-surface flows interacting with cylinders is briefly summarized. This is followed by an introduction to the specific case of a 2D cylinder rising toward the free surface. Afterwards, mesh and time steps convergence analyses are presented to determine the required accuracy of the space and time discretization. As aforementioned, different aspects of the flow physics are then addressed, including the influence of the Froude number on the free-surface elevation and overall drag, the interactions between wake and free surface, the influence of other flow and geometrical parameters, etc. These features are investigated comparing the PFEM simulations with existing results from the literature (numerical or experimental), and/or with in-house experiments of a rising cylinder in oil. Finally, the rise of an axisymmetric sphere is also considered to highlight the main differences compared to the cylinder.

- Chapter 6 introduces a low-order model for the estimation of the dynamics of the flow between the rising body and the free surface. This model assumes a medium to low film thickness above the body so that the equations can be averaged radially. In a first step, the model is formally derived for both a cylinder and an axisymmetric sphere. In a second step, the calibration of the single model parameter is performed using PFEM simulations. Finally, the thin film model is tested to check its ability to predict the thickness variation at different Reynolds and Froude numbers.
- Finally, Chapter 7 provides a global conclusion and perspectives of the work, structured according to the two main objectives described above. First, the limitations of the new mesh adaptation algorithm are discussed, followed by an introduction to potential improvements, in particular regarding mass conservation, computing efficiency and extension to 3D. At last, avenues for future research regarding the flow physics of bodies crossing a free surface and potential improvements for the low-order models are proposed.

Chapter 2

The Particle Finite Element Method (PFEM)

The purpose of this chapter is to present the PFEM and discuss the associated challenges, including the extension to non-uniform meshes and the problematic mass conservation errors inherent to the boundary recognition algorithm. First, the equations to be solved are introduced and their discretization using the finite element method (FEM) is formally presented. In this context, the main challenge associated with the violation of the so-called Ladyzhenskaya–Babuška–Brezzi (LBB), or inf-sup, condition, for the resolution of saddle point problems and the resulting need for stabilization are highlighted. In a second step, the boundary recognition algorithm based on the α-shape technique and its underlying assumption of a regular mesh are discussed. At last, the central role of the α-shape technique in the problem of mass conservation is explained and illustrated through simple examples.

2.1 The unstable discrete form of the incompressible Navier-Stokes equations

The PFEM relies on a finite element discretization of the equations of motion. In this work, the focus is on the incompressible Navier-Stokes equations for a Newtonian viscous liquid with constant properties given by

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mu \nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \rho \mathbf{b} \quad \text{in} \quad \Omega(t)\,, \tag{2.1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in} \quad \Omega(t)\,, \tag{2.2}$$

where p is the pressure scalar field, $\mathbf{u}$ the velocity field, ρ the density and μ the dynamic viscosity[1] of the fluid inside the (possibly deforming) domain $\Omega(t)$, and $\mathbf{b}$ is the vector containing the body forces. These equations are complemented with some kinematic and dynamic boundary conditions on the boundaries Γ_k and Γ_σ, respectively,

$$\mathbf{u}(\mathbf{x}, t) = \overline{\mathbf{u}}(\mathbf{x}, t) \quad \text{on} \quad \Gamma_k\,, \tag{2.3}$$

$$\sigma \cdot \mathbf{n} = \overline{\mathbf{t}} \quad \text{on} \quad \Gamma_\sigma\,, \tag{2.4}$$

where $\overline{\mathbf{u}}$ and $\overline{\mathbf{t}}$ are imposed velocities and surface tractions, $\mathbf{n}$ is the unit outward normal to the boundary, σ is the stress tensor and $\overline{\mathbf{t}}$ is the vector of surface forces (including in particular surface tension). Additionally, one has that $\Gamma_k \cap \Gamma_\sigma = \varnothing$ and $\overline{\mathbf{t}}$ $\Gamma_k \cup \Gamma_\sigma = \Gamma$, the complete boundary of the domain $\Omega(t)$.

[1] The viscous term in Eq. (2.1) could be rewritten in terms of the Laplacian of the velocity. Although analytically equivalent, the two formulations usually differ in their discretized form. The one in Eq. (2.1) is chosen for consistency with Cerquaglia [1].

To formulate the weak forms of the equations, trial functions $\mathbf{u}$ and p, representing the velocity and pressure fields, and some test functions $\mathbf{w}$ and q, are considered as being parts of the following spaces,

$$\mathcal{S} = \{\mathbf{u} \in \mathbf{H}^1(\Omega) | \mathbf{u} = \overline{\mathbf{u}} \quad \text{on} \quad \Gamma_k\}\,, \tag{2.5}$$

$$\mathcal{S}_0 = \{\mathbf{w} \in \mathbf{H}^1(\Omega) | \mathbf{w} = \mathbf{0} \quad \text{on} \quad \Gamma_k\}, \tag{2.6}$$

$$\mathcal{Q} = \{p, q \in L^2(\Omega)\}\,, \tag{2.7}$$

where $L^2(\Omega)$ is the space of square integrable functions and $\mathbf{H}^1(\Omega)$ is the space of square integrable functions with square integrable first derivative, both over the domain Ω. The weak forms of Eqs. (2.1) and (2.2) can be obtained by multiplying them by the test functions $\mathbf{w}$ and q, respectively, and integrating over the domain Ω:

$$\int_\Omega \rho \frac{D\mathbf{u}}{Dt} \cdot \mathbf{w}\, d\Omega = \int_\Omega -\nabla p \cdot \mathbf{w}\, d\Omega + \int_\Omega \mu \nabla \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \mathbf{w}\, d\Omega + \int_\Omega \rho \mathbf{b} \cdot \mathbf{w}\, d\Omega\,, \tag{2.8}$$

$$\int_\Omega q \nabla \cdot \mathbf{u}\, d\Omega = 0\,. \tag{2.9}$$

Integrating by parts and applying Gauss theorem leads to the final form

$$\begin{aligned} &\int_\Omega \rho \frac{D\mathbf{u}}{Dt} \cdot \mathbf{w}\, d\Omega = \int_\Omega p\, \nabla \cdot \mathbf{w}\, d\Omega - \int_\Omega \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) : \nabla \mathbf{w}\, d\Omega + \int_\Omega \rho \mathbf{b} \cdot \mathbf{w}\, d\Omega - \int_\Gamma p \mathbf{w} \cdot \mathbf{n}\, d\Gamma \\ &+ \int_\Gamma \mu\, \mathbf{w} \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \mathbf{n}\, d\Gamma\,, \end{aligned} \tag{2.10}$$

$$\int_\Omega q \nabla \cdot \mathbf{u}\, d\Omega = 0\,, \tag{2.11}$$

where the last two terms of the first equation above are simply the integral of surface traction over the domain boundary:

$$\begin{aligned} -\int_\Gamma p \mathbf{w} \cdot \mathbf{n}\, d\Gamma + \int_\Gamma \mu\, \mathbf{w} \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \mathbf{n}\, d\Gamma = -\int_{\Gamma_\sigma} p \mathbf{w} \cdot \mathbf{n}\, d\Gamma_\sigma + \\ \int_{\Gamma_\sigma} \mu\, \mathbf{w} \cdot (\nabla \mathbf{u} + \nabla \mathbf{u}^T) \cdot \mathbf{n}\, d\Gamma_\sigma = \int_{\Gamma_\sigma} \overline{\mathbf{t}} \cdot \mathbf{n}\, d\Gamma_\sigma\,. \end{aligned} \tag{2.12}$$

The exact solution of the original system of PDEs, Eqs. (2.1) and (2.2), can be obtained if Eqs. (2.8) and (2.9) are satisfied for every possible test functions $\mathbf{w}$ and q. However, in practice, the finite element approximation results from the choice of some arbitrary trial and test functions. Their number is based on the number of degrees of freedom of the system (to close the discrete system of equations) and their shape depends on other parameters related to both the accuracy of the space discretization and the stability properties of the time integration. As for the present case, for a linear finite element formulation of the incompressible Navier-Stokes equations, the degrees of freedom simply correspond to the pressure and velocity variables defined at the nodes. Therefore, in two dimensions, three test functions $\mathbf{w}_x^I$, $\mathbf{w}_y^I$ and q^I are used respectively to get an equation for the two velocity components and the pressure at each node I, as illustrated in Fig. 2.1(left). Moreover, linear shape functions $\hat{N}^J(\mathbf{x})$ are then introduced for the velocity and pressure fields, as illustrated in Fig. 2.1(right), such that for each finite element, one has

$$u_i(\mathbf{x}) = \hat{N}^I(\mathbf{x}) U_i^I\,, \tag{2.13}$$

$$p(\mathbf{x}) = \hat{N}^I(\mathbf{x}) P^I\,, \tag{2.14}$$

where i and I are respectively the space direction and the shape function (or node) indices, and U_i^I, P^I are the velocity and pressure variables at the nodes (note that the Einstein summation convention has been used). In this work, the same shape functions are used for both the velocity and pressure fields. This choice can cause difficulties with saddle point problems like this one, so that a stabilization is required, as discussed later. Within a standard Galerkin formulation, the test functions are the same

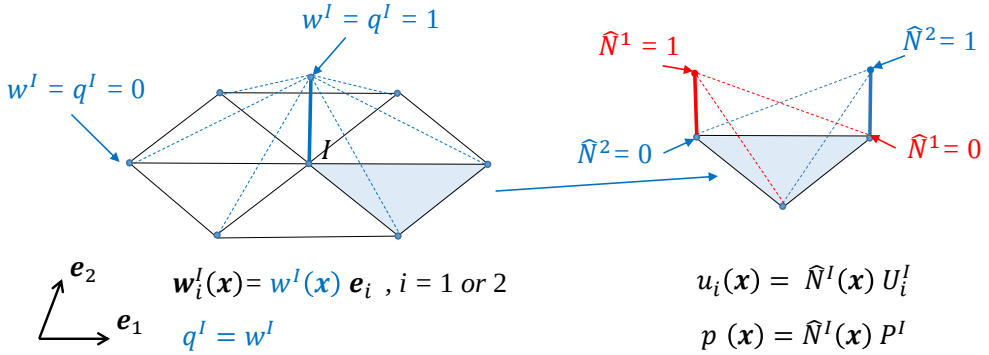


Figure 2.1: Diagrams of the linear test functions (left) and trial functions (right) used for the standard Galerkin formulation of the incompressible Navier-Stokes equation. The values of the test and trial functions is schematically represented using an altitude in the third dimension.

as the trial functions (i.e., they are made of the same shape functions, as illustrated in Fig. 2.1), such that the equations for an element e become (in index notation):

$$\int_e \rho \hat{N}^I \hat{N}^J d\Omega_e \frac{D\,U_i^J}{Dt} = \int_e \frac{\partial \hat{N}^I}{\partial\, x_i} \hat{N}^J\, d\Omega_e\, P^J - \int_e \mu \frac{\partial \hat{N}^I}{\partial\, x_j} \frac{\partial \hat{N}^J}{\partial\, x_j}\, d\Omega_e\, U_i^J - \int_e \mu \frac{\partial \hat{N}^I}{\partial\, x_j} \frac{\partial \hat{N}^J}{\partial\, x_i}\, d\Omega_e\, U_j^J$$
$$+ \int_e \rho b_i \hat{N}^I\, d\Omega_e + f_i - \int_e \hat{N}^I \frac{\partial \hat{N}^J}{\partial\, x_i}\, d\Omega_e\, U_i^J = 0\,, \tag{2.15}$$

where f_i represents either the element-element interactions or the boundary surface traction. The former cancels when the global matrices are built (according to the action-reaction principle), while the latter is simply integrated over the respective boundary line elements.

Using Eq. (2.15), the elementary matrices and elementary vectors are, in two dimensions, defined in the following way:

$$(6 \times 6) \quad M^e_{3i+I,3j+J} = \int_e \rho \hat{N}^I \hat{N}^J d\Omega_e\, \delta_{ij}\,, \tag{2.16}$$

$$(3 \times 6) \quad D^{e,T}_{3i+I,J} = -\int_e \frac{\partial \hat{N}^I}{\partial\, x_i} \hat{N}^J d\Omega_e\,, \tag{2.17}$$

$$(6 \times 6) \quad K^e_{3i+I,3j+J} = \int_e \frac{\partial \hat{N}^I}{\partial\, x_j} \frac{\partial \hat{N}^J}{\partial\, x_i} d\Omega_e (1 + \delta_{ij})\,, \tag{2.18}$$

$$(6) \quad f^e_{3i+I} = \int_e \rho b_i \hat{N}^I\, d\Omega_e + \int_{\Gamma_{\sigma,e}} \hat{N}^J\, \bar{t}_i^J\, d\Gamma_{\sigma,e}\,, \tag{2.19}$$

where $\Gamma_{\sigma,e}$ is the part of Γ_σ that belongs to the element. Global matrices can then be built from these elementary matrices to obtain the semi-discrete equations:

$$\mathbf{M}\frac{D\mathbf{u}}{Dt} + \mathbf{K}\mathbf{u} + \mathbf{D}^T\mathbf{p} = \mathbf{f}_{\text{ext}}\,, \tag{2.20}$$

$$\mathbf{D}\,\mathbf{u} = 0\,, \tag{2.21}$$

where $\mathbf{u}$ is a vector of size $2N$ ($3N$ in three dimensions) containing velocity components and $\mathbf{p}$ a vector of size N containing the pressure at the N nodes of the mesh. In the above equations, $\mathbf{M}$ is the mass matrix, $\mathbf{K}$ the viscous matrix, $\mathbf{D}$ the discrete divergence operator, $\mathbf{D}^T$ the discrete gradient operator, and $\mathbf{f}_{\text{ext}}$ the vector of external forces. Finally, the time derivative is discretized using in the present work a backward Euler scheme with time step size Δt to obtain the final matrix equation for the unknown nodal values at time step $n+1$,

$$\begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}^{n+1} + \mathbf{K}^{n+1} & \mathbf{D}^{n+1,T} \\ \mathbf{D}^{n+1} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}^n\mathbf{u}^n + \mathbf{f}_{\text{ext}}^{n+1} \\ \mathbf{0} \end{pmatrix}. \tag{2.22}$$

The Ladyzhenskaya–Babuška–Brezzi (LBB) condition:

The system of equations (2.22) has a saddle point structure, characterized by a square matrix $\mathbf{A}$ in the top left, a zero block matrix in the bottom right and rectangular matrices $\mathbf{B}$ and $\mathbf{B}^T$ in the remaining quadrants:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} . \tag{2.23}$$

By identification with Eq. (2.22), it follows that $\mathbf{A} = \frac{1}{\Delta t}\mathbf{M}^{n+1} + \mathbf{K}^{n+1}$, $\mathbf{B} = \mathbf{D}^{n+1}$, etc... The name "saddle-point" refers to the fact that solving the system (2.22) amounts to finding the saddle-point $(\mathbf{u}^*, \mathbf{p}^*)$ of a Lagrangian function $\mathcal{L}(\mathbf{u}, \mathbf{p})$ [26], i.e. ,

$$\mathcal{L}(\mathbf{u}, \phi) \leq \mathcal{L}(\mathbf{u}^*, \mathbf{p}^*) \leq \mathcal{L}(\mathbf{w}, \mathbf{p}) \quad \forall \mathbf{w} \in \mathcal{W}_h \quad \forall \phi \in \mathcal{Q}_h \tag{2.24}$$

where $\mathcal{W}_h$ and $\mathcal{Q}_h$ are two vector spaces containing the velocity and pressure variables $\mathbf{u}$ and $\mathbf{p}$, respectively. Such problems are challenging to handle because they can lead to instabilities, which in this case translates into spurious pressure oscillations. In this respect, the LBB condition, also called "inf-sup" condition, is a sufficient condition to have a stable scheme when inverting the system, i.e., for the system to have a unique solution that depends continuously on the boundary and initial conditions [27].

To illustrate this, one can consider spaces of infinite dimensions. Let $\mathcal{W}$ and $\mathcal{Q}$ be two Hilbert spaces, let $a : \mathcal{W} \times \mathcal{W} \to \mathbb{R}$ and $b : \mathcal{W} \times \mathcal{Q} \to \mathbb{R}$ be bilinear forms. Let $f \in \mathcal{W}^*$ and $g \in \mathcal{Q}^*$ where $\mathcal{W}^*$ and $\mathcal{Q}^*$ are the dual spaces. The saddle-point problem for the pair (a, b) corresponds to finding a pair of fields u in $\mathcal{W}$, p in $\mathcal{Q}$ such that, for all w in $\mathcal{W}$ and q in $\mathcal{Q}$,

$$a(u, w) + b(w, p) = \langle f, w \rangle , \tag{2.25}$$

$$b(u, q) = \langle g, q \rangle , \tag{2.26}$$

where $\langle f, w \rangle$ is the scalar product between f and w (same for $\langle g, q \rangle$). The LBB conditions states that a unique solution to this saddle point problem can be obtained as soon as

$$\exists \beta > 0 \ , \ \inf_{q \in \mathcal{Q}, q \neq 0} \sup_{w \in \mathcal{W}, w \neq 0} \frac{b(w, q)}{||w||_{\mathcal{W}} ||q||_{\mathcal{Q}}} \geq \beta . \tag{2.27}$$

If this condition is not satisfied, then the solution may not be unique and may be unstable. The condition can be rewritten removing the lower bound, being therefore valid for all q, in the form

$$\exists \beta > 0 \ , \ \forall q \in \mathcal{Q} \ , \ q \neq 0 \ , \ \sup_{w \in \mathcal{W}, w \neq 0} \frac{b(w, q)}{||w||_{\mathcal{W}}} \geq \beta ||q||_{\mathcal{Q}} . \tag{2.28}$$

The mathematics behind the LBB condition is rather complex but explains why spurious pressure oscillations may happen when the condition is not satisfied. In the particular case here, the discrete version of the second term of Eq. (2.25), $b(w, p)$, corresponds to $\mathbf{D}^T\mathbf{p}$ in Eq. (2.22) (the superscript $n+1$ has been dropped to simplify the notation)), where the dependency on $\mathcal{W}$ is included in $\mathbf{D}^T$(i.e., the dependency on the shape function[2]). From this observation, one can show [28] that Eq. (2.28) translates into the discrete form

$$\exists \beta > 0 \, , \, ||\mathbf{D}^T\mathbf{p}|| > \beta ||\mathbf{p}||. \tag{2.29}$$

If this is violated, or in other words if β becomes arbitrarily small as the mesh is refined, it implies that some pressure modes $\mathbf{p}$ may actually induce very limited changes on $||\mathbf{D}^T\mathbf{p}||$, such that large oscillations involving these modes may occur. Particularly, it is known that using equal-order trial functions for the velocity and pressure (first order in the present case) leads to the violation of the LBB condition, and therefore leads to an unstable discretization [27, 29–32]. In this case, methods to stabilize the solution should be applied.

[2] As explained above, the exact solution of the original PDE is obtained if the equations are written in the weak form for all possible values of the functions $\mathbf{w}$ and q. Practically speaking, by choosing the shape functions, particular sets of functions $\mathbf{w}$ and q are chosen.

2.2 The monolithic pressure stabilizing Petrov Galerkin (PSPG)

To address this issue, a Petrov-Galerkin formulation can be used instead of the classical Galerkin formulation. As shown by Hughes et al. [33], even when using equal-order velocity-pressure couples, this formulation does satisfy the LBB condition. The main idea is to use test functions $\overline{\mathbf{w}}$ that are different from the interpolation functions for the derivation of the weak form of the problem. Specifically,

$$\overline{\mathbf{w}} = \mathbf{w} + \frac{\tau^e_{\text{PSPG}}}{\rho}\nabla q\,, \tag{2.30}$$

where τ^e_{PSPG} is a characteristic stabilization time and ∇q is the gradient of the test function used for the continuity equation. It results in the insertion into the continuity equation of a weighted sum of the residuals of the (vectorial) momentum equation, respectively projected onto the node direction in each adjacent elements[3] (in the expression below, the vector ∇q is of course always pointing toward the direction of the concerned node):

$$-\int_\Omega \nabla\cdot\mathbf{u}\,q\,d\Omega + \sum_{e=1}^{N_{el}} \tau^e_{\text{PSPG}} \int_{\Omega_e} \frac{1}{\rho}\nabla q\cdot\left(\rho\frac{D\mathbf{u}}{Dt} + \nabla p - \mu\nabla(\nabla\mathbf{u} + \nabla\mathbf{u}^T) - \rho\mathbf{b}\right) d\Omega_e\,. \tag{2.31}$$

The method is a residual method, as the stabilization term in the continuity equation vanishes at convergence, which leads to a fully consistent scheme. Note that, while the present work is based on the monolithic pressure stabilizing Petrov Galerkin (PSPG) method, other stabilization approaches exist, such as pressure splitting and algebraic splitting schemes [1], finite increment calculus (FIC) [34] and others.

With this new stabilization term the semi-discrete system of equations becomes (see Cerquaglia [1] for detailed derivation and implementation),

$$\mathbf{M}\frac{D\mathbf{u}}{Dt} + \mathbf{K}\mathbf{u} + \mathbf{D}^T\mathbf{p} = \mathbf{f}\,, \tag{2.32}$$

$$\mathbf{C}\frac{D\mathbf{u}}{Dt} - \mathbf{D}\,\mathbf{u} + \mathbf{L}_\tau\mathbf{p} = \mathbf{h}\,, \tag{2.33}$$

where $\mathbf{C}$ is a dynamic stabilization matrix, $\mathbf{L}_\tau$ is a discretized stabilization Laplacian operator, and $\mathbf{h}$ represents the projected body forces. For a backward Euler time integration, the system writes

$$\begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}^{n+1} + \mathbf{K}^{n+1} & \mathbf{D}^{n+1,T} \\ \frac{1}{\Delta t}\mathbf{C}^{n+1} - \mathbf{D}^{n+1} & {\mathbf{L}_\tau}^{n+1} \end{pmatrix} \begin{pmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{pmatrix} = \begin{pmatrix} \frac{1}{\Delta t}\mathbf{M}^n\mathbf{u}^n + \mathbf{f}^{n+1} \\ \frac{1}{\Delta t}\mathbf{C}^n\mathbf{u}^n + \mathbf{h}^{n+1} \end{pmatrix}. \tag{2.34}$$

Unlike the formulation without stabilization, Eq. (2.22), there is no more zero block matrix in the bottom right quadrant, as the matrix ${\mathbf{L}_\tau}^{n+1}$ multiplies the pressure vector. In particular, this leads to a stable time integration while keeping the consistency with respect to the original PDEs.

In this work, this implicit system of equations based on the Backward Euler scheme is used together with a Picard algorithm to iterate over the nonlinearity.[4] This combination leads to a very robust scheme with fast convergence, which motivates its use for the present applications. However, it is not the most efficient approach, and faster algorithms could be used, such as the pressure splitting scheme or the algebraic splitting of the monolithic PSPG equations. Nonetheless, they suffer from other limitations with respect to numerical mass conservation [35], or convergence[5] for instance.

[3] Considering the weak form of the momentum equation, Eq. (2.8), it follows that the additional term should have been added into the momentum equation. However, as it is supposed to be 0 at convergence, it is placed into the continuity equation instead, which has the advantage of filling the zero block matrix in the bottom right quadrant of the original Galerkin formulation of the problem, Eq. (2.22).

[4] Even if the equations are linear in the unknown variables, the node coordinates vary in time as the method is Lagrangian. Because the matrices in Eq. (2.22) depend on the coordinates of the moving nodes at time step n+1, the overall scheme is nonlinear.

[5] This has been observed in practice by the author.

Choice of the characteristic stabilization times τ^e_{PSPG}

The mathematics underlying the choice of the stabilization terms τ^e_{PSPG} is a very complex topic that has received very limited considerations in the literature. Most authors (for instance Tezduyar et al. [36]) simply take similar expressions as for the SUPG (Streamline-Upwind Petrov-Galerkin) stabilization for Eulerian advection-diffusion problems.

The interpretation of these stabilization times for the SUPG scheme, τ_{SUPG}, is somehow simpler. It is for instance well known that for advection problems, central finite discretization with explicit time integration is unstable, and an upwind scheme is often used to get stable time integration schemes. In this context, the standard Galerkin formulation on a uniform mesh leads to a spatially centered discretization, as the shape functions are similar irrespective of the direction. The SUPG stabilization simply consists in decentralizing the shape function in the direction of the advection speed. As the viscosity has a stabilizing effect in advection-diffusion problems, one can define a viscous stabilization time,

$$\tau_\nu = \frac{h^2}{12\nu}\,, \tag{2.35}$$

where ν is the kinematic viscosity of the fluid, and h is a characteristic element length [36]. This stabilization time is low when the viscosity is high, leading to a limited stabilization.

The other kind of stabilization time is the advection time, which depends on the norm of the local velocity field $||\mathbf{u}||$,

$$\tau_u = \frac{h}{2||\mathbf{u}||}\,. \tag{2.36}$$

Both τ_ν and τ_u are usually combined to obtain a smooth transition between the two possible values:

$$\tau^e_{\text{SUPG}} = \left(\tau_\nu^{-1/2} + \tau_a^{-1/2}\right)^{-1/2}\,. \tag{2.37}$$

Based on this, Tezduar et al. [36] have simply considered a similar expression for τ^e_{PSPG},

$$\tau^e_{\text{PSPG}} \simeq \tau^e_{\text{SUPG}}\,, \tag{2.38}$$

where the local velocity $\mathbf{u}$ has been replaced by a global scaling velocity U (based on a characteristic velocity scale of the application), and the element length h has been replaced by another one, h^e, defined as the length of the diameter of the circle which is area-equivalent to the element.

In the present work, the stabilization implemented by Cerquaglia [1] and based on Tezduyar et al. [36] has been used. It depends on a local Reynolds number,

$$Re_U = \frac{U\,h^e}{2\nu}\,, \tag{2.39}$$

to determine whether the problem is locally dominated by convective or diffusive effects. From there, a sharp transition between $\tau^e_{\text{PSPG}} = \tau_u$ and $\tau^e_{\text{PSPG}} = \tau_\nu$ (using a global velocity U and the element size h^e in their respective definitions) is performed for regions where $Re_U \geq 3$ and $0 < Re_U < 3$, respectively. In practice, this stabilization has given good results, but it is not known how close it is from optimality. In particular, the optimal choice of τ^e_{PSPG} is usually not clear and has not been widely studied, which may be partly explained by the fact that other authors prefer to use compatible shape function pairs (shape functions of different orders for $\mathbf{u}$ and p).

Lumped mass matrix and consistent mass matrix

As it is classically done in the finite element method, the integrals involving the shape functions are evaluated using Gauss-Legendre quadrature, which consists in computing the integral by evaluating the value of the argument at specific points, with some predetermined weights associated with them. The advantage is that using n Gauss points, the methods enables to calculate exactly the integral of polynomials of degree up to $2n-1$. Except for the mass matrix, the integrand of all matrices in Eqs. (2.16) to (2.19), are constant or linear, which means that a single Gauss point at the center

of the triangular (tetrahedral in 3D) elements is sufficient to compute these integrals. On the other hand, the correct computation of the consistent mass matrix, because of the bi-linear form inside the integral, requires the use of three (four in three dimensions) Gauss points (two points is not sufficient here because of the symmetry requirement for the position of the Gauss points). Using only one Gauss point leads to an incorrect value of the integral, and results in a mass matrix with all the components being equal to $\frac{\rho \Omega_e}{9}$, Ω_e being the element area. This would lead to larger space discretisation errors. Another possibility, but which unfortunately also leads to larger space discretization errors, is the use of a lumped mass matrix $\mathbf{M}_l$ [37, 38], for which the only non-zero terms are on the diagonal. It is obtained by summing all the rows of the consistent mass matrix $\mathbf{M}_c$[6]. At the element level, one has

$$M^e_{l,3i+I,3j+J} = \delta_{IJ} \sum_{K=1}^{3} M^e_{c,3i+I,3j+K} = \delta_{IJ} \sum_{K=1}^{3} \int_e \rho \, \hat{N}^I \hat{N}^K d\Omega_e \, \delta_{ij} \, . \tag{2.40}$$

By definition, the sum of all shape functions is equal to one,

$$\sum_{K=1}^{3} \hat{N}^K = 1 \, , \tag{2.41}$$

and one finally has

$$M^e_{l,3i+I,3j+J} = \delta_{IJ} \, \rho \int_e \hat{N}^I d\Omega_e \, \delta_{ij} = \rho \, \frac{\Omega_e}{3} \, \delta_{IJ} \delta_{ij} \, . \tag{2.42}$$

The advantage of a lumped mass matrix is that it can be easily inverted when an explicit time integration is used. Moreover, its computation is immediate and does not require any Gauss point. For this last reason, it is the default approach here for all test cases considered, even if an implicit time integration is used. Consequently, the resulting space discretization is at most second order accurate for a uniform mesh and lower for a non-uniform mesh[7] (while it would theoretically be fourth order accurate using a consistent mass matrix on a uniform mesh [37]). This should be kept in mind as meshes in the classical PFEM are distorted at each time step so that a perfectly uniform size cannot be ensured.[8] Moreover, the present work aims to develop a new algorithm precisely for non-uniform meshes.

2.3 The α-shape technique

Because the particles move with the flow, the initial mesh is rapidly distorted. In the PFEM, the mesh used at a previous time step is thus discarded and a new mesh is constructed from the new cloud of particles, possibly at every time step or at least whenever the mesh becomes too distorted. The process is illustrated for the two-dimensional case in Fig. 2.2. Given a cloud of particles in the plane (a), a fast Delaunay triangulation is performed to define the new triangular mesh (b). Nonetheless, the entire convex hull defined by the particles is triangulated. In order to identify the actual boundary of the fluid, some elements must be eliminated (c). In the PFEM, this is traditionally achieved through a geometric criterion based on the α-shape technique.

Originally, the α-shape technique has been developed as a tool to separate subgroups of simplices contained in the convex hull of a point set [25, 39]. It simply consists in only keeping in the triangulation the elements having a circumscribed circle whose radius r_c is smaller than a threshold α. Because otherwise the threshold α would be problem-dependent, it is customary in the PFEM to use the non-dimensional form of the α-shape criterion,

$$\frac{r_c}{h} < \alpha \, , \tag{2.43}$$

[6]The consistent mass matrix is by definition the one obtained with the original finite element formulation, using the appropriate number of Gauss points. However here, the results of the "lumping process" would be the same whether the appropriate number of Gauss points was used or not.

[7]This can be shown for the advection-diffusion problem on an Eulerian mesh.

[8]This is further compounded by the fact that the matrix is built from the positions at time $n+1$, i.e., using a mesh slightly more distorted that the one at the current time step n.

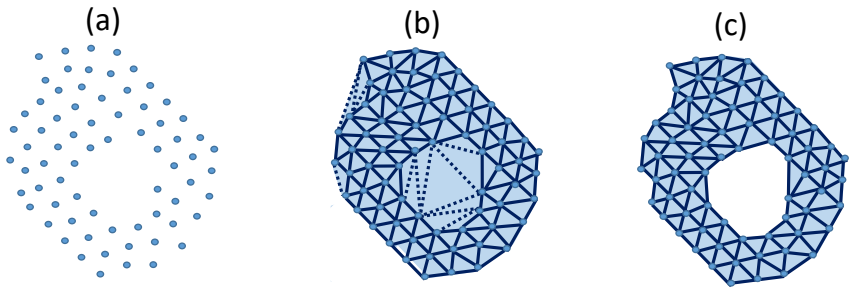


Figure 2.2: Illustration of the meshing procedure and α-shape technique. (a) Cloud of particles obtained from the previous time step, (b) mesh over the convex hull of the particle cloud after Delaunay triangulation, (c) final mesh after elimination of the elements that are too elongated (i.e., with an α-value larger than the threshold). The deleted triangles are considered as empty space.

where h is a characteristic length scale of the mesh. Typically, h is defined as the average length of the smallest element edge over the mesh [1, 40]. Choosing a global value α between 1.2 and 1.5 allows to discard triangles that are too badly shaped or too large.

The drawback of the above method is that it presupposes some uniformity of the mesh size, preventing thus its direct use with mesh adaptation. More specifically, applying the above α-shape technique to a non-uniform mesh would eliminate all triangles that are too large even if they are equilateral and part of the fluid domain, as illustrated in Fig. 2.3. Increasing the value of h would remedy it, but then smaller badly-shaped elements that should be eliminated would be incorrectly kept.

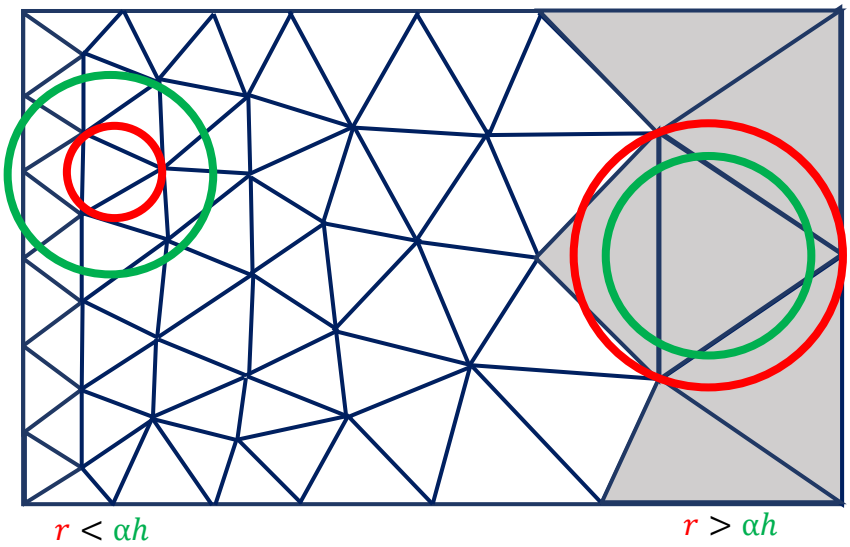


Figure 2.3: Non-uniform mesh with a size progression from left to right. Applying the criterion of Eq. (2.43) using a constant value of h and α would wrongfully eliminate all triangles that are "large" (the grey triangles in this figure) despite their regular shape.

To circumvent this issue, one could replace the global length scale h by a local value h_e, e.g., the shortest length of the corresponding triangle. Although this approach would eliminate badly-shaped triangles, some larger elements would be wrongly kept, as illustrated in Fig. 2.4. In this case, the large triangle inside the cylinder is not discarded and the algorithm fails to identify the cylinder immersed in the fluid as a solid body.

Nonetheless, such an approach could be considered if the local value h_e is understood as an average over some local region rather than directly related to the corresponding element, and if only smooth spatial variations of h_e are permitted. A related method is proposed in the present work, as described in Section 3.2.1.

Another option is to rely on a generalization of the α-shape technique, the so-called weighted α-shape

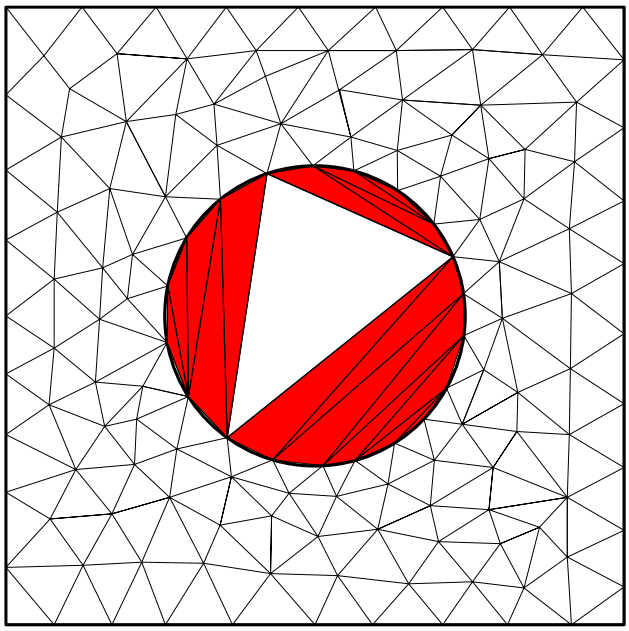


Figure 2.4: Mesh obtained after Delaunay triangulation for a cylindrical solid body immersed in a fluid. The use of a local scaling length h_e instead of a global measure h of the mesh size in the α-shape technique would correctly eliminate the badly-shaped elements (in red) but fail to discard the large regular triangle inside the solid body.

method [25, 41, 42], in conjunction with weighted Delaunay triangulation (also referred to as regular triangulation) [43]. In this case, large triangles or small badly-shaped triangles can be simultaneously discriminated by introducing user-defined weights at each node. Note that it can be shown that the aforementioned approach based on a local length scale h_e is equivalent to the weighted α-shape method in the limit of a uniform mesh (i.e., with a roughly a constant spacing between the nodes and a unique weight for all of them). The two approaches are therefore similar for smoothly varying meshes. Nevertheless, the weighted α-shape method is still a geometric criterion and special treatment is required in the vicinity of the free surface to minimize the mass creation or destruction that is inherent to the PFEM. The weighted α-shape method is introduced in details in Section 3.1.2.

2.4 The α-shape technique and mass conservation

For incompressible flows, the total mass is directly proportional to the volume of the fluid. In other words, mass conservation implies volume (surface area in 2D) conservation. Variations of total volume, ΔV, can thus be used to quantify the error in mass conservation. In the PFEM, one can identify two different sources for variations of in total volume. The first contribution, ΔV_{num}, is common to other numerical methods for free-surface flows and is associated with the time integration and the corresponding displacement of boundary nodes. It can be affected by various aspects of the iterative solvers, such as the stabilization techniques or the time integration scheme [44]. It is shown in Section 3.3 how, for an ideal space discretization, a backward Euler time integration can affect the mass conservation.

The second contribution to volume variations, ΔV_{rem}, is related to the addition and deletion of mesh elements when applying the α-shape technique during the boundary detection process. It is usually the dominating contribution in the PFEM. The mesh adaptation algorithm developed in this work aims to reduce this second kind of mass conservation error.

As an example, volume creation/destruction due to remeshing can take place when two free surfaces, or two parts of the same free surface, approach each other, as illustrated by the falling droplet merging with the bulk of the fluid in Fig. 2.5. When the distance between the two boundaries becomes of the order of the mesh size, the elements between them are not discarded by the boundary identification algorithm, thus creating mass. On the other hand, mass can be destroyed along a free surface that is stretched, as also illustrated in Fig. 2.5. In this case, the stretching pulls boundary nodes apart leading

to obtuse triangles that are eliminated by the α-shape criterion. Similar errors in mass conservation can also be observed around the contact point of a free surface and a solid wall (see Fig. 2.6), where mass is created at the front of the moving fluid wetting the surface and destroyed at its back.

Two approaches can be considered to limit mass creation/destruction inherent to the PFEM. On the one hand, one can locally refine the mesh in the critical regions of the free surface, which provides another motivation for mesh adaptation. On the other hand, a different treatment can be applied to the nodes on the free surface during the boundary recognition step. Nevertheless, this requires some kind of tracking of the free surface, e.g., by tagging the nodes belonging to it. Both approaches are considered in the present work, as explained in Chapter 3.

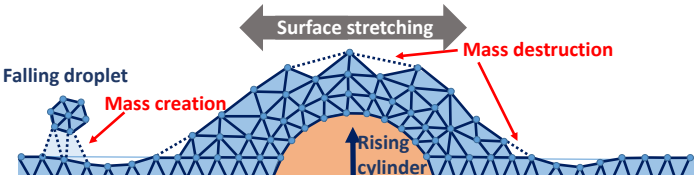


Figure 2.5: Typical examples of error in mass conservation at a free surface: mass creation due to two free surfaces approaching each other (falling droplet on the left) and mass destruction due to the stretching of the free surface (above the rising cylinder in the middle).

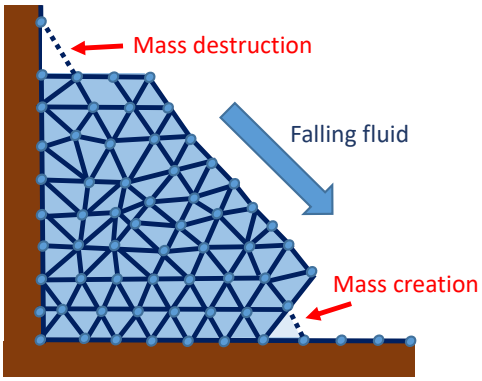


Figure 2.6: Typical examples of error in mass conservation at a solid wall: mass creation at the fluid front along the horizontal surface and mass destruction behind the falling fluid on the vertical wall.

2.5 Adding and removing nodes in uniform meshes

In the PFEM, the displacement of nodes can be such that some nodes become progressively very close to each other and others quite far apart. In these cases, and despite a new triangulation, the mesh quality can significantly deteriorate, with some large elements adjacent to much smaller ones. This can become problematic as the α-shape algorithm might then incorrectly eliminate elements that should be kept or conversely. To remedy this issue, the classical PFEM typically relies on the addition of new nodes or the deletion of old ones to ensure a more or less uniform node density.

Concretely, a node is simply added in the middle of an element of area A when

$$A > \frac{1}{2} k_1 \, h^2 \,, \tag{2.44}$$

where h is the average size of the minimal edge over all elements, and k_1 is a user-defined parameter. The new node is then assigned the averaged value (velocity, pressure, etc.) of the three nodes defining the original element.

When two nodes become too close to each other, one of them is simply eliminated. More precisely, edges whose length L is smaller than a threshold value, $L < k_2 h$, where k_2 is another user-defined value, are identified and one of their two nodes is then discarded. In the following, all simulations based on the classical PFEM with a uniform mesh rely on the parameter values $k_1 = 0.5$ and $k_2 = 0.5$.

It should be emphasized that, in practice, the addition / removal of nodes in the classical PFEM remains usually very limited because of the incompressibility condition. The use of an optimal criterion and a robust algorithm is thus not critical. This is however not the case for mesh adaptation, and more advanced approaches for adding and removing nodes are proposed in the next chapter.

Chapter 3

Mesh adaptation algorithm for the Particle Finite Element Method

As explained in the previous chapter, the classical PFEM, and in particular the α-shape technique, assume a more or less uniform mesh. This precludes local mesh refinement and limits the computational efficiency of the method. One of the main objectives of this work is thus to develop and implement a mesh adaptation algorithm for the PFEM to address this shortcoming and to possibly also improve the mass conservation property of the method. This chapter first presents a literature review on mesh adaptation in general and on specific aspects in the context of the PFEM. Based on this, the choice of the novel mesh adaptation strategy is rationalized and a detailed description of the different steps of the algorithm is provided in Section 3.2. This is followed by practical guidelines on how to use the method, including the choice of the space and time discretizations. Finally, the theoretical extension of the algorithm to three-dimensional tetrahedral meshes is discussed in the last section of the chapter.

3.1 Literature review on mesh adaptation and boundary recognition algorithms for non-uniform meshes

To choose an appropriate strategy for mesh adaptation and boundary recognition in the context of the PFEM, an overview of the existing methods in a more general context is first given. Afterwards, an overview of the two possible paths to extend the α-shape technique to non-uniform meshes is given.

3.1.1 Mesh adaptation

Mesh adaptivity essentially refers to the use of a discretization based on mesh elements of different sizes, such that high resolution is used in regions where the solution features require it, while coarser elements are used where the solution is smooth. This strategy has been successfully used for decades in Eulerian mesh-based methods for either fixed meshes or meshes that evolve during the simulation. At the most basic level, adaptivity can simply be achieved by defining a priori a fixed but non-uniform mesh. In this case, the characteristics of the mesh are usually based on user expertise and/or some mesh convergence study. At a more advanced level, iterative or time dependent solution-based mesh adaptation relies on the local solution to define the local mesh properties. This is often referred to as Adaptive Mesh Refinement (AMR) and is typically used with unstructured meshes or octree (quadtree in 2D) grids [45–48].

The first step of the AMR is the identification of the mesh elements that need to be adapted. In that regard, Bansch et al. [49] suggest that such mesh adaptation techniques can be divided into two categories. The error-based methods use an estimate of the interpolation error (see [49–51] for some examples). The other category encompasses the heuristic methods in which the elements are refined following physical criteria [52]. These methods are described in Sections 3.1.1.1 and 3.1.1.2, respectively. Beyond this classification, one can also highlight the metric-based mesh adaptation. Roughly, the latter approach defines the local mesh size (possibly anisotropically) based on solution-based quantities, themselves being defined from error estimates or not (heuristic). A more complete description of this

type of methods is given in Section 3.1.1.3.

Irrespective of the method used to identify the mesh elements to be adapted, the second step of AMR consists in adapting the mesh, typically through splitting or grouping of mesh cells/nodes. This is generally done iteratively until the error reaches a given threshold (error-based), or until the local mesh size reaches the prescribed one (heuristic/metric-based). While splitting or grouping cells is relatively straightforward in the context of octree grids, refining and coarsening an unstructured mesh without hanging nodes often requires to modify not only the targeted element itself but also its neighbors.

At last, beyond the geometric aspects of defining new, coarser or finer, elements, the main challenge is to assign them a solution value. In particular, interpolation of new values should ensure that conservation principles are satisfied and that no large error is introduced.

3.1.1.1 Error-based mesh adaptation

A classical way to do mesh adaptation is to use error estimates (see [49–51] for some examples). To illustrate the approach, one can consider the incompressible Navier-Stokes equations in the following compact form:

$$\boldsymbol{\mathcal{C}}(\mathbf{u}, p) = \mathbf{0}\,. \tag{3.1}$$

In order to perform numerical simulations, this set of continuous equations must be transformed into a set of discrete equations. More precisely, the objective is to find the discrete solution $(\mathbf{u}^*, \mathbf{p}^*)$ of a discretized version of $\boldsymbol{\mathcal{C}}(\mathbf{u}, \mathbf{p})$:

$$\boldsymbol{\mathcal{D}}(\mathbf{u}^*, \mathbf{p}^*) = \mathbf{0}\,. \tag{3.2}$$

In other words, the discrete system of equations $\boldsymbol{\mathcal{D}}(\mathbf{u}^*, \mathbf{p}^*) = \mathbf{0}$ tries to approximate the system of differential (i.e., continuous) equations $\boldsymbol{\mathcal{C}}(\mathbf{u}, \mathbf{p}) = \mathbf{0}$. With a finite mesh size, the exact solution of the discrete equations does not satisfy exactly its continuous form and one has

$$\boldsymbol{\mathcal{C}}(\mathbf{u}^*, \mathbf{p}^*) = \boldsymbol{\mathcal{D}}(\mathbf{u}^*, \mathbf{p}^*) + \boldsymbol{\epsilon}_T \Longleftrightarrow \boldsymbol{\mathcal{C}}(\mathbf{u}^*, \mathbf{p}^*) = \boldsymbol{\epsilon}_T\,, \tag{3.3}$$

where $\boldsymbol{\epsilon}_T$ is called the truncation error and depends on higher-order derivatives of the variables than those present in $\boldsymbol{\mathcal{C}}(\mathbf{u}, \mathbf{p})$, and on the vector of local mesh size L (the side of the squares in a Cartesian mesh for instance).

For a consistent and stable system of finite difference equations, $\boldsymbol{\mathcal{D}}(\mathbf{u}^*, \mathbf{p}^*) = \mathbf{0}$, i.e., a discretization whose solution converges towards the continuous solution as the mesh is refined, the truncation error can be expressed as a power series of the mesh size L with coefficients that depend on the derivatives of the variables. The leading order term of this series (i.e., the term with the lowest power of L) mainly determines the convergence of the approximate solution to the exact solution when the mesh is refined. The calculation (or estimation) of this leading-order term can thus be used to guide the mesh adaptation. In error-based mesh techniques, the mesh is locally and iteratively adapted to ensure that the leading-order term of the truncation error remains within a given threshold range. This has the advantage to provide, for a given number of nodes, an optimal distribution of the space discretization errors, provided that sufficiently good error estimates are used, which is unfortunately not always possible. In particular, the estimation of the truncation error can be difficult and/or expensive to compute, especially if the problem is highly nonlinear and/or if complex discretization schemes are used, such as in CFD and FSI problems [53]. Additionally, it usually requires the evaluation of higher-order derivatives. If linear finite elements are used, these derivatives cannot be evaluated at the element level and adjacent cells must be considered, which considerably increases the complexity of the approach with unstructured meshes (as shown in Appendix B, estimating higher order derivatives is possible but rather inaccurate and expensive when using linear finite elements on an unstructured mesh).

3.1.1.2 Heuristic mesh adaptation

When good error estimates cannot be computed in a cost-efficient way, a more attractive approach may be to identify flow features qualitatively related to the space discretization errors, but without knowing the exact quantitative link between them. Such flow features can be identified through quantities derived from the flow variables. As the truncation error of the discretized scheme is large when the

derivatives of the variables are large, it is therefore relevant to use quantities that are related to the variable derivatives of different orders, but avoiding to compute those. In this regard, it is advantageous to use reference quantities that can be computed at the element level. This is for instance the case for the first order derivatives, or gradients ([54, 55]), either on a regular or on a non-regular mesh, as first-order derivatives are generally large when the higher-order derivatives are themselves large.

Such reference quantity ϕ can for instance be the norm of the velocity gradient or the Hessian matrix[1] (i.e., the matrix of second-order derivatives), or the norms of their determinants. In any case, once ϕ has been chosen, one can define a pseudo-error $\tilde{E}(\phi, L)$ related to it. For instance,

$$\tilde{E} = \mathcal{F}(\phi) L^p \,, \tag{3.4}$$

where $\mathcal{F}(\phi)$ is some function of ϕ and $p \geq 1$. The mesh is locally refined when this pseudo-error is too large and locally coarsened when it is too low[2], which implies that it should remain in a given range:

$$\tilde{E}_1 < \tilde{E} < \tilde{E}_2 \,. \tag{3.5}$$

In Eq. (3.4), the expression is invertible and one can obtain the mesh size as a function of the pseudo-error and ϕ, which gives the following inequality,

$$\left(\frac{\tilde{E}_1}{\mathcal{F}(\phi)} \right)^{\frac{1}{p}} < L < \left(\frac{\tilde{E}_2}{\mathcal{F}(\phi)} \right)^{\frac{1}{p}} \,. \tag{3.6}$$

In other words, the mesh size is enforced to remain in some user-defined range. Furthermore, Eq. (3.6) can be rewritten as

$$k_1 \, L^*(\phi) < L < k_2 \, L^*(\phi) \,, \tag{3.7}$$

where k_1 and k_2 are user-defined thresholds and $L^*(\phi)$ is a local target mesh size that depends on ϕ. The heuristic approach is therefore equivalent (provided that the pseudo-error $\tilde{E}(\phi, L)$ can be inverted) to imposing a local target mesh size L^* based on the reference quantity ϕ, and by enforcing the local mesh size L to remain within a given range around L^*.

Note that the "heuristic" nature of the method lies in the arbitrariness of the definition of $\tilde{E}$. Indeed, if $\tilde{E}$ was an accurate representation of the error, the method would therefore be considered as an "error-based" method. However, the errors often consist of more complex expressions that are difficult to invert, such that within error-based methods, the mesh is rather iteratively refined to control the errors, as explained above.

3.1.1.3 Metric-based mesh adaptation

A generalization of the idea of imposing a local target mesh size consists in the metric-based mesh adaptation [56]. The formalism of the method is based on a one-to-one mapping between the real physical space $\mathcal{X}$ and another abstract space $\tilde{\mathcal{X}}$. The length scales in one space are connected to those in the other one by the metric defined at each point:

$$\boldsymbol{\mu} \triangleq \frac{d\tilde{\mathbf{x}}}{d\mathbf{x}}. \tag{3.8}$$

The method is such that the mesh elements expressed in the abstract space $\tilde{\mathcal{X}}$ should always remain uniform, with non-dimensional element sizes of the order of unity. Note that the dimensionless nature of the space $\tilde{\mathcal{X}}$ immediately imposes that the dimension of the metric $\boldsymbol{\mu}$ is the inverse of a length scale[3]. As soon as the mesh in the abstract space $\tilde{\mathcal{X}}$ deviates from a mesh of approximate uniform size, the problematic elements are adapted by adding or removing nodes in the space $\tilde{\mathcal{X}}$. The resulting mesh in the real space $\mathcal{X}$ is then evidently non-uniform, exhibiting large elements where $||\boldsymbol{\mu}||$ is small and

[1] Note that there is one Hessian matrix associated with each variable, such that this choice requires to build some combination of them.

[2] Some caution should be exercised to ensure that the local mesh size remains inside a given threshold, in case the errors cannot be decreased down/increased up to the user-defined upper and lower thresholds.

[3] This implies that, irrespective of the quantities involved to build the metric, they should be normalized accordingly.

conversely. Within this framework, the mesh refinement can also be anisotropic (i.e., enabling the use of different cell dimensions depending on the direction), which is the case as soon as the metric deviates from a multiple of the identity matrix.

The key question is therefore to know how to build the metric. Here again, the building procedure can either be error-based or heuristic. A particular case of metric-based mesh adaptation based on error estimators relies on a combination of the Hessian matrices associated with the different flow variables [56–58]. Without going into the mathematical details, this approach ensures that the interpolation error remains under the same bound everywhere in the mesh [56].

For heuristic approaches, the arbitrariness of the choice of the metric makes it more suited for isotropic refinement, as only one scalar is required to define the isotropic metric. In this case, the metric is proportional to the identity matrix $\mathbf{I}$, i.e.,

$$\boldsymbol{\mu} = \frac{1}{L^*(\phi)}\mathbf{I}\,, \tag{3.9}$$

where the coefficient of proportionality is the inverse of the target mesh size L^* already introduced in Section 3.1.1.2. Using Eq. (3.8), it follows that the element sizes dx in the real space scale as L^* while those in the abstract space, $d\tilde{x}$, scale as unity.

The advantage of such a definition is that it introduces more flexibility in the way the mesh refinement is performed. The target mesh size L^* can be computed based on multiple criteria, either solution-based or geometry-based, depending on the application. This feature, as shown in the next section, makes the use of an isotropic metric a method of choice for the present application. In particular, it is preferred over the choice of an anisotropic metric, because this last choice would lead to several difficulties. In the specific context of triangular (tetrahedral in three-dimensions) unstructured meshes, the main difficulties come from the fact that the metric should be integrated at all the steps involving the mesh, from the remeshing procedure (Delaunay or regular triangulation), to the boundary recognition algorithm (α-shape technique) and the mesh adaptation. All these steps would have to be performed in the abstract space $\tilde{\mathcal{X}}$ while solving the discrete system of equations would be done in the real physical space $\mathcal{X}$. Since these steps are usually performed using existing libraries, these libraries would have to allow the incorporation of customized metrics, which is not always the case.

After the metric, either isotropic or anisotropic, is defined, the second step of the metric-based mesh refinement consists in adding or removing nodes so that the actual element size (in the physical space $\mathcal{X}$) actually corresponds to the imposed metric. The main challenge is to make sure that conservation principles are satisfied and that no large error is introduced when adding/deleting nodes. It is even more challenging when the nodes are subjected to displacements (within a Lagrangian formalism as in the present case), because the mesh should be permanently adapted, even for a case that would be steady in the Eulerian frame of reference. The advected elements may be permanently coarsened or refined depending on, among other things, where they are going. In this context, anisotropic mesh adaptation may be an issue, such that isotropic mesh adaptation is preferred here.

At last, it should be emphasized that there are several ways to add and remove nodes depending on the type of mesh used. Introducing them would bring us too far for this short introduction about mesh refinement methods.

3.1.2 Boundary recognition for non-uniform meshes in the PFEM

In the context of the PFEM, as a new mesh is frequently rebuilt, mesh adaptivity also includes the way the new elements are created or removed from the new triangulation, i.e., how the domain boundary is determined. The main challenge here consists in devising a boundary recognition method adapted to non-uniform meshes. The two main methods that are presented below differ by the kind of triangulation used. To understand both of them, the Delaunay [59–61] and the regular [43] triangulations are thus first briefly exposed.

3.1.2.1 Delaunay and regular triangulations

The best approach to explain the difference between both triangulations consists in first introducing the concept of Voronoï diagram and power cells. For a set of points $\mathbf{x}^I$ in the plane (space in 3D)), the

Voronoï diagram is the partition of the entire plane (space) into small regions around each point $\mathbf{x}^I$, called Voronoï cells and defined as [62]

$$\mathcal{V}^I \triangleq \{\mathbf{x} \in \mathbb{R}^2 | d(\mathbf{x}, \mathbf{x}^I) \leq d(\mathbf{x}, \mathbf{x}^J) \, \forall J \neq I\} \,, \tag{3.10}$$

where $d(\mathbf{x}^A, \mathbf{x}^B)$ is the Euclidean distance between points $\mathbf{x}^A$ and $\mathbf{x}^B$. Alternatively, the Voronoï cell associated with point $\mathbf{x}^I$ can be defined as the intersection of all half planes $\mathcal{H}^{IJ}$ containing the point $\mathbf{x}^I$, where

$$\mathcal{H}^{IJ} \triangleq \{\mathbf{x} \in \mathbb{R}^2 | d(\mathbf{x}, \mathbf{x}^I) \leq d(\mathbf{x}, \mathbf{x}^J)\} \,. \tag{3.11}$$

The dual of the Voronoï diagram is called the Delaunay triangulation and is obtained by connecting the points $\mathbf{x}^I$ and $\mathbf{x}^J$ of adjacent Voronoï cells $\mathcal{V}^I$ and $\mathcal{V}^J$, as illustrated in Fig. 3.1. It has been shown that among all possible two-dimensional triangulations, the Delaunay is optimal regarding several aspects [61]:

- it maximizes the minimum angle;
- it minimizes the maximum circumradius;
- any triangulation exempt of obtuse triangle is a Delaunay triangulation;
- several algorithms exist to transform any triangulation into a Delaunay triangulation, with a complexity down to $\mathcal{O}(N \log N)$, where N is the number of points.

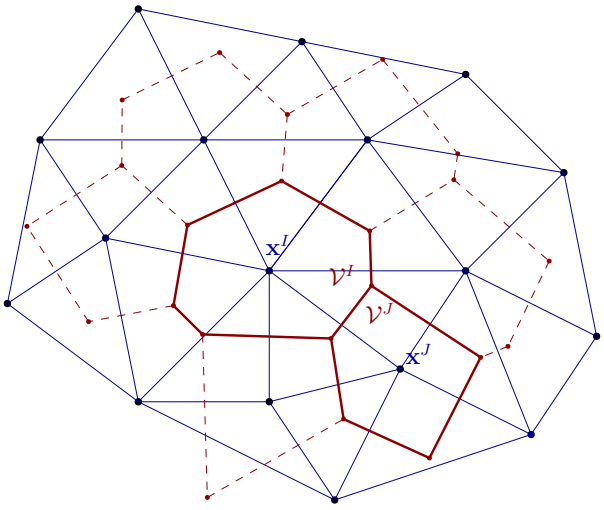


Figure 3.1: Schematic illustration of the Voronoï diagram (red lines) of a 2D cloud of points (blue dots), together with its dual set corresponding to the Delaunay triangulation (blue lines).

The regular triangulation is defined in an analogous manner to the Delaunay triangulation. The difference comes from the fact that the Euclidean distance d between points $\mathbf{x}^I$ is replaced by another distance d_p between weighted points $(\mathbf{x}^I, w_I)$, where w_I is the weight of point $\mathbf{x}^I$. Specifically, the new distance between an arbitrary point $\mathbf{x}$ of the plane (space) and the weighted point $(\mathbf{x}^I, w_I)$ is called the power distance and is defined as

$$d_p\left(\mathbf{x}, (\mathbf{x}^I, w_I)\right) \triangleq d(\mathbf{x}, \mathbf{x}^I)^2 - w_I \,. \tag{3.12}$$

The power cells $\mathcal{P}^I$ are then obtained similarly to the Voronoï cells $\mathcal{V}^I$ by replacing the Euclidean distance d by the new power distance d_p in its definition (3.10), i.e., they are also the intersections of half-planes given by Eq. (3.11) (with d replaced by d_p). A noticeable difference is that, with this new definition, the half planes might not be located between the two corresponding points. As a consequence, their common intersections may be empty. The ensemble of all power cells is called the power diagram, and its dual set is the regular triangulation and is obtained by connecting the points $\mathbf{x}^I$ and $\mathbf{x}^J$ of adjacent power cells $\mathcal{P}_i$ and $\mathcal{P}_j$. As some power cells are empty, their associated nodes do not belong to

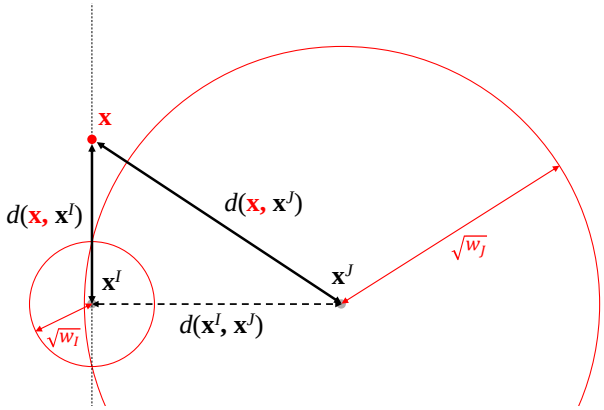


Figure 3.2: Limiting case when the squared Euclidian distance between two weighted points $(\mathbf{x}^I, w_I)$ and $(\mathbf{x}^J, w_J)$ is equal to the difference of their weight. In this case, the line delimiting the half plane $\mathcal{H}^{IJ}$ (vertical dashed line), i.e., the locus of equal power distance between $(\mathbf{x}^I, w_I)$ and $(\mathbf{x}^J, w_J)$, contains the point $\mathbf{x}^I$. Note that the nodal weights corresponds to the squared radius of the circles (in red) centered on the mesh nodes.

the regular triangulation and are thus ignored. This event can happen if there is a too large mismatch between the weights and the real distance between nodes. For instance, the half planes equidistant (using the distance d_p) to two weighted nodes $(\mathbf{x}^I, w_I)$ and $(\mathbf{x}^J, w_J)$ can be outside the segment joining these nodes if the absolute value of the difference of their weights is greater than the squared Euclidian distance between them:

$$|w_I - w_J| > d(\mathbf{x}^I, \mathbf{x}^J)^2 \,. \tag{3.13}$$

This can be checked by visualizing the limiting case in which the half plane boundary contains the node with the smallest weight, as illustrated in Fig. 3.2. In this case, by equalizing the power distances $d_p(\mathbf{x}, \mathbf{x}^I)$ and $d_p(\mathbf{x}, \mathbf{x}^J)$, one has

$$d_p(\mathbf{x}, (\mathbf{x}^I, w_I)) = d_p(\mathbf{x}, (\mathbf{x}^J, w_J)) \tag{3.14}$$

$$\Longleftrightarrow$$

$$d(\mathbf{x}, \mathbf{x}^I)^2 - w_I = d(\mathbf{x}, \mathbf{x}^J)^2 - w_J \tag{3.15}$$

$$\Longleftrightarrow$$

$$w_J - w_I = d(\mathbf{x}^I, \mathbf{x}^J)^2 \,, \tag{3.16}$$

which corresponds to the bound of Eq. (3.13) for the case $w_J > w_I$, and where the last equivalence is simply obtained by using the relation of Pythagoras:

$$d(\mathbf{x}, \mathbf{x}^J)^2 = d(\mathbf{x}, \mathbf{x}^I)^2 + d(\mathbf{x}^I, \mathbf{x}^J)^2 \,. \tag{3.17}$$

At last, it should be emphasized that a weighted node $(\mathbf{x}^I, w_I)$ satisfying Eq. (3.13) with all its neighbours $(\mathbf{x}^J, w_J)$ does not belong to the regular triangulation, as no power cell is associated with it.

3.1.2.2 The non-uniform and the weighted α-shape techniques

In its simplest form, the α-shape technique consists in eliminating from the triangulation all the triangles (tetrahedra in three dimensions) whose circumscribed circle (sphere) violates the α-shape criterion,

$$r_c < \alpha \,, \tag{3.18}$$

which is similar to Eq. (2.43) without the global variable h used for non-dimensionalization. As aforementioned, this is a non-local criterion disabling the use of a non-uniform mesh. However, the criterion can become local if the global length scale h is replaced by the local target mesh size L^*_{elem} of the element (this target mesh size is rigorously defined in Section 3.2.1):

$$r_c < \alpha L^*_{\text{elem}} \,. \tag{3.19}$$

The other approach relies on the regular triangulation and is called the weighted α-shape technique [25, 41, 42]. The weighted points $(\mathbf{x}^I, w_I)$ of the triangulation can be seen as small circles (spheres in 3D) centered on $\mathbf{x}^I$ with a radius $\sqrt{w_I}$, as illustrated in Fig. 3.3. The method first consists in finding the circle (sphere) simultaneously orthogonal to the three circles (four spheres) associated to the three (four) weighted points of the triangular (tetrahedral) elements of the regular triangulation. In other words, the center $\mathbf{x}^C$ and radius r of this circle (sphere) are sought. They are obtained by solving the system of three (four) equations associated with the weighted nodes $(\mathbf{x}^I, w_I)$ of the triangle (tetrahedron):

$$d(x_i^I, x_i^C)^2 = r^2 + w_I \,. \tag{3.20}$$

In a second step, the radius of the circumscribed radius r_c is replaced by the new radius[4] r in Eq. (3.18), keeping a global value for α. With this new definition, this technique enables to deal with non-uniform meshes by adapting the nodal weights to some local characteristic element size L:

$$w \approx L^2 \,. \tag{3.21}$$

The method is elegant but introduces new challenges. The first difficulty lies in defining a convenient relation between node weights and the (target) element sizes. But the major challenge stems from the regular triangulation underlying the weighted alpha-shape technique. First, the regular triangulation must be implemented in the solver. Then, and more importantly, nodes could be automatically eliminated if the condition in Eq. (3.13) were satisfied. The information associated with these nodes would then be discarded in an uncontrolled manner.

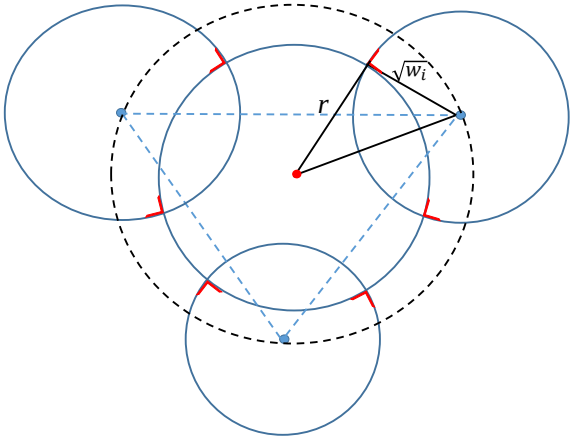


Figure 3.3: Illustration of the reference radius used for the weighted α-shape technique (in two dimensions). Unlike the classical α-shape technique, the radius is not that of the curcumscribed circle, but that of the circle simultaneously orthogonal to the three weighted points of weights w_i.

3.2 Choice and description of the novel method for two-dimensional triangular meshes

Following the discussion of the different possible approaches, the choices for the novel mesh adaptation are now rationalized, taking into account the specific aspects of the existing PFEM solver. In particular, following points should be considered:

- **The use of linear shape functions**, together with non-uniform meshes, limits the order of the derivatives that can be accurately computed at the element level. For instance, the first order derivative is trivially obtained using the slopes of the shape functions, and the second order derivatives can also be computed considering multiple finite elements, as shown in Appendix B. The estimation of higher order derivatives would unavoidably involve adjacent elements and lead to inaccuracies. A simple and fast solution-based mesh adaptation should therefore be based on first-order, and potentially second-order derivatives.

[4]Note that if all the weights are set to zero, the circumscribed circle is recovered.

- **The simulated flows are always unsteady** because of the Lagrangian formulation. Even for a flow that would be at steady state in the Eulerian frame of reference, the particles, elements and nodal variables constantly change. Therefore, it is not suited to use an optimal (in terms of accuracy) and time consuming mesh-refinement strategy such as the error-based mesh refinement, because the refinement criteria is evaluated at almost each time step. The use of a simpler and faster criterion seems more adequate, even if it is not optimal regarding the error distribution.

- **The boundary recognition algorithm** leads to volume variations ΔV_{rem} that are not related to the truncation error coming from the spatial discretization. This feature requires alternative criteria in order to decrease this other source of errors. Therefore, as multiple criteria should be used in the present case, a combination of them through an heuristic and isotropic metric-based formalism seems to be the most appropriate path.

- **The presence of (moving) boundaries** calls for the use of dynamics geometry-based mesh refinement near the no-slip boundaries (moving or not), which further motivates the use of a metric-based formalism.

Regarding the fundamentally unsteady nature of the flows within a Lagrangian formalism, combined with the need for different criteria that encompass the different sources of errors, an heuristic and isotropic metric-based formalism has been chosen. It has the advantage of being flexible regarding the different types of errors to mitigate, while being fast to compute at the element level (as it will be shown, at most first-order derivatives are used). As the chosen metric is isotropic, it is characterized by a local target mesh size L^* (see Eq. (3.9)). The method thus consists first in prescribing a target mesh size L^* based on several heuristic criteria, and then approximately enforcing it by adding or removing nodes in elements whose size differs too much from their target value.

The addition and removal of nodes should remain under the control of the mesh refinement algorithm rather than that of the boundary recognition algorithm. As discussed in Section 3.1.2, the use of the weighted α-shape technique together with regular triangulation may remove nodes from the triangulation in an uncontrolled manner. Moreover, enabling the regular triangulation in the existing PFEM solver would require a major implementation effort. For these reasons, a boundary detection algorithm based on a local α-shape technique is here favored.

A last point has to be mentioned about the boundary recognition algorithm. As previously illustrated, it leads to mass conservation errors in the vicinity of free surfaces, as triangles are created or removed. This effect is mitigated by increasing locally the spatial resolution where the surface gets more deformed. However, this is not sufficient as it does not fundamentally change how the boundary recognition works, which is where the problem of mass conservation originates. The fundamental problem is that each time the mesh is rebuilt, the boundary recognition proceeds on the cloud of points without any additional information, and the boundary known at a previous time step is not leveraged to identify the boundary at the next time step. Quite logically, reusing the information of the boundary location in an appropriate way should enable to decrease mass conservation errors. Therefore, the tracking of boundary nodes constitutes, with the non-uniform α-shape technique, the last key ingredient of the method.

To summarize, the method mainly consists of the following four aspects that are described in detail in Falla et. al [63] and in the next sections:

- **The definition of a target mesh size L^* attached to the nodes** that prescribes how the mesh resolution should be adapted by the addition and removal of nodes. This target value is calculated based on different heuristic criteria.

- **The addition and removal of nodes** to approximately enforce the target mesh size.

- **The use of a non-uniform α-shape criterion** where the parameter h used to obtain the non-dimensional Eq. (2.43) is replaced by L^* to get Eq. (3.19).

- **The tagging of boundary nodes** for locally customizing the boundary detection and the mesh adaptation algorithms, so as to limit the mass conservation errors.

3.2.1 Definition of the target mesh size

The local target mesh size $L^* = \sqrt{A^*}$ is a measure of the desired surface area A^* of the mesh elements. Two approaches are combined to define it. First, geometric mesh refinement is based on the absolute position of the particles in the computational domain and/or on their relative position with respect to solid surfaces. While this type of criterion is adequate in many cases, the actual location of viscous layers is not always known a priori. For instance, the trajectory of a buoyant object is a result of the simulation, so that the position of its wake cannot be easily defined purely geometrically. Therefore, a solution-based mesh refinement is also considered, which relies on a measure of the local velocity gradients. In practice, different criteria (geometric or solution-based) can be considered simultaneously. The actual target mesh size L^* is thus obtained by combining the target mesh sizes $L^*_{(k)}$ from each individual criterion k. Unless specified otherwise, a global minimum is used in this work:

$$L^* = \min_k L^*_{(k)} \,. \tag{3.22}$$

Note that, because the information about elements is lost at each time step when the mesh is discarded, the local target mesh size L^* is calculated and stored at each node n. It is also recomputed at each time step to accommodate for the displacement of the nodes.

3.2.1.1 Geometry-based target mesh size

Viscous layers adjacent to walls are characterized by large velocity gradients. A finer mesh is thus required close to solid boundaries to accurately capture these gradients. The target mesh size at a given location can thus be defined as a continuous function $L^*(d)$ of the shortest distance d to solid walls, whether fixed boundaries or moving bodies. Different analytical functions can be considered. For instance, $L^*(d)$ could correspond to the smallest mesh size L^*_{min} in some region adjacent to a wall, then increase linearly with d to the maximum mesh size L^*_{max} over some prescribed distance and finally remain constant at L^*_{max} further away. The wake region behind a body also typically requires a finer mesh. If the location of the wake is known beforehand, the target mesh size can be prescribed through pseudo geometrical entities defined in either an absolute frame of reference or relative to the body, for instance. Such pseudo-entities could be a rectangular zone or a center line behind the body. Another option could be to define the target mesh size as an analytical function of both the distance and angle with respect to the body.

For complex geometries the calculation of the exact minimal distance to a wall is not trivial and potentially expensive from a computational point of view. In practice it is therefore more convenient to consider each (pseudo-) geometric entity k separately. This leads to several criteria $L^*_{(k)}$. Note that each criterion can use different values for L^*_{min} and L^*_{max}. The actual target size can then be defined as the minimum over the different criteria, according to Eq. (3.22). An example is shown in the top image of Fig. 3.4 for the simulation of the flow around a fixed cylinder. In this case, the target mesh size is set based on the distance to the cylinder and a rectangular zone downstream of it.

Additionally, complex boundaries or bodies can be approximated through simpler geometric shapes (e.g., line, circle, rectangle) to further simplify the definition of the target mesh size. This also provides the opportunity to implement generic criteria that can be used for different configurations. Nevertheless, an adaptation for each specific case is usually still required. The complexity also increases with the number of geometrical elements considered. An additional drawback is that regions of large gradients sometimes result from the flow dynamics and are thus unknown before the computation.

3.2.1.2 Solution-based target mesh size

To avoid the need for an a priori knowledge of the flow features, it is helpful to combine geometric criteria with solution-based mesh refinement. While different metrics can be considered, the present work relies on the Froebenius norm of the local velocity gradient tensor, $||\nabla \mathbf{u}||$. The goal is to prescribe a small mesh size in regions with large velocity gradients, and vice versa.

In the proposed approach, the target mesh size is based on a linear interpolation of a negative power

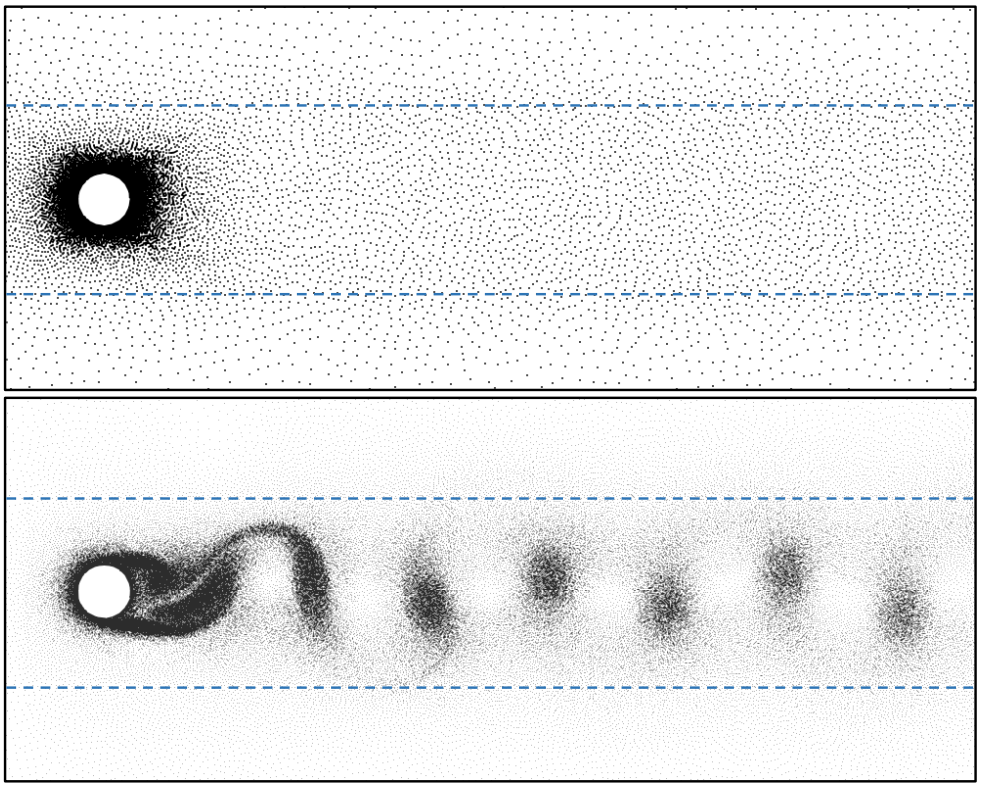


Figure 3.4: Non-uniform meshes for the simulation of the flow around a fixed cylinder at $Re_D = 200$. Only the nodes are shown, and each image corresponds to a specific instant in time after a limit cycle oscillation has been reached. The first mesh (top) uses only geometrical mesh refinement based on the distances from the cylinder and from the center line. The second mesh (bottom) also uses this refinement but, in addition, includes a solution−based refinement where the target mesh size is given by Eq. (3.23) with the parameter $\beta = \frac{1}{3}$. The dashed blue lines represent the rectangular area for geometric refinement. Note that for better visualization of the most refined regions, the point size is twice smaller in the bottom image.

of $||\nabla \mathbf{u}||$,

$$\begin{aligned} L^* =& L^*_{\min} + (L^*_{\max} - L^*_{\min}) \\ & \times \max\left(\min\left(\frac{||\nabla \mathbf{u}||^{-\beta} - ||\nabla \mathbf{u}||^{-\beta}_{\max}}{||\nabla \mathbf{u}||^{-\beta}_{\min} - ||\nabla \mathbf{u}||^{-\beta}_{\max}}, 1\right), 0\right) \end{aligned} \tag{3.23}$$

where β, $||\nabla \mathbf{u}||_{\min}$ and $||\nabla \mathbf{u}||_{\max}$ are user-defined parameters. The parameter β, usually between zero and one, controls the rate of increase of the mesh size, while the two other parameters are threshold values that can be estimated beforehand based on characteristic length and velocity scales of the problem considered or known solutions of similar problems. The rationale behind Eq. (3.23) is to retrieve a classical grid stretching for a wall boundary layer, but other functional forms could also be considered.

An example of this technique is shown in the bottom image of Fig. 3.4, where, in addition to the previously mentioned geometrical refinement, a solution-based refinement is used. In this case, a small target mesh size is imposed, in accordance with Eq. (3.23), in regions of large gradients such as in the wake of the cylinder. In particular, a high grid resolution can be seen in and around the von Karman vortices and shear layers. Such grid refinement allows a more accurate prediction of drag, lift and Strouhal number, as discussed in more detail in Chapter 4.

In flows where features of interest have widely different levels of velocity gradient magnitude, Eq. (3.23) might not provide an adequate target mesh size. For instance corner vortices might be much weaker than other flow features but still important to capture. In this case, it is better to replace $||\nabla \mathbf{u}||$ in the above expression by a normalized measure of the velocity gradient magnitude, e.g.,

$$||\widetilde{\nabla \mathbf{u}}|| = \frac{||\nabla \mathbf{u}||}{||\mathbf{u} - \mathbf{U}_\infty|| + U_\epsilon}, \tag{3.24}$$

where U_ϵ is a small user-chosen velocity to avoid division by 0, and $\mathbf{U}_\infty$ is the constant free-stream velocity. Note that other options can be considered depending on the case of interest.

For problems involving free-surface deformations, a finer mesh is also needed to accurately capture the dynamics and location of free surfaces. Therefore, we also rely on a solution-based measure related to the free-surface deformation. In particular, the local target mesh size is prescribed as a linear function of the free-surface radius of curvature r_s to ensure an approximately constant angular mesh size:

$$L^* = \min\left(\max\left(\frac{\pi r_s}{m}, L^*_{\min}\right), L^*_{\max}\right), \tag{3.25}$$

where m is a user-defined parameter that controls the angular mesh resolution, i.e., the number of edges over a half-circle. The radius of curvature itself is obtained as the radius of the circle passing through the corresponding boundary node and its two direct neighbors on the free surface. The target mesh size of neighbor nodes in the fluid bulk can be prescribed in a second step using the smoothing algorithm introduced in the next subsection.

Finally, it should be mentioned that such solution-based approaches, or similar ones, can partly replace and/or complement some of the aforementioned geometric criteria. Equation (3.22) is then used to prescribe the actual local target mesh size. This is illustrated in Fig. 3.4, where the geometric criteria are combined with the solution-based refinement approach of Eq. (3.23) to better resolve the von Karman vortices.

3.2.1.3 Smoothing of the target mesh size

For numerical accuracy of the discretization, but also for applying the α-shape technique locally, it is important to ensure a certain smoothness of the mesh size. In particular, small and large elements should not be directly adjacent to each other. This implies that the target mesh size should also be sufficiently smooth. While rapid variations of L^* can be easily avoided for geometric criteria, this is less trivial for solution-based approaches. If variations in space and/or time of velocity gradients are too rapid, discontinuities in the target mesh size can occur. The refinement of the free surface is another example because the curvature-based target mesh size can only be prescribed for the boundary nodes, and not for their direct and more distant neighbors.

A smoothing algorithm is thus subsequently applied to the target mesh size, so as to enforce a maximum target mesh size ratio p_r between two neighbors. More specifically, if the condition

$$\frac{1}{p_r} \leq \frac{L^*_m}{L^*_n} \leq p_r \tag{3.26}$$

is not satisfied for a node n and its neighbor node m, then the larger target mesh size is reduced to satisfy Eq. (3.26). Starting from nodes with the smallest target mesh size, the neighbors are, if needed, gradually updated. The process is repeated recursively for all nodes that have been themselves updated. Note that the complexity of the algorithm is of order N, as the above condition is tested for each mesh node a finite number of times that depends only on the number of direct neighbour nodes.

This smoothing step also allows refining the mesh around solid boundaries with a complex shape, in a similar manner as it is done with the curvature-based refinement of the free surface. In this case, the target mesh size can be imposed on the boundary itself and the smoothing algorithm can be leveraged to "propagate", with some progression factor, the target mesh size into the fluid domain.

3.2.1.4 Delaying of the mesh coarsening

After a mesh refinement, it might be sometimes advantageous to delay any potential subsequent coarsening. This could be to avoid a repeating cycle of node creation and destruction that would introduce unnecessary numerical dissipation.

Another situation in which delaying mesh coarsening might be required is when the need for a fine mesh is anticipated but the local target mesh size provided by the refinement criteria is larger than required. This is for instance the case during the merging of two fluid regions, as illustrated by the falling drop in Fig. 2.5. In such a case, one of the regions (e.g., the droplet) might be much more refined than the other (e.g., the liquid bath). As predicted by the α-shape criterion, merging typically

occurs when the distance between the two regions is of the same order as the mesh size of the coarser fluid region. Therefore, an anticipatory local mesh refinement around the liquid bath free surface, when the droplet approaches, allows reducing mass creation. To ensure that the mesh resolution remains sufficient before the actual merging, despite a larger calculated target mesh size, the coarsening that would otherwise take place should be delayed for a duration Δt_{delay} until the merging process has started.

This is achieved by assigning to each node a time counter t^* that is incremented at each time step, but reset to zero every time the new calculated target mesh size is smaller than the current one. If the new calculated target mesh size is larger and if $t^* < \Delta t_{\text{delay}}$, then the current target mesh size at the node is not updated and its old value is kept, thus preventing a potential coarsening during the time step. More detail on how the delay time Δt_{delay} is actually computed are given later in Section 3.2.3.2.

3.2.2 Addition and removal of nodes

Once a smooth target mesh size has been defined everywhere, it must be enforced up to some tolerance by adapting the actual mesh. In practice, this is achieved by creating/destroying nodes wherever the actual mesh size L is outside a user-defined range around its target value.

As explained in the following section, the comparison between target and actual mesh size is usually performed at the edge or, for boundary elements, at the element level. Because the target mesh size L^*_n is initially defined at the nodes n, we also define a target mesh size L^*_{elem} associated with an element and L^*_{edge} with an edge using a simple arithmetic average over the corresponding nodal values,

$$L^*_{\text{elem}} = \frac{1}{3} \sum_{n_{\text{elem}}} L^*_{n_{\text{elem}}} , \tag{3.27}$$

$$L^*_{\text{edge}} = \frac{1}{2} \sum_{n_{\text{edge}}} L^*_{n_{\text{edge}}} . \tag{3.28}$$

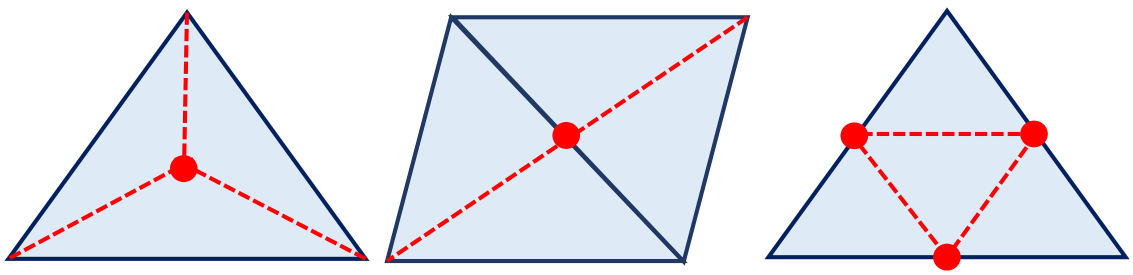
where n_{elem} and n_{edge} indicate the corresponding nodes of the element and edge, respectively.

3.2.2.1 Adding nodes

In general, new nodes can be added either within elements (Fig. 3.5(a)) or on their edges (Fig. 3.5(b,c)). In the present algorithm, nodes are only added at the mid-point of selected edges to avoid new elements that are too obtuse [64] and a too agressive mesh refinement, i.e., a too large increase in nodal density[5], from one time step to the next. Nonetheless, the addition of nodes in the center of elements could in principle also be considered.

The main idea is to divide elements in the bulk that are too large by a factor of two through the addition of a new node on only one of their two largest edges (Fig. 3.5(b)), and elements at the boundary

[5] In 2D, the nodal density equals $1/(2A_{\text{elem}})$, where A_{elem} is the surface area of the element. This notion is explained in more detail in Section 3.4.1



(a) $\times$ 3 **(b)** $\times$ 2 **(c)** $\times$ 4

Figure 3.5: Different strategies to add nodes and corresponding increase of the nodal density: one node added at the center of the element (left), one node added at the mid-point of a single edge (middle), nodes added at the mid-point of each edge of an element (right). The new nodes are indicated by red circles and the new virtual edges by red dashed lines.

by a factor of four through the addition of new nodes on each of their edges (Fig. 3.5(c)). The special treatment of boundary elements is dictated by the boundary recognition step. In order to correctly identify the boundary using the α-shape technique, and thus reduce the mass conservation error, it is crucial to ensure high-quality boundary elements. This is discussed in more detail in section 3.2.3. It is also important to emphasize that the new edges (red dashed lines in Fig. 3.5) are only virtual because the mesh is discarded and a new triangulation is performed right after the node creation/destruction step).

Concretely, an edge in the bulk is refined by adding a node at its mid-point if the three following conditions are all satisfied:

- the average area of the two elements sharing this edge is larger than the threshold value:

$$\frac{1}{2}(A_{\text{elem}_1} + A_{\text{elem}_2}) > \frac{4}{3}L^{*2}_{\text{edge}}\,, \tag{3.29}$$

- the edge is not the shortest edge of any of the two elements sharing this edge, and
- the edge has not been previously tagged to prevent its refinement.

The addition of a node on an edge impacts the two adjacent elements sharing this edge. To avoid the further division of these elements through the potential addition of new nodes on their other edges, these other edges are then tagged. This tag prevents their refinement, thereby ensuring that elements in the bulk are at most divided by two. Additionally, if the edge that is refined belongs to a boundary element (i.e., an element with an edge on the boundary), this boundary element is automatically refined by adding nodes on all its edges irrespective of its size (Fig. 3.5(c)). The rationale is to ensure that the shape of the boundary elements is as optimal as possible to improve the boundary detection in the subsequent step, as is illustrated and explained in Fig. 3.6 (see also Section 3.2.3)

On the other hand, if the edge is on the boundary and if the area of the associated boundary element is larger than the threshold value,

$$A_{\text{elem}} > \frac{4}{3}L^{*2}_{\text{elem}}\,, \tag{3.30}$$

then this boundary element is divided by four. This is achieved by adding a new node on each of the edges of that element (Fig. 3.5(c)). Furthermore, if this boundary element has a direct neighbor (i.e., sharing a common edge) that is also a boundary element (i.e., with an edge on the boundary), this neighbor element is also refined irrespective of its size.

The algorithm is illustrated in Fig. 3.7. It should be noted that, with this algorithm, the refinement has some dependence on the order in which the edges are considered. This is however not deemed to be a problem, especially since there are anyway many different strategies to refine the mesh. The threshold coefficient 4/3 in Eqs. (3.29) and (3.30) is chosen so that, on average, the elements that are refined have afterwards an area larger than $\frac{2}{3}L^{*2}$ in the bulk and $\frac{1}{3}L^{*2}$ at the boundaries, and the unrefined elements an area smaller than $\frac{4}{3}L^{*2}$. Increasing this coefficient reduces the number of nodes added but induces a larger discrepancy between actual and target mesh size, and conversely.

Once nodes have been added, the associated nodal values of physical quantities (e.g., velocity, pressure, etc.) must be prescribed. This initialization uses the interpolation shape function of the finite element discretization. In practice, it implies that each variable is simply obtained using the mean value of the corresponding variable at the nodes of the refined edge. This ensures that the local fields remain unchanged during the process.

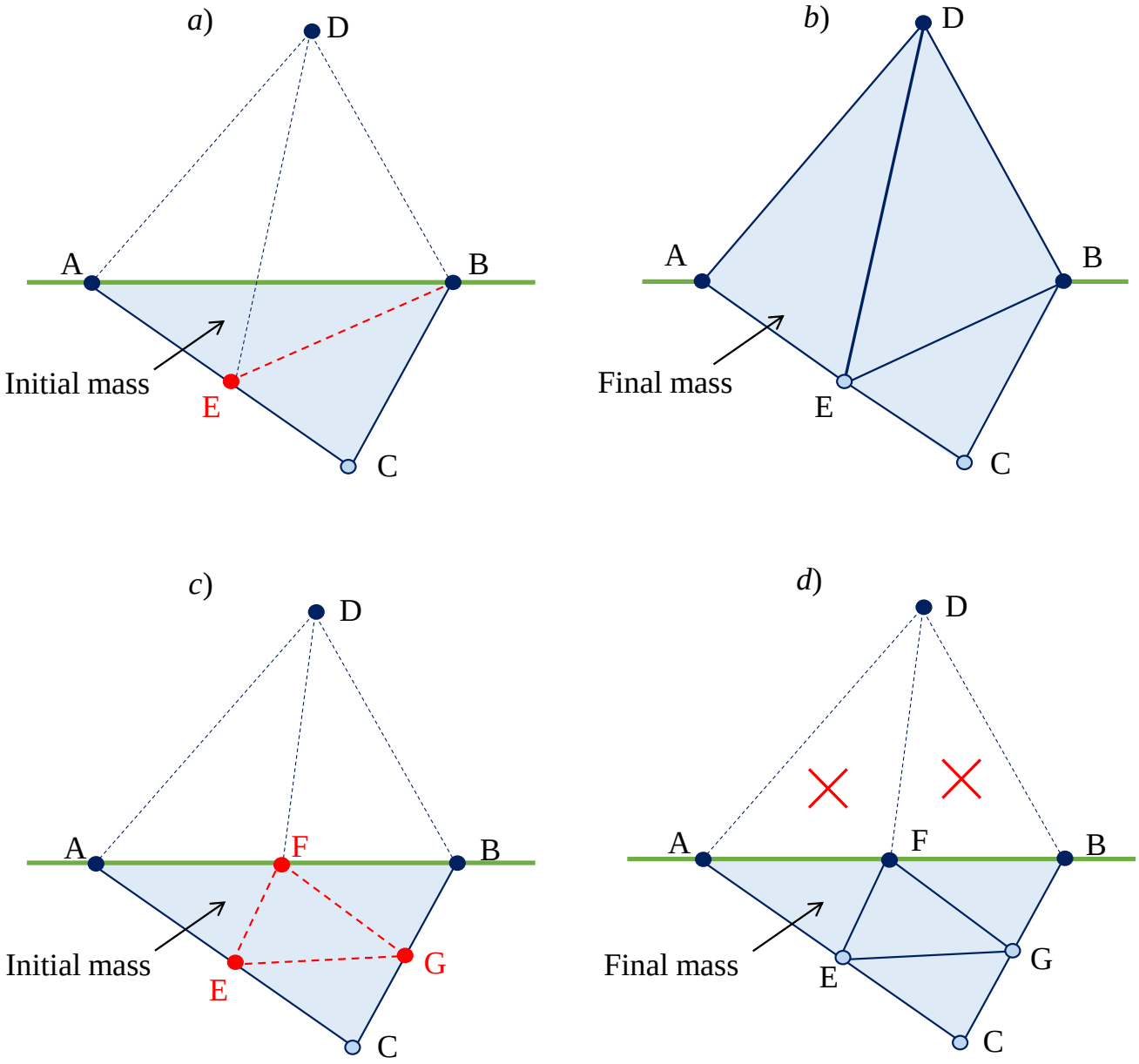


Figure 3.6: Motivation for the special treatment of boundary elements when a non-boundary node (node E) is added in the middle of an immersed edges ([AC]). If nodes are not added on the other edges of the element [ABC] (diagrams *a* and *b*), the triangulation may keep spurious triangles [AED] and [EBD] because of the new interior node E (as explained in Section 3.2.3, the criteria to remove triangles containing a non-boundary nodes are not all easily satisfied). Therefore, the algorithm also adds nodes F and G, where F is on the boundary (diagram *c* and *d*). This avoids spurious creation of mass by correctly removing triangles [AFD] and [FBD], as they have all their nodes on a boundary (as also explained in Section 3.2.3, the criteria to remove triangles containing only boundary nodes are more easily satisfied). The original boundary of the fluid is represented by the thick green line.

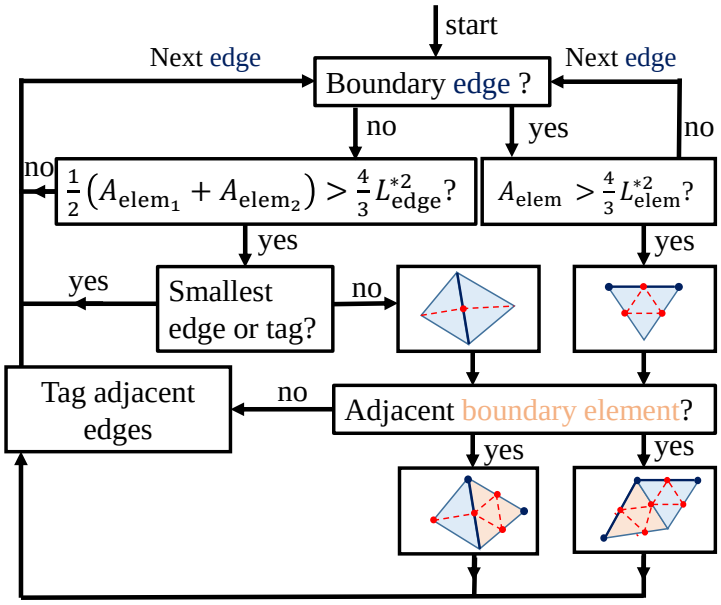


Figure 3.7: Algorithm for node addition. The edge considered is indicated by the bold dark blue line. The dark blue dots represent boundary nodes while the red dots and the red dashed lines are respectively the new nodes and the new virtual edges if the edge is refined.

3.2.2.2 Removing nodes

For the same reason as mentioned above, node removal is implemented at boundaries differently than in the bulk. In the bulk, the main idea is to "collapse" an element that is too small into a single node at its center. In other words, the nodes of a triangle whose area is smaller than the threshold,

$$A_{\text{elem}} < \gamma L^{*2}_{\text{elem}}\,, \tag{3.31}$$

are deleted and a new node at the element center is created. The new (virtual) elements have thus all more or less the same size and are larger than the original elements, which ensures a smoother transition between coarse and fine mesh.

This new node is assigned nodal values corresponding to the average over the three deleted nodes. Deleting the nodes of the element has a direct impact on its neighbors that share these nodes. More specifically, it is important to realize that collapsing all elements of a mesh region would increase the nodal density rather than decrease it[6]. Therefore, the direct neighbor elements and their own direct neighbors are prevented to be themselves collapsed, irrespective of their size. An example is shown in Fig. 3.8 where the direct neighbors of the collapsed element 10 and their respective neighbors, indicated by green circles, cannot be collapsed. It can be shown that, for a uniform mesh of equilateral triangles, this coarsening process corresponds to increasing the area of the elements by a factor 4/3.

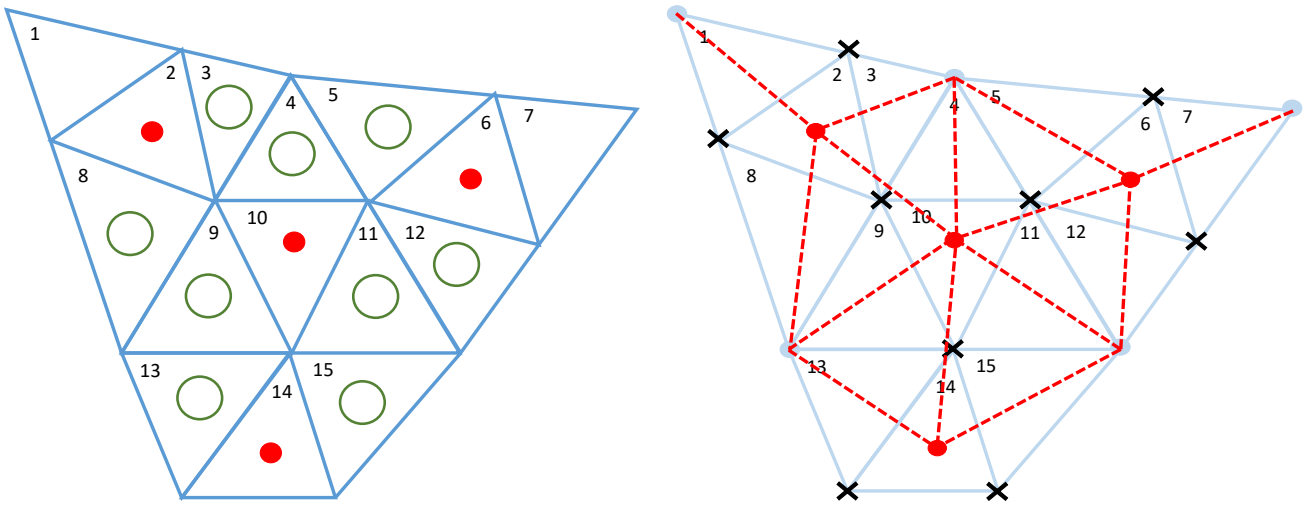


Figure 3.8: Illustration of the coarsening process of a mesh region in the bulk, i.e., without boundary elements, where the triangular elements are numbered from 1 to 15. (Left) If element 10 is collapsed, then all its neighbors and their respective neighbors (green circles) are tagged to prevent them from being collapsed. On the other hand, elements 2, 6 and 14 can be collapsed. Note that collapsing all elements would not decrease the nodal density. (Right) Resulting nodes and virtual mesh after coarsening where the red dots indicate the new nodes, the black crosses the nodes that have been deleted and the red dashed lines the corresponding new virtual triangulation.

The special coarsening algorithm at the boundary is applied to all elements with at least one node (and not only an edge as for node addition) on the boundary. First, the actual element area A_{elem} used in the criterion for coarsening, Eq. (3.31), is replaced by $\frac{1}{2}h^2_{\text{min}}$,

$$\frac{1}{2}h^2_{\text{min}} < \gamma L^{*2}_{\text{elem}}\,, \tag{3.32}$$

where h_{min} is the length of the smallest edge of the element. This is to avoid too skewed boundary elements, as illustrated in Fig. 3.9. Then, the smallest edge of the element rather than the element itself is collapsed. More specifically, if either both nodes of the smallest edge are on the boundary or none of them is, then the two nodes are deleted and replaced by a single node at their mid-point. The average value of the physical quantities at the two deleted nodes is assigned to the new node. If the shortest edge involves a boundary node and an interior node, then this interior node is simply deleted. This minimizes the impact of the coarsening on the shape of the boundary.

[6] discussed in Section 3.4.1, a triangular mesh has more or less twice more elements than nodes so that collapsing each element into a node at its center would approximately double the number of nodes, and thus the nodal density.

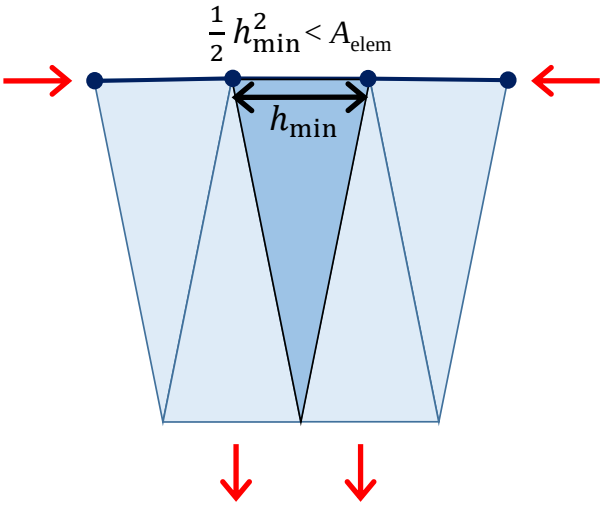


Figure 3.9: Illustration of a squeezed boundary (e.g., free-slip boundary) where the boundary nodes are represented by the dark blue dots and the boundary edges by the thick dark blue lines. If the free surface is squeezed, boundary elements are compressed along, and extended perpendicularly to, the free surface, as indicated by the red arrows. This would lead to highly skewed elements with a surface area A_{elem} sufficiently large that the coarsening criterion, Eq. (3.31), is not satisfied. The modified criterion, Eq. (3.32), reduces the threshold for coarsening and thus avoids that boundary elements become too skewed.

The choice of the threshold coefficient γ in Eq. (3.31) is important to avoid a continuously alternating mesh refinement and coarsening in the same region. In particular, if γ is too large, new nodes created at the previous iteration might be deleted because their associated element is now too small, or conversely, elements that have been coarsened might be refined at the next time step. Moreover, as the coarsening step averages locally the solution, it introduces unwanted numerical diffusion that should be minimized. It is thus suggested to impose an upper bound, $\gamma < 1/2$, such that the coarsened mesh in the bulk is, on average, smaller than $\frac{2}{3}L^{*2}$ to avoid overlapping of refined and coarsened elements in the same region. In practice the parameter γ is chosen between $1/3$ and $1/2$ in the bulk. Reducing its value leads to more variability in the element size, potentially slightly more deformed elements but fewer node eliminations, and conversely. At boundaries the increase of nodal density, if particles are added, is twice that in the bulk. Therefore, the parameter γ in Eq. (3.31) is there divided by two and a smooth transition between $\gamma/2$ at the boundary and γ in the bulk is imposed over a few cells.

The impact of the threshold value γ on mesh quality and possible numerical dissipation is further illustrated through a specific example. Mesh quality is quantified by the triangle distortion parameter λ [1],

$$\lambda = \sqrt{3}\frac{r_c}{h_{\min}}\,, \tag{3.33}$$

where r_c is the radius of the circumscribed circle of the element considered and $h_{\min}$ its shortest edge. With this definition, λ is equal to one for an equilateral triangle and becomes large for badly-shaped elements (high aspect ratio). On the other hand, because numerical dissipation induced by node deletion is challenging to calculate, it is here assessed by monitoring the number of nodes added and deleted.

The example considered is that of a cylinder of diameter D rising towards and crossing a free surface at constant velocity U. This case is analyzed in depth in Chapter 5. The computational domain is kept relatively small to focus on the region where mesh adaptation takes place (see Fig. 3.10). The main parameters are summarized in Table 3.1. Mesh adaptation is performed using the novel algorithm based on two geometrical and one solution-based criteria. The first criterion prescribes $L^*_{(1)}$ to increase from $L^*_{\min}$ at the cylinder surface to $L^*_{\max}/5$ one diameter or more away. The second criterion imposes $L^*_{(2)} = L^*_{\max}/5$ within a distance $D/2$ of the vertical centerline, $L^*_{(2)} = L^*_{\max}$ for nodes further than $5D$ away from this centerline and a linear increase in-between. Finally, the third criterion is based on Eq. (3.23) with $\beta = 1/3$, $||\nabla \mathbf{u}||_{\min} = 0.15$ and $||\nabla \mathbf{u}||_{\max} = 4$. The actual target mesh size is finally obtained as the minimum over all three criteria, as given by Eq. (3.22). Examples of the mesh at $t = 0.08$ (the initial mesh is not shown because the effect of the geometrical refinement slightly modifies the mesh within a few time steps) and $t = 2.5$ (during the interface crossing) are shown in Fig. 3.11.

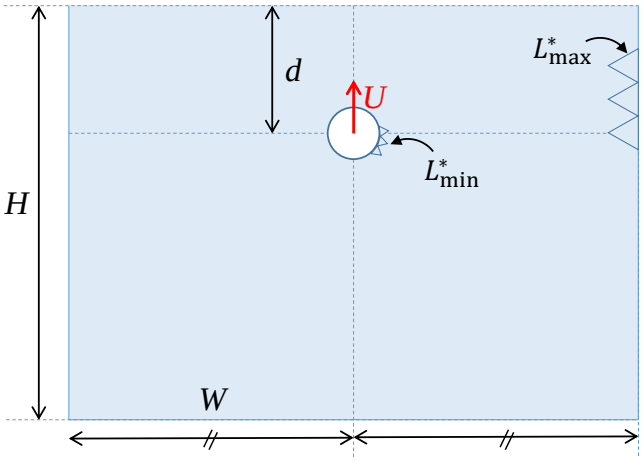


Figure 3.10: Geometrical setup for the evaluation of the influence of the threshold value γ used for node deletion. The value of the different parameters is given in Table 3.1.

H	8
W	5.5
d	2.5
$r = g^{-1}$	1
$Re = \nu^{-1}$	50
L^*_{min}	0.02
L^*_{max}	0.66
Δt	0.02
β	1/3
$\|\|\nabla \mathbf{u}\|\|_{\text{min}}$	0.15
$\|\|\nabla \mathbf{u}\|\|_{\text{max}}$	4

Table 3.1: Simulation parameters for the evaluation of the influence of the threshold value γ used for node deletion. All quantities are non-dimensionalized using the cylinder velocity U and diameter D. The geometrical parameters are defined in Fig. 3.10.

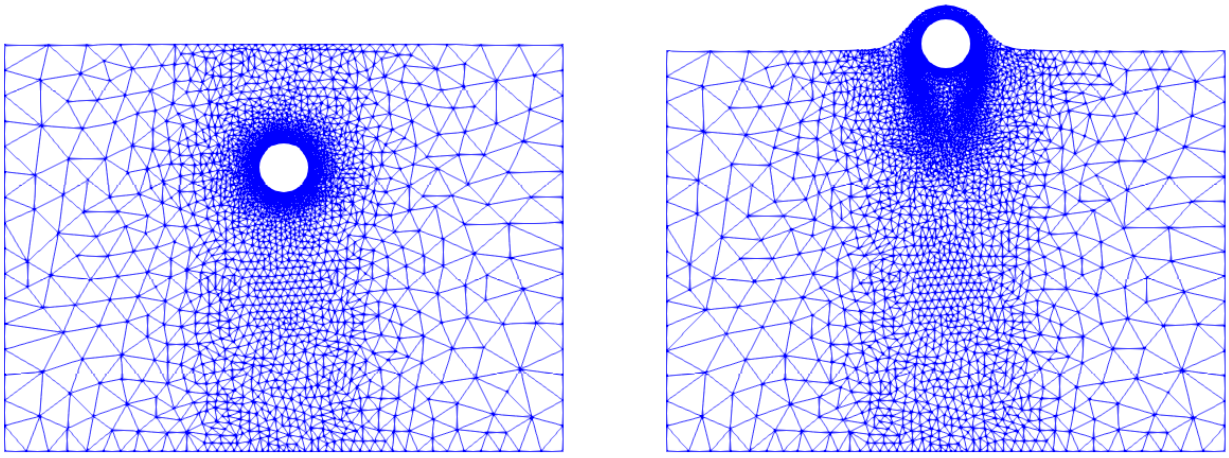


Figure 3.11: Illustration of the mesh at t=0.08 (left) and at t=2.5 (right) for the case where $\gamma = 7/18$. The geometrical refinement is always visible around the center line and around the cylinder (left and right) while the effect of the solution-based refinement is later visible in the wake of the cylinder (right).

The evolution of the average and standard deviation of the triangle distortion parameter λ along the simulation is shown for different values of the node deletion threshold parameter γ in Fig. 3.12. In the figure, y^* is the vertical position of the cylinder center and $y^* = 0$ corresponds to the initial free-surface vertical position. One can observe that the mesh element quality is on average good. Additionally, after an initial rapid increase of λ following the impulsive start of the cylinder, the distortion parameter seems to slowly and non-monotoneously increase until the cylinder has fully emerged, and then to decrease. More importantly, and as expected, the mesh distortion is found to be larger for smaller values of γ, as nodes are less often eliminated (and thus also new nodes are less frequently added). This latter point is clearly demonstrated in Fig. 3.13, which shows the percentage of nodes that are deleted (left) and added (right) along the simulation. Overall, one can thus expect more numerical dissipation (more nodes deleted) but a slightly better mesh quality (lower mesh distortion) with a large value of the threshold γ, and conversely, such that a trade-off has to be made.

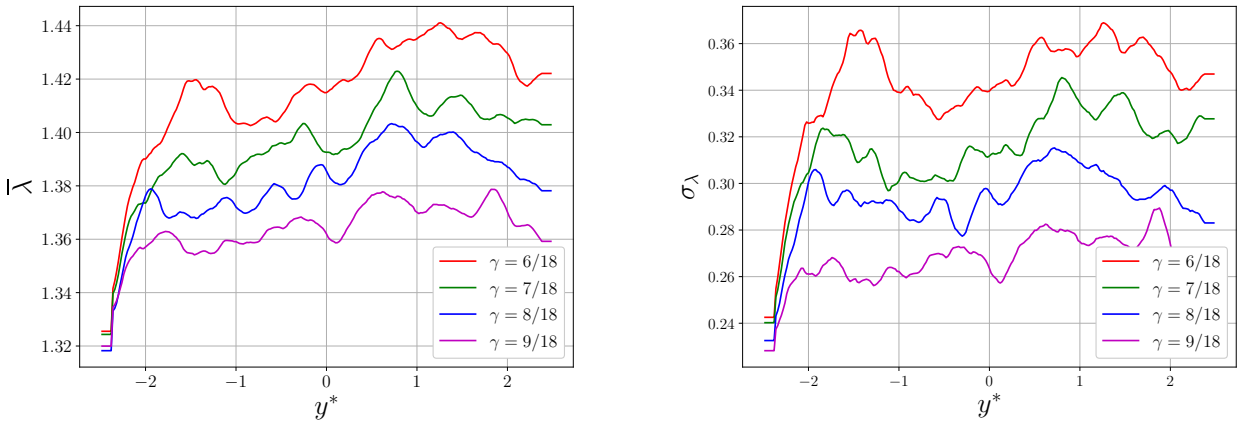


Figure 3.12: Average (left) and standard deviation (right) of the mesh distortion parameter as a function of the position y^* of the cylinder center ($y^* = 0$ corresponds to the initial free-surface position) for different values of the node deletion threshold γ. The statistics are computed over all mesh elements and are filtered with a running average over ten time steps to reduce the noise.

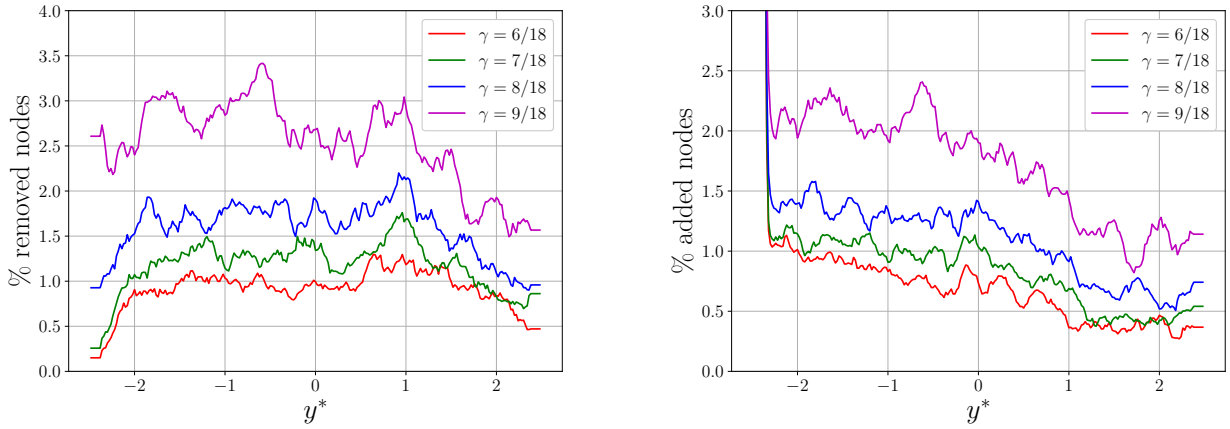


Figure 3.13: Fraction of nodes deleted (left) and added (right) as a function of the position y^* of the cylinder center ($y^* = 0$ corresponds to the initial free-surface position) for different values of the node deletion threshold γ. The statistics are computed over all mesh elements and are filtered with a running average over ten time steps to reduce the noise.

3.2.3 Boundary detection algorithm

Once the node addition/destruction step has been completed, the old mesh can be discarded and a new Delaunay triangulation is performed. Because the full convex hull of the domain is triangulated, mesh elements that do not belong to the fluid must be discarded.

This boundary recognition step relies both on boundary tracking and a local α-shape technique. Concretely, nodes belonging to the boundary at the previous time step are tagged and the global characteristic length scale h in Eq. (2.43) is replaced by the local target mesh size. In particular, the α-criterion is modified to

$$r_c < \alpha L^*_{\text{elem}} \,, \tag{3.34}$$

where the user-defined constant α should take a value between 1.2 and 1.8 depending on the case. Note that, because of the difference in the definition of h and L^*_{elem}, the typical value $\alpha = 1.2$ in Eq. (2.43) corresponds approximately to $\alpha = 1.8$ in Eq. (3.34) for an equilateral triangle. However, the boundary tracking allows using a much lower value of α, so that in practice $\alpha = 1.2$ in Eq. (3.34) provides good results. The influence of α is further discussed in Chapter 4.

As already illustrated in Figs. 2.5 and 2.6, the mass conservation error is directly linked to the boundary identification algorithm. To minimize this error, three important aspects must be ensured: 1) a sufficiently refined mesh in the boundary vicinity, 2) well-shaped elements, and 3) the knowledge of which nodes belonged to the boundary prior to remeshing (boundary tracking). These requirements explain the aforementioned special treatment of boundary elements during node addition/destruction.

Because elements with at most one node on the boundary can be safely considered as being part of the fluid domain, the α-shape criterion is only applied to elements with two or three nodes tagged during the previous time step as belonging to the boundary. This reduces the computational cost and simultaneously prevents the unphysical destruction of fluid elements in the fluid domain.

3.2.3.1 Keeping or discarding boundary elements

An element with two nodes on the boundary is only eliminated if the following three conditions are all met:

- the local α-criterion, Eq. (3.34), is violated,
- the element edge on the boundary is the longest edge of its element ($l_{\text{b.e.}} = l_{\max}$), and
- the element area is smaller than a minimum value imposed by the global minimal target mesh size: $A_{\text{elem}} < \frac{1}{2} L^{*2}_{\min}$.

The third requirement ensures that the element is removed from the triangulation only if the associated error on mass conservation is of the order of the finest mesh resolution. On the other hand, if only the first two conditions are satisfied, the boundary is likely locally stretched. In this case, the node that is not on the boundary is tagged to be removed and the boundary edge is tagged to be refined at the next time step, as illustrated in Fig. 3.14a. The motivation for this refinement of the boundary is explained and illustrated in Fig. 3.15.

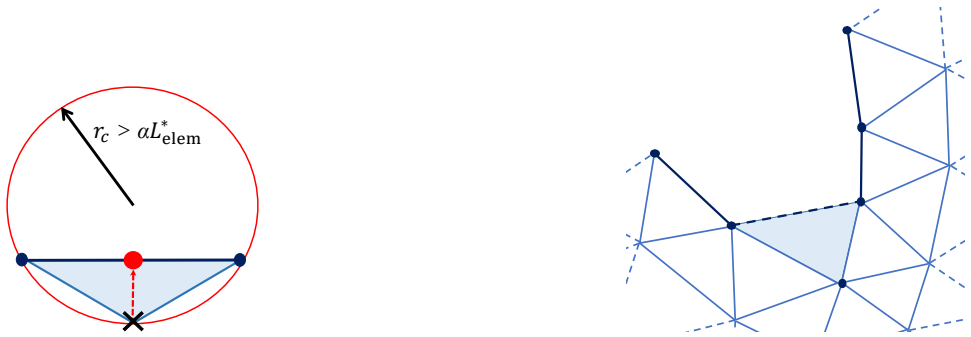


Figure 3.14: Illustration of two different situations encountered during the application of the boundary recognition algorithm. (**a**) Addition of a new node (red dot) at the mid-point of the longest edge of an obtuse element when this edge (thick dark blue line) is located on the boundary and the corresponding element violates the α-shape criterion, Eq. (3.34). The black cross indicates the node that is tagged for removal. (**b**) Illustration of mass creation when a free surface folds on itself. At the previous time iteration, the blue shaded triangle had been discarded by the boundary recognition algorithm and has thus three nodes on the boundary (dark blue dots). After remeshing, this element would be kept if it satisfies the α-shape criterion. To minimize the resulting mass creation, the element is only kept if, additionally, it is of the same size as its direct neighbors. The dark blue dashed line indicates in this case the new boundary edge.

Elements with three nodes on the boundary should a priori be empty elements and thus discarded. Notable exceptions are elements at domain "corners" (see Fig. 2.6) and elements in a thin film whose thickness corresponds to the element size (i.e., only one element thick). Such elements also play a key role during the merging phase of two fluid regions, as already shown in Fig. 2.5. They are thus discarded of the mesh as soon as one of the following two conditions is satisfied:

- the local α-criterion, Eq. (3.34), is violated, or

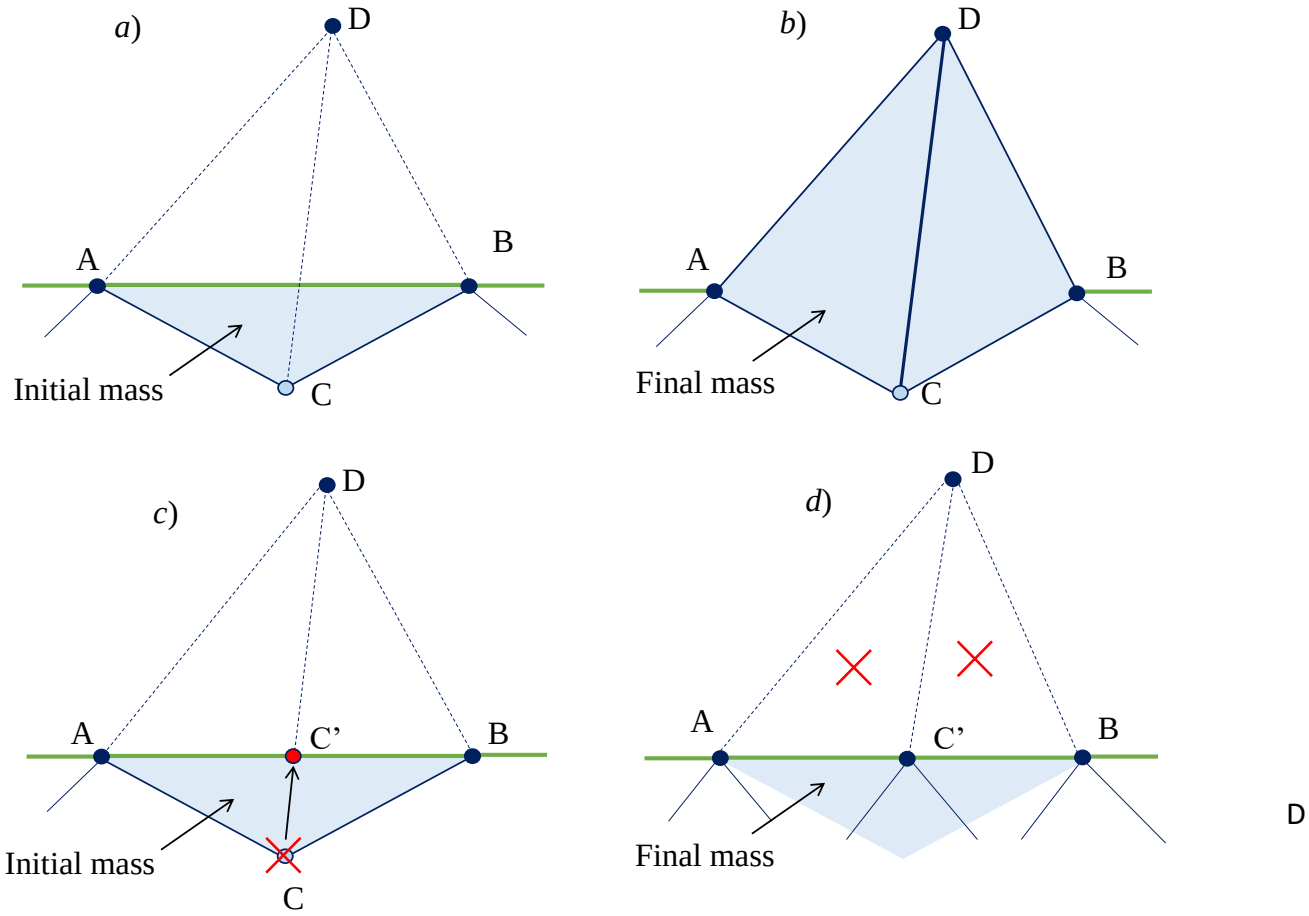


Figure 3.15: Motivation for the refinement of boundary edges when a non-boundary node (node C) closely approaches the free surface. If nothing is done (diagrams *a* and *b*), the triangulation finally leads to the creation of triangles [ACD] and [CBD]. To avoid the associated spurious mass creation (diagrams *c* and *d*), at earlier time when the triangulation still includes triangle [ABC], the interior node C is replaced by the boundary node C', which enables to correctly remove triangles [AC'D] and [C'BD], as both have all their nodes on a boundary. The original boundary of the fluid is represented by the thick green line.

- their area is significantly larger than that of one of their direct neighbors. In practice, this condition is expressed as

$$A_{\text{elem}} > 2\left(\min_n(L_n^*)\right)^2, \tag{3.35}$$

  where the index n corresponds to the three nodes of the element and the factor 2 has been chosen arbitrarily to ensure that the element size is sufficiently different from that of the neighbors.

The purpose of this second condition is to minimize the spurious creation of mass, as illustrated in Fig. 3.14b for a free surface folding on itself. A new fluid element is only created if its size is comparable to that of its smallest neighbor. Note that, in practice, the curvature-based refinement of the free surface ensures that this new element has a size of the order of the global minimum mesh size.

The merging of two boundaries with a significant difference in their corresponding mesh resolution is another example to illustrate the use for this second condition, as shown in Fig. 3.16. If a significant difference is identified among the target mesh sizes L_n^* at the nodes of the element connecting the two boundaries, this element is discarded and marked as potential "collision element". Additionally, the surface with the coarser mesh is imposed a new target mesh size that matches the resolution of the surface with the finer mesh, and this new target mesh size is propagated into the bulk through smoothing, as explained in more detail in the next section.

Once all the elements that are considered non-fluid elements have been removed, the new boundary can be identified. In particular, all the nodes that define edges belonging to a single element are tagged as boundary nodes for the next time iteration. The overall algorithm for boundary detection is summarized in Fig. 3.17

3.2.3.2 Collision anticipation algorithm and local delaying of mesh coarsening

Once all "collision elements" have been identified, the new target mesh size of these elements, corresponding to the mesh size of the more resolved of the two merging surfaces, is propagated gradually from node to node through smoothing (see Section 3.2.1.3) in two steps: a first smoothing is only applied to free-surface nodes while another smoothing is subsequently applied to propagate the new free-surface target mesh size through the bulk. As already mentioned in Section 3.2.1.4, this way of imposing the target mesh size possibly conflicts with the minimum target mesh size prescribed by the geometry and solution-based criteria (see Eq. (3.22)). To avoid a direct coarsening of the newly refined surface region at the next time step, any possible mesh coarsening in this region is delayed until collision occurs, as described in Section 3.2.1.4. The delay time ΔT_{delay} is thus defined as the time until collision.

The time until collision, ΔT_{c}, is estimated from the relative position and velocity of the two surfaces about to merge. For instance, consider a potential "collision element" with one node s on the finer surface and two nodes S1 and S2 on the coarser surface. The relative position, $\mathbf{x}_{\text{rel,s}}$, and velocity, $\mathbf{v}_{\text{rel,s}}$, of node s with respect to the edge defined by nodes S1 and S2 are given by

$$\mathbf{x}_{\text{rel,s}} = \mathbf{x}_{\text{s}} - \frac{1}{2}\left(\mathbf{x}_{\text{S1}} + \mathbf{x}_{\text{S2}}\right) , \tag{3.36}$$

$$\mathbf{v}_{\text{rel,s}} = \mathbf{v}_{\text{s}} - \frac{1}{2}\left(\mathbf{v}_{\text{S1}} + \mathbf{v}_{\text{S2}}\right) , \tag{3.37}$$

$$\tag{3.38}$$

and the collision angle is

$$\theta = \arccos\left(\frac{|\mathbf{v}_{\text{rel,s}} \cdot \mathbf{L}|}{||\mathbf{v}_{\text{rel}}||\,||\mathbf{L}||}\right) , \tag{3.39}$$

where $\mathbf{L}$ is the vector connecting the two nodes S1 and S2, i.e., along the surface edge (see Fig. 3.18a).

Two cases can be considered. If the collision angle $\theta > 15°$, then the collision takes place at relatively high incidence, and the time until collision can be estimated as

$$\Delta t_c \approx -\frac{\mathbf{x}_{\text{rel,s}} \cdot \mathbf{x}_{\text{rel,s}}}{\mathbf{v}_{\text{rel,s}} \cdot \mathbf{x}_{\text{rel,s}}} . \tag{3.40}$$

This time is imposed as coarsening time delay, Δt_{delay}, for all surface nodes whose target mesh size has been decreased because of this potential collision.

In case of a low incidence collision (see Fig. 3.18b), the collision is likely to take place at surface elements further away (more left in Fig. 3.18b) from the "collision element" considered. A varying time to collision is thus used for the different surface nodes i marked for refinement,

$$\Delta t_{c,i} = \frac{\mathbf{x}_{\text{rel},i} \cdot \mathbf{x}_{\text{rel},i}}{\mathbf{v}_{\text{rel,s}} \cdot \mathbf{x}_{\text{rel},i}} , \tag{3.41}$$

where

$$\mathbf{x}_{\text{rel},i} = \mathbf{x}_i - \frac{1}{2}\left(\mathbf{x}_{\text{S1}} + \mathbf{x}_{\text{S2}}\right) , \tag{3.42}$$

such that coarsening is delayed longer for nodes closer to the expected collision point.

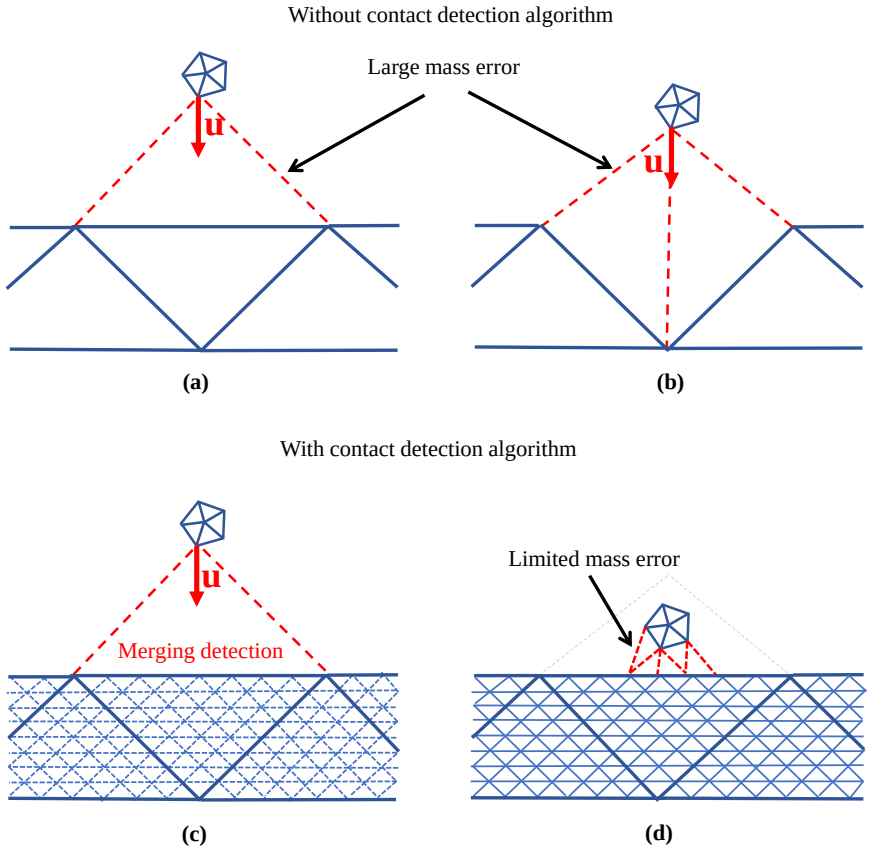


Figure 3.16: Merging of two fluid regions without $(\mathbf{a}, \mathbf{b})$ and with $(\mathbf{c}, \mathbf{d})$ merging detection algorithm. $(\mathbf{a})$ When the more refined region is sufficiently close to the other boundary, the connecting element (red dashed lines) is not discarded by the α-shape criterion, leading to significant mass creation. Considering the second condition based on Eq. (3.35) eliminates this connecting element from the fluid region and, thus, prevents this artificial mass creation. $(\mathbf{b})$ However, when the more refined region becomes closer to the coarse boundary, Delaunay triangulation might yield a mesh with a different topology, e.g., with two elements (red dashed lines) connecting the two fluid regions. In this case, it is unclear whether these two elements should be kept or discarded. Moreover, in both cases the error in mass conservation is of the same order as the size of the coarser mesh. $(\mathbf{c})$ Refining the coarser boundary and its neighborhood after discarding the connecting element (red dashed lines) delays the merging of the two fluid regions. $(\mathbf{d})$ Finally, delaying the coarsening of the newly refined boundary region ensures that the merging takes only place when the two boundaries are within a distance of the order of the minimum mesh size and, thereby, significantly reduces the mass conservation error.

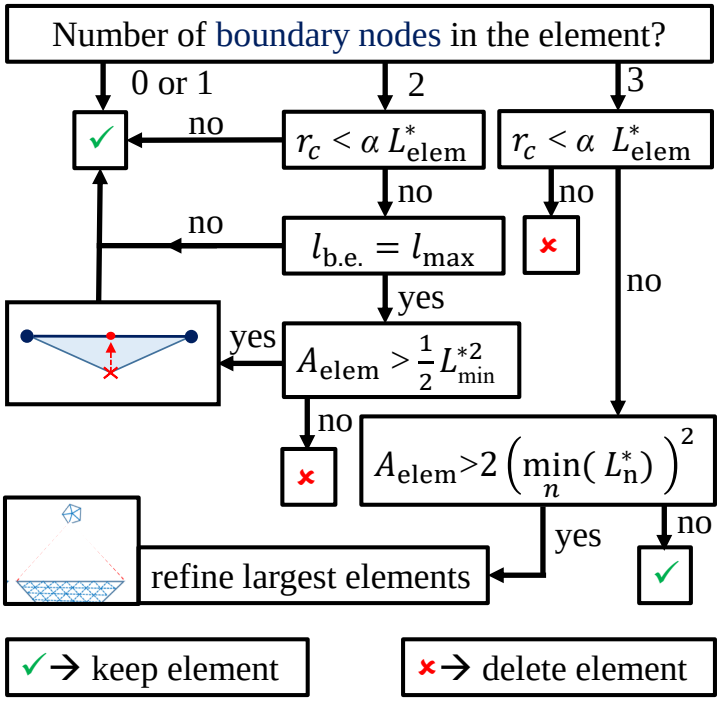


Figure 3.17: Algorithm for boundary recognition combining a local α-shape criterion and boundary tracking.

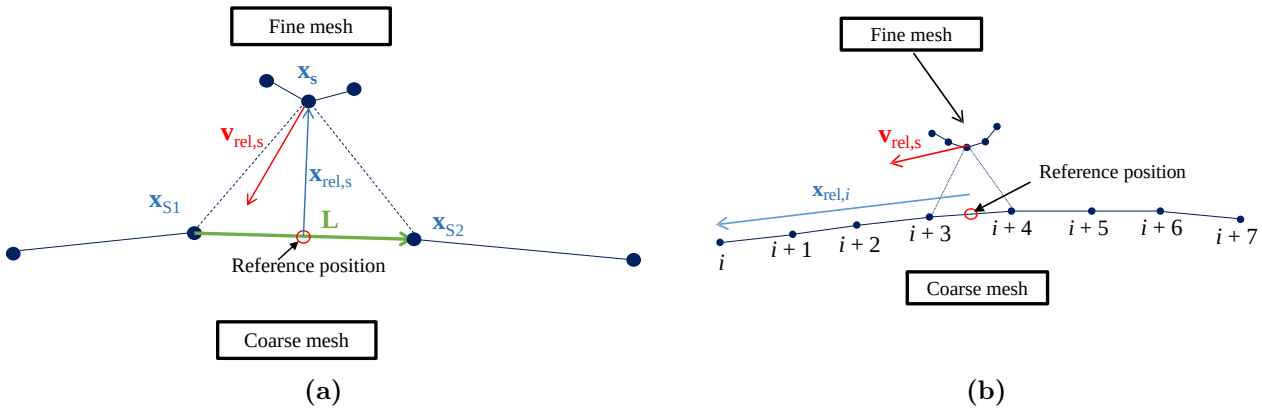


Figure 3.18: Definition of the key quantities for the estimation of the time until collision in the collision anticipation algorithm. Expected collision at (a) low and (b) high incidence between a surface with a fine mesh (top) and one with an initially coarser mesh (bottom).

3.3 Main guidelines for CFD simulations with the PFEM

In order to minimize the different sources of numerical errors, some guidelines for the use of the PFEM in the context of CFD simulations are given. Numerical errors mainly stem from the space and time discretization, and thus directly depend on the mesh, and in particular its resolution, and on the time step size. More importantly, the fact that the mesh is subjected to distortion because of nodal displacements has repercussions on both the choice of the spatial resolution and time step size, and is probably the most determinant aspect for these choices.

Choice of the mesh resolution

For CFD simulations, the resolution should be high in regions of high gradients such that the nodal values do not vary too much from one node to another. Intuitively, there should be enough linear finite elements in a region of fluid such that the exact local velocity and pressure fields are well captured by adjacent linear regions. In that regard, the critical part of a fluid flow consists in boundary layers next to walls, which are known to be thin for usual value of the viscosity, such that the velocity is rapidly varying from 0 (or the wall velocity for a moving body) to U (the value of the velocity outside the boundary layer, of the order of some characteristic velocity scale of the problem). Moreover, the velocity profile is not linear in that region, and the element resolution should therefore be smaller than the boundary layer thickness, such that several mesh elements are used to represent the boundary layer.

It is well known that for a laminar boundary layer at a distance x from the stagnation point, the boundary layer thickness $\delta(x)$ scales as

$$\delta(x) \sim \frac{x}{\sqrt{Re_x}} \quad \text{with} \quad Re_x = \frac{Ux}{\nu}\,, \tag{3.43}$$

ν being the fluid kinematic viscosity. For an estimation of a characteristic size of the boundary layer around a cylinder of radius a, the following scaling is simply obtained by replacing x by a in Eq. (3.43)[7]:

$$\frac{\delta}{a} = \frac{1}{\sqrt{Re_D/2}}. \tag{3.44}$$

In practice, the required resolution at the cylinder surface should be determined rigorously using a mesh convergence analysis, but the proposed scaling helps defining a range of element sizes on which this analysis should be performed. Moreover, several elements are required along the boundary layer thickness in order to get accurate results, such that the minimal mesh size should be smaller than the scaling of Eq. (3.44). As an initial guess, the following minimal mesh size at the cylinder surface is proposed:

$$\frac{L^*_{\min}}{a} = \frac{1}{4\sqrt{Re_D/2}}\,. \tag{3.45}$$

Based on this initial guess, a mesh convergence analysis for the case of a cylinder rising toward a free surface at constant velocity is given in Chapter 5.

Choice of the time step

Two main issues should be considered for the choice of the time step. First, the magnitude of the mesh distortion directly depends on the time step. Secondly, the latter has also an impact on the accuracy of the time integration.

Maximum time step $\Delta t_{\max}$

The advantage of a Lagrangian formalism over an Eulerian one is the absence of convective terms, well known to be responsible for a time step limitation, called the CFL (Courant–Friedrichs–Lewy) condition, for stable explicit time integration. The condition can be understood as a limit on the travel

[7] In the literature, the Reynolds number definition for a cylinder commonly involves its diameter D. Therefore Re_a is simply rewritten $Re_D/2$.

speed of information across the mesh, which cannot exceed the physical convective times, and can be written in a simplified way as

$$\Delta t < \min_{\mathbf{x}} \left(\frac{L(\mathbf{x})}{u(\mathbf{x})} \right) , \tag{3.46}$$

where $L(\mathbf{x})$ and $u(\mathbf{x})$ are respectively a measure of the cell size and magnitude of the velocity at location $\mathbf{x}$.

Within a Lagrangian formalism, such a limitation also exists because the mesh deformation occurring during a single time step should remain limited. On the one hand, the symmetric part of the velocity gradient tensor is a measure of the rate of element deformation (even for a perfect time integration). On the other hand, the anti-symmetric part of the velocity gradient tensor is a measure of the element rotation speed. Such rotation also results in some indirect elements distortion inherent to the finite accuracy of the time integration (first order backward Euler in the present case). Therefore, a characteristic deformation time can be defined as the inverse of the norm of the overall velocity gradient (deformation rate + rotation speed). The time step Δt should be lower than this upper bound in order to avoid too important mesh distortion during a time step:

$$\Delta t < \min_{\mathbf{x}} \left(\frac{1}{||\nabla \mathbf{u}||(\mathbf{x})} \right) , \tag{3.47}$$

where $\nabla \mathbf{u}$ is the velocity gradient tensor at location $\mathbf{x}$.

With no-slip boundary conditions, the gradients are typically the largest inside the boundary layers, in the region directly adjacent to the wall, as illustrated in Fig. 3.19. There, the norm of the velocity gradient matrix is approximately

$$||\nabla \mathbf{u}||_{\text{wall}} \simeq \left. \frac{\partial u}{\partial n} \right|_{\text{wall}} \tag{3.48}$$

where $\left. \frac{\partial u}{\partial n} \right|_{\text{wall}}$ is the wall-normal derivative of the tangential component of the velocity.

At that location, assuming the boundary layer to be well resolved, the velocity gradient can be approximated by

$$\left. \frac{\partial u}{\partial n} \right|_{\text{wall}} \approx \frac{u_c}{L^*_{\text{min}}} \tag{3.49}$$

where u_c is the tangential component of the velocity of the closest node to the wall and L^*_{min} is approximately its distance from it.

In practice, it is proposed to limit the time step according to

$$\Delta t < \frac{L^*_{\text{min}}}{U} < \frac{L^*_{\text{min}}}{u_c} , \tag{3.50}$$

where U is the velocity outside the boundary layer (as illustrated in Fig. 3.19). This provides a conservative (more stringent) criterion than

$$\Delta t < \frac{L^*_{\text{min}}}{u_c} , \tag{3.51}$$

and avoids computing u_c, which may depend on different a priori unknown quantities, such as boundary layer velocity profile and thickness. Moreover, for problems involving curved no-slip boundaries, U is chosen as a reference velocity of the problem, for instance the relative velocity of a moving body (cylinder/sphere) with respect to the surrounding fluid. Note that, because of the particular form of the norm of the velocity gradient near the walls, the condition on the time step (Eq. (3.50)) closely resembles that of the classical CFL condition used for Eulerian time integration (Eq. (3.46)).

Time discretization error

The second issue is related to the error coming from the time integration, inherent to the time discretization scheme. In that regard, the time step should be sufficiently small to capture rapidly varying features of the flow, which can be estimated based on physical considerations but should be verified

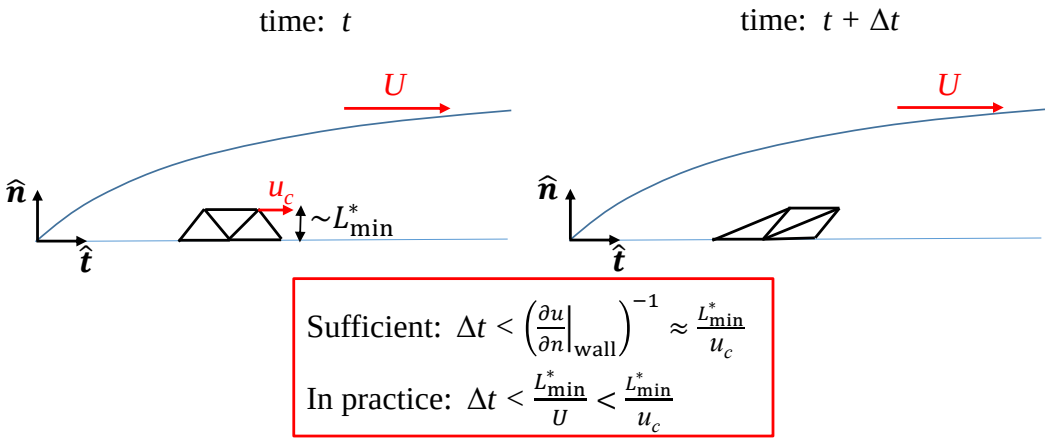


Figure 3.19: Illustration of the bound on the time step size resulting from the presence of a boundary layer on a flat wall. At the beginning of the time step (time t), the mesh elements are well shaped inside the boundary layer, assuming that a Delaunay triangulation has just been performed a the end of the previous time step. At the end of the time step (time $t + \Delta t$), the mesh elements inside the boundary layer are significantly deformed, especially those directly adjacent to the wall.

using time step convergence analysis. It is particularly crucial in the present case since the backward Euler scheme is only first order accurate in time, and is subjected to much numerical dissipation when larger time steps are used. In practice, however, it can be shown that the time step limit due to the mesh distortion, as introduced above, is a good starting point to perform a time step convergence analysis (see Chapter 5). Finally, it should also be mentioned that too small time steps may lead to spurious pressure oscillations arising from the remeshing procedure, as pointed out by Cerquaglia [65].

Mass conservation error

A particular case of time discretization error is that associated with mass conservation. Even if the present work mainly focuses on the reduction of the mass conservation error caused by the remeshing (i.e., stemming from the boundary recognition step of the remeshing), it is important to also consider the error in mass conservation linked to the numerical time integration.

For simplicity, only the backward Euler time integration of the position and continuity equations at time $n+1$ are here considered (a more complete analysis is given in Appendix C):

$$(\nabla \cdot \mathbf{u})^{n+1} = 0\,, \tag{3.52}$$

$$\frac{\mathbf{x}^{n+1} - \mathbf{x}^n}{\Delta t} = \mathbf{u}^{n+1}\,. \tag{3.53}$$

One can consider Eqs. (3.52-3.53) for a triangular element, whose velocity is taken in a non-rotating frame of reference attached to one of the nodes, as illustrated in Fig. 3.20. The nodal positions and

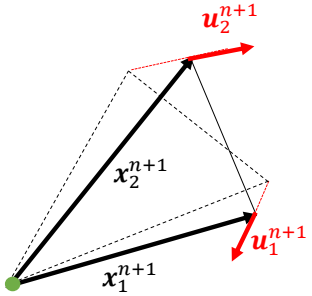


Figure 3.20: Deformation of a triangular element during a single time step of a backward Euler time integration. The two red vectors are the velocities of two nodes in the (non-rotating) system of reference fixed with the third node (green). The dashed lines represent the triangle before deformation, at time $t_n = -\Delta t$.

velocities at time t_{n+1} are thus:

$$\mathbf{x}_0^{n+1} = (0,0)\,, \qquad \mathbf{u}_0^{n+1} = (0,0)\,, \tag{3.54}$$

$$\mathbf{x}_1^{n+1} = (x_1^{n+1}, y_1^{n+1})\,, \quad \mathbf{u}_1^{n+1} = (u_1^{n+1}, v_1^{n+1})\,, \tag{3.55}$$

$$\mathbf{x}_2^{n+1} = (x_2^{n+1}, y_2^{n+1})\,, \quad \mathbf{u}_2^{n+1} = (u_2^{n+1}, v_2^{n+1})\,, \tag{3.56}$$

For this linear finite element, the two derivatives $\left.\frac{\partial u}{\partial x}\right|^{n+1}$ and $\left.\frac{\partial v}{\partial y}\right|^{n+1}$ can be computed as

$$\left.\frac{\partial u}{\partial x}\right|^{n+1} = \frac{u_1^{n+1}\, y_2^{n+1} - u_2^{n+1}\, y_1^{n+1}}{x_1^{n+1}\, y_2^{n+1} - x_2^{n+1}\, y_1^{n+1}}\,, \tag{3.57}$$

$$\left.\frac{\partial v}{\partial y}\right|^{n+1} = \frac{v_2^{n+1}\, x_1^{n+1} - v_1^{n+1}\, x_2^{n+1}}{x_1^{n+1}\, y_2^{n+1} - x_2^{n+1}\, y_1^{n+1}}\,, \tag{3.58}$$

which gives, for the divergence-free equation:

$$(\nabla\cdot\mathbf{u})^{n+1} = 0 \Longleftrightarrow \left.\frac{\partial u}{\partial x}\right|^{n+1} + \left.\frac{\partial v}{\partial y}\right|^{n+1} = 0 \Longleftrightarrow u_1^{n+1}\, y_2^{n+1} + x_1^{n+1}\, v_2^{n+1} - u_2^{n+1}\, y_1^{n+1} - x_2^{n+1}\, v_1^{n+1} = 0\,. \tag{3.59}$$

Now, taking the origin of time $t = t_{n+1} = 0$, the times between t_n and t_{n+1} correspond to negative values of t, and the relative positions of nodes 1 and 2 with respect to node 0 are, according to Eq. (3.53), given by

$$\mathbf{x}_1(t) = (x_1^{n+1} + u_1^{n+1}\, t, y_1^{n+1} + v_1^{n+1}\, t) \quad \text{for} \quad -\Delta t < t < 0\,,$$
$$\mathbf{x}_2(t) = (x_2^{n+1} + u_2^{n+1}\, t, y_2^{n+1} + v_2^{n+1}\, t) \quad \text{for} \quad -\Delta t < t < 0\,.$$

The element area at a time $t \in [-\Delta t, 0]$ is thus given by (assuming the non-zero component of $\mathbf{x}_1(t) \times \mathbf{x}_2(t)$ is positive)

$$A(t) = \frac{1}{2}||\mathbf{x}_1(t) \times \mathbf{x}_2(t)|| = \frac{1}{2}\left[(x_1^{n+1} + u_1^{n+1}\, t)(y_2^{n+1} + v_2^{n+1}\, t) - (x_2^{n+1} + u_2^{n+1}\, t)(y_1^{n+1} + v_1^{n+1}\, t)\right]\,. \tag{3.60}$$

Taking the derivative of that expression with respect to time leads to

$$\frac{\partial A}{\partial t}(t) = \frac{1}{2}[u_1^{n+1}\,(y_2^{n+1} + v_2^{n+1}\, t) + v_2^{n+1}\,(x_1^{n+1} + u_1^{n+1}\, t) - u_2^{n+1}\,(y_1^{n+1} + v_1^{n+1}\, t) - v_1^{n+1}\,(x_2^{n+1} + u_2^{n+1}\, t)]\,. \tag{3.61}$$

Then imposing this to be 0 at $t = 0$ (i.e., at time step $n+1$) yields

$$\left.\frac{\partial A}{\partial t}\right|_{t=0} = 0 \Longleftrightarrow u_1^{n+1}\, y_2^{n+1} + x_1^{n+1}\, v_2^{n+1} - u_2^{n+1}\, y_1^{n+1} - x_2^{n+1}\, v_1^{n+1} = 0\,, \tag{3.62}$$

which is the same expression as the divergence-free condition given by Eq. (3.59). As a consequence, the area of the element does not vary much at the end of a time step (close to $t = 0$) but varies strongly at the beginning of the time step (i.e., for t close to $-\Delta t$). This can be observed more precisely by injecting the result of Eq. (3.59) into Eq. (3.61),

$$\frac{\partial A}{\partial t}(t) = (u_1^{n+1}\, v_2^{n+1} - u_2^{n+1}\, v_1^{n+1})\, t\,, \tag{3.63}$$

which only vanishes when the relative velocities $\mathbf{u}_1^{n+1}$ and $\mathbf{u}_2^{n+1}$ are parallel to each other. The total error on the area between time $t = -\Delta t$ and $t = 0$ is finally given by

$$\Delta A = \int_{-\Delta t}^{0} \left(u_1^{n+1}\, v_2^{n+1} - u_2^{n+1}\, v_1^{n+1}\right)\, t\, dt = -\left(u_1^{n+1}\, v_2^{n+1} - u_2^{n+1}\, v_1^{n+1}\right)\frac{\Delta t^2}{2} \tag{3.64}$$

and the scheme is therefore second order accurate in terms of mass conservation.[8]

[8]Despite the second order accuracy, and considering the time step to scale as $L^*_{\min}/U$ and the area of the smallest element as $L^{*2}_{\min}$, the relative area variation $\Delta A/A$ near the no-slip boundary conditions may be consequent because of large mesh distortion (i.e., the term $L^*_{\min}$ vanishes in the local expression of $\Delta A/A$). However, at that location the velocity is almost parallel to the wall, and therefore, the two velocities u_1 and u_2 are also almost parallel to each other. This means that the factor in front of $\Delta t^2/2$ in Eq.3.64 is small.

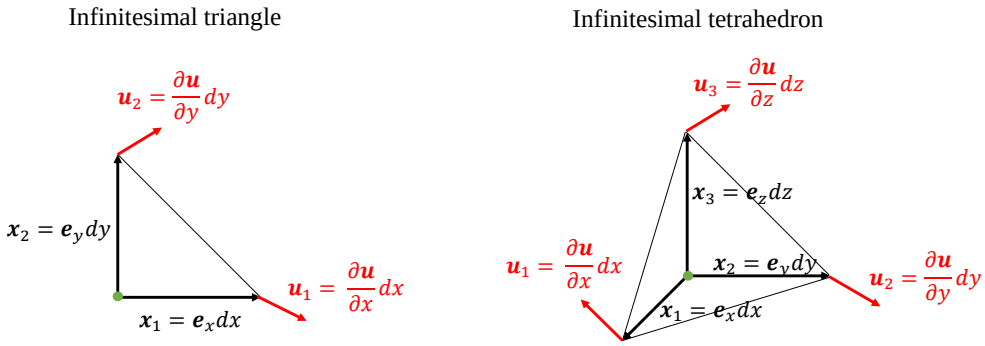


Figure 3.21: Particular case of an infinitesimal triangle (left) and tetrahedron (right) rectangle at the reference node (green). At the infinitesimal level, velocities (red vectors) of the other nodes can be expressed with the velocity gradients. The relative position of these nodes is respectively expressed by infinitesimal multiples of the orthogonal axes in the global Cartesian frame of reference.

Furthermore, this is further compounded by the stabilization term added to the incompressibility constraint that also impacts the mass conservation, lowering the accuracy to first order, as briefly explained in Appendix C.

Link with velocity gradient tensor

The above demonstration considered the specific case of linear triangular elements, i.e., a specific space discretization. But even with a perfect space discretization, the backward Euler scheme is subjected to a mass conservation error. This error is associated with the residual divergence of the momentum equation (as shown in Appendix C) and with the invariants of the velocity gradient tensor. To show this latter relation, we can consider a particular infinitesimal triangle (tetrahedron in 3D) which is rectangle at the node $\mathbf{x}_0$, with each adjacent edge parallel to one of the frame axes, as illustrated in Fig. 3.21.[9] The positions and velocities relative to this node $\mathbf{x}_0$ are

$$\begin{aligned}
&\mathbf{x}_0^{n+1} = (0,0)\,, && \mathbf{u}_0^{n+1} = (0,0)\,,\\
&\mathbf{x}_1^{n+1} = (dx,0) = \mathbf{e}_x dx\,, && \mathbf{u}_1^{n+1} = \left(\left.\frac{\partial u}{\partial x}\right|^{n+1}, \left.\frac{\partial v}{\partial x}\right|^{n+1} \right) dx\,,\\
&\mathbf{x}_2^{n+1} = (0,dy) = \mathbf{e}_y dy\,, && \mathbf{u}_2^{n+1} = \left(\left.\frac{\partial u}{\partial y}\right|^{n+1}, \left.\frac{\partial v}{\partial y}\right|^{n+1} \right) dy\,,
\end{aligned}$$

for a triangle, u,v being the respective velocity components along the directions x-y, and

$$\begin{aligned}
&\mathbf{x}_0^{n+1} = (0,0,0)\,, && \mathbf{u}_0^{n+1} = (0,0,0)\,,\\
&\mathbf{x}_1^{n+1} = (dx,0,0) = \mathbf{e}_x dx\,, && \mathbf{u}_1^{n+1} = \left.\frac{\partial \mathbf{u}}{\partial x}\right|^{n+1} dx = \left(\left.\frac{\partial u}{\partial x}\right|^{n+1}, \left.\frac{\partial v}{\partial x}\right|^{n+1}, \left.\frac{\partial w}{\partial x}\right|^{n+1} \right) dx\,,\\
&\mathbf{x}_2^{n+1} = (0,dy,0) = \mathbf{e}_y dy\,, && \mathbf{u}_2^{n+1} = \left.\frac{\partial \mathbf{u}}{\partial y}\right|^{n+1} dy = \left(\left.\frac{\partial u}{\partial y}\right|^{n+1}, \left.\frac{\partial v}{\partial y}\right|^{n+1}, \left.\frac{\partial w}{\partial y}\right|^{n+1} \right) dy\,,\\
&\mathbf{x}_2^{n+1} = (0,0,dz) = \mathbf{e}_z dz\,, && \mathbf{u}_3^{n+1} = \left.\frac{\partial \mathbf{u}}{\partial z}\right|^{n+1} dz = \left(\left.\frac{\partial u}{\partial z}\right|^{n+1}, \left.\frac{\partial v}{\partial z}\right|^{n+1}, \left.\frac{\partial w}{\partial z}\right|^{n+1} \right) dz\,,
\end{aligned}$$

for a tetrahedron, w being the third velocity component along the third direction z. The area $A(t)$ of this triangle and the volume $V(t)$ of this tetrahedron, for time $t|t_n = -\Delta t < t < t_{n+1} = 0$, are given

[9] This simplifies the derivation of the mass conservation error, although it could have been obtained from an arbitrary infinitesimal triangle/tetrahedron.

by the following expressions

$$\begin{aligned} A(t) &= \frac{1}{2}\det\left(\mathbf{e}_x dx + t\left.\frac{\partial \mathbf{u}}{\partial x}\right|^{n+1} dx\,,\, \mathbf{e}_y dy + t\left.\frac{\partial \mathbf{u}}{\partial y}\right|^{n+1} dy\right)\,, \\ &= \frac{dx\,dy\,t^2}{2}\det\left(-\frac{1}{t}\mathbf{I} - \nabla\mathbf{u}^{n+1}\right)\,, \end{aligned} \tag{3.65}$$

$$\begin{aligned} V(t) &= \frac{1}{6}\det\left(\mathbf{e}_x dx + t\left.\frac{\partial \mathbf{u}}{\partial x}\right|^{n+1} dx\,,\, \mathbf{e}_y dy + t\left.\frac{\partial \mathbf{u}}{\partial y}\right|^{n+1} dy\,,\, \mathbf{e}_z dz + t\left.\frac{\partial \mathbf{u}}{\partial z}\right|^{n+1} dz\right)\,, \\ &= -\frac{dx\,dy\,dz\,t^3}{6}\det\left(-\frac{1}{t}\mathbf{I} - \nabla\mathbf{u}^{n+1}\right)\,, \end{aligned} \tag{3.66}$$

where det is the determinant. One can recognize the characteristic determinant of the opposite of the velocity gradient tensor, $-\nabla\mathbf{u}^{n+1}$, on the right-hand side of both equations (As a reminder, the characteristic polynomial of a matrix $\mathbf{A}$ has the form $det(\mathbf{A} - \lambda\mathbf{I})$. Using the invariants of $-\nabla\mathbf{u}^{n+1}$, this can be rewritten in terms of the characteristic polynomial,

$$\begin{aligned} A(t) &= \frac{dx\,dy}{2}\left(1 + \mathrm{tr}\left(\nabla\mathbf{u}^{n+1}\right)t + \det\left(\nabla\mathbf{u}^{n+1}\right)t^2\right) \\ &= \frac{dx\,dy}{2}\left(1 + \det\left(\nabla\mathbf{u}^{n+1}\right)t^2\right)\,, \end{aligned} \tag{3.67}$$

$$\begin{aligned} V(t) &= -\frac{dx\,dy\,dz}{6}\left(-1 - \mathrm{tr}\left(\nabla\mathbf{u}^{n+1}\right)t - \frac{1}{2}\left(\mathrm{tr}\left(\nabla\mathbf{u}^{n+1}\right)^2 - \mathrm{Tr}\left(\left(\nabla\mathbf{u}^{n+1}\right)^2\right)\right)t^2 - \det\left(\nabla\mathbf{u}^{n+1}\right)t^3\right) \\ &= \frac{dx\,dy\,dz}{6}\left(1 + \frac{1}{2}\mathrm{tr}\left(\left(\nabla\mathbf{u}^{n+1}\right)^2\right)t^2 + \det\left(\nabla\mathbf{u}^{n+1}\right)t^3\right)\,, \end{aligned} \tag{3.68}$$

where the trace tr of the velocity gradient tensor has vanished because of the continuity equation: $\mathrm{tr}\left(\nabla\mathbf{u}^{n+1}\right) = \nabla\cdot\mathbf{u}^{n+1} = 0$. Replacing t by $-\Delta t$ in these expressions clearly shows that, even for a perfect space discretization, the accuracy on the mass conservation is second order accurate in time, and the error is larger where the velocity gradients are large.

One can conclude that the local numerical mass conservation error is reduced through the use of a smaller time step, such that the elements do not deform too much during a single time step. Finally, it should be emphasized that the local numerical mass conservation error has a direct implication for the global numerical mass conservation. If deformable boundaries are involved in a simulation (such as free surfaces for instance), the exact compensation of the local surface area increases and decreases across all elements cannot be guaranteed, such that total mass is not conserved.

Finally, it should be mentioned that the above derivation is linked with the geometric conservation law (GCL) of the Arbitrary Lagrangian-Eulerian approach [66, 67]. When applying ALE to convection problems, the error on the variation of volume of mesh cells directly impacts the computation of the fluxes and the order of accuracy of the numerical method. In this situation, satisfying the GCL equation helps to get stable and accurate solution. It is given by

$$\frac{d}{dt}\int_{K(t)} dx = \int_{\partial K(t)} \boldsymbol{\kappa}(s,t)\cdot\mathbf{n}(s,t)ds\,, \tag{3.69}$$

where $\boldsymbol{\kappa}$ is the velocity of the volume $K(t)$, and $\mathbf{n}$ the normal on its boundary $\partial K(t)$. In particular, satisfying this equation guarantees to recover the exact solution of a uniform stream flow. Within the present Lagrangian approach, the flow velocity $\mathbf{u}$ and the volume velocity κ are the same thing, such that the GCL equation is automatically satisfied. Moreover, it implies that in case of an incompressible fluid flows, mass conservation errors can be directly obtained from purely geometric considerations, as just shown above.

3.4 A theoretical framework for the mesh adaptation algorithm in 3D

The purpose of this section is to discuss the extension of the mesh adaptation algorithm to 3D. First, the concept of nodal density is formalized. To demonstrate the relevance of the approach, some

useful identities are derived within this framework. Afterwards, the 3D nodal density and its associated quantities are rigorously defined and theoretical algorithms are presented for 3D boundary recognition, mesh refinement, and mesh coarsening. Eventually, the case of slivers, i.e., highly flatten tetrahedra, are discussed, and it is shown how the concept of nodal density can intuitively explain why such elements are present in 3D meshes.

3.4.1 Definition of the nodal fraction and the nodal density

In this section, the concept of nodal density is defined in two dimensions. In addition, it is shown that it is a useful measure that can be used to relate the number of nodes of a mesh to the number of triangular cells and boundary edges. A generalization of this concept is then given for the 3D case in Section 3.4.2, and analogous relations are derived there.

The concept of nodal density is ambiguous as a cloud of nodes is by definition discrete while a density is usually a concept dealing with continuous quantities. Intuitively, the nodal density simply refers to the number of nodes per unit volume/area, and basically measures the density and homogeneity of nodes in space.

The drawback of this definition is that, because it is understood in an average sense, a sufficient number of points, and thus a sufficiently large surface/volume, is required to obtain a meaningful estimate of the nodal density. This is however at the cost of locality. This is illustrated for a 2D uniform mesh of equilateral triangles as shown in Fig. 3.22. The nodal density is obtained by dividing the number of nodes inside the circle of radius r by its corresponding area πr^2. The resulting nodal density is shown as a function of the circle size in Fig. 3.23. It can be seen that for a small r, the nodal density depends strongly on the circle size (and in general on the reference surface/volume considered). When the circle is increased, new nodes are included leading to jumps in the nodal density. Only when the circle becomes sufficiently large, the nodal density starts to converge to a constant value. As explained in the next paragraph, this limit value corresponds to $1/(2A_\Delta)$ where A_Δ is the surface area of one of the equilateral triangles, which means that there is a nodal fraction $f_N = 1/2$ inside a triangle.

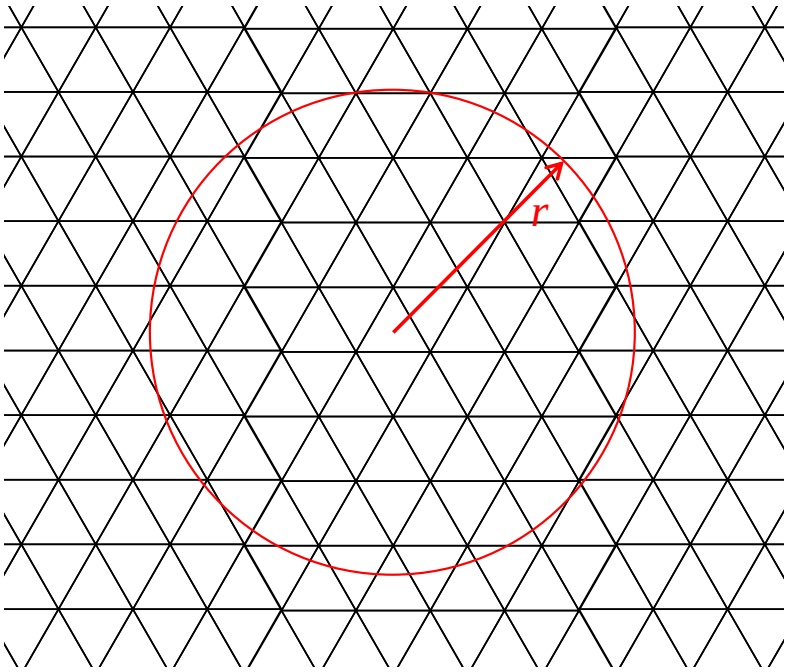


Figure 3.22: Illustration of the concept of nodal density, which is computed by dividing the number of nodes inside the circle of radius r by its area πr^2. The larger the radius is, the more accurate the nodal density measure is, as illustrated in Fig. 3.23, but the less local it is.

This can obviously be generalized for any polygon, where the nodal density of a mesh element e is by definition

$$\sigma^e_{2\mathrm{D}} \triangleq \frac{f^e_N}{A^e}\,, \tag{3.70}$$

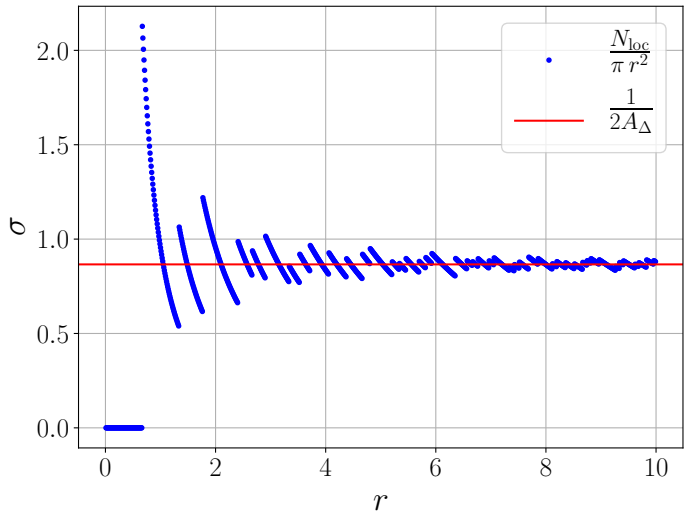


Figure 3.23: Variation of the nodal density of a uniform mesh made of equilateral triangles, as one varies the circular control volume of radius r, as illustrated in Fig. 3.22. The red line corresponds to the theoretical value, which is obtained by dividing the number of nodes in one triangle (1/2) by its area.

f_N^e and A^e being respectively the nodal fraction and the area of the element. Specifically, for a node i shared by several elements, a fraction $\phi_i^e/(2\pi)$ of that node can be assigned to a specific element, where ϕ_i^e is the internal angle of this element at that node i (see Fig. 3.24a for the case of a triangular element). Considering the N_{nodes}^e nodes of this element, the total nodal fraction f_N inside this element e is

$$f_N^e = \frac{1}{2\pi} \sum_{i=1}^{N_{\text{nodes}}^e} \phi_i^e \,. \tag{3.71}$$

Because the angles of a polygon satisfy

$$\sum_{i=1}^{N_{\text{nodes}}^e} \phi_i^e = \pi(N_{\text{edges}}^e - 2)\,, \tag{3.72}$$

where N_{edges}^e is the number of edges of element e, Eq. (3.71) can be rewritten

$$f_N^e = \frac{(N_{\text{edges}}^e - 2)}{2}\,, \tag{3.73}$$

with $f_N^e = f_{N,\Delta} = 1/2$ for triangular elements. Finally, the general nodal density of an element e can be expressed as

$$\sigma_{\text{2D}}^e = \frac{(N_{\text{edges}}^e - 2)}{A^e}\,. \tag{3.74}$$

In particular, this gives, for an element e of a triangular mesh:

$$\sigma_\Delta^e = \frac{1}{2A_\Delta^e}\,. \tag{3.75}$$

The advantage of this definition is that it provides a local and accurate estimate of the nodal density, which is furthermore easy to compute. It is also consistent with the limit value obtained in the above example.

The same approach can be considered for boundaries. As shown in Fig. 3.24b, half a node can be assigned to a boundary edge on a straight boundary. One can thus define a linear nodal density for a boundary edge b of length l^b:

$$\sigma^b = \frac{1}{2l^b}\,. \tag{3.76}$$

If the boundary is convex (concave), a small fraction of each boundary node is not assigned to any edge (double counted), as shown in Fig. 3.25. For the external boundary of a meshed region, these angle fractions sum up to 2π, i.e., one additional node that is not accounted for. For a possible hole in the meshed region, these angle fractions sum up to -2π, i.e., one node doubled counted in this case.

Consequently, for any 2D mesh, possibly divided into N_{dom} domains, including N_{holes} unmeshed holes, the total number of nodes is

$$N_{\text{nodes}} = \sum_{e \in \mathcal{E}} A^e \sigma^e + \sum_{b \in \mathcal{B}} l^b \, \sigma^b + N_{\text{dom}} - N_{\text{holes}} \, . \tag{3.77}$$

where $\mathcal{E}$ and $\mathcal{B}$ are respectively the ensembles of elements and boundary edges of the mesh. For a mesh containing only one type of polygonal faces, it gives

$$N_{\text{nodes}} = f_N^e \, N_{\text{elems}} + \frac{1}{2} N_{\text{b.edges}} + N_{\text{dom}} - N_{\text{holes}} \, , \tag{3.78}$$

with $f_N^e = f_{N,\Delta} = 1/2$ for the particular case of a triangular mesh, and where N_{elems} and $N_{\text{b.edges}}$ are respectively the number of faces and the number of boundary edges of the considered mesh.

Finally, a relation that also relates the number of interior edges with the previous quantities can also be found. The Euler formula [68] for two-dimensional graphs[10] is

$$N_{\text{nodes,g}} - N_{\text{edges,g}} + N_{\text{faces,g}} = 2 \, , \tag{3.79}$$

where $N_{\text{nodes,g}}$, $N_{\text{edges,g}}$ and $N_{\text{faces,g}}$ are respectively the number of nodes, edges and faces of the graph. The number of nodes of the graph, $N_{\text{nodes,g}}$, corresponds to the number of nodes of the mesh, N_{nodes}. However, as the formula is valid for planar graphs in 1 piece (multiples pieces are not taken into account), $N_{\text{dom}} - 1$ fictitious edges can be added to minimally reconnect the N_{dom} distinct graphs together into a single graph, as illustrated in Fig. 3.26, such that $N_{\text{edges,g}}$ should be replaced by $N_{\text{edges}} + N_{\text{dom}} - 1$. Morover, the number of faces can be decomposed according to

$$N_{\text{faces,g}} = N_{\text{elems}} + N_{\text{holes}} + 1 \, , \tag{3.80}$$

N_{elems} corresponding to the number of mesh elements, N_{holes} to the number of holes, and the additional term 1 is here to account for the remaining unbounded plane, as illustrated in Fig. 3.26. With this, the Euler formula reads

$$N_{\text{nodes}} - N_{\text{edges}} + N_{\text{elems}} = N_{\text{dom}} - N_{\text{holes}} \, . \tag{3.81}$$

Furthermore, N_{edges} can be decomposed into its interior and boundary contributions:

$$N_{\text{edges}} = N_{\text{b.edges}} + N_{\text{i.edges}} \, . \tag{3.82}$$

Replacing this expression and Eq. (3.78) for N_{nodes} ($N_{\text{nodes},\Delta}$) in the Euler relation (Eq. (3.81)) yields for a triangular mesh

$$N_{\text{i.edges}} = \frac{3}{2} \, N_{\text{elems},\Delta} - \frac{1}{2} N_{\text{b.edges}} \, . \tag{3.83}$$

The use of Eqs. (3.78) and (3.83) is illustrated in Fig. 3.27.

Equation (3.83) could have been obtained in a simpler way by recognizing, because an edge is shared by its two adjacent triangles, that each triangle contains $\frac{3}{2}$ edges. Furthermore, boundary edges have only one triangle adjacent to them so that half of the boundary edges should be added. Therefore, one has

$$N_{\text{edges}} = \frac{3}{2} \, N_{\text{elems},\Delta} + \frac{1}{2} N_{\text{b.edges}} \, , \tag{3.84}$$

and subtracting $N_{\text{b.edges}}$ from both sides directly gives Eq. (3.83). Equation (3.78) could have thus been derived from Euler formula (Eq. (3.79)).

It should be emphasized that relations like Eq. (3.84) can be useful, in the context of meshes, to evaluate the complexity of the remeshing algorithm (which has not been done in the context of this

[10] The formula is the same for the entities (vertices, edges and faces) at the bounding surface of a convex polyhedron.

thesis but could be considered for future work). In particular, algorithms doing more or fewer operations depending on the type of geometry entities considered (faces, edges or nodes[11]), whether they are located near boundaries or not for instance, have a complexity that depends on the number of geometry entities of each type. In this case, the overall complexity of the algorithm has a more complex expression than a simple relation involving the total number of nodes N_{nodes}.

Particular case of a large mesh

If the mesh is sufficiently large and the length of the boundary sufficiently short, the number of boundary edges may be neglected in Eq. (3.78), in comparison to the total number of nodes. It leads to a direct approximate relation between the number of mesh nodes and the number of elements:

$$N_{\text{nodes}} \simeq f_N^e \, N_{\text{elems}} \tag{3.85}$$

For instance, for triangular meshes, there are approximately twice more elements than nodes. It is interesting because, despite the two degrees of freedom on the number of mesh entities (i.e., number of nodes, edges and faces), those being constrained by Euler relation (Eq. (3.79)), particularizing the latter to sufficiently large meshes (i.e., whose number of entities on the boundary can be neglected) enables to get almost one-to-one relations between two entities. For triangular meshes, one has for instance

$$\text{from Eq. (3.85)} \quad : \; N_{\text{nodes}} \simeq \frac{1}{2} \, N_{\text{elems},\Delta} \, , \tag{3.86}$$

$$\text{from Eq. (3.84)} \quad : \; N_{\text{edges}} \simeq \frac{3}{2} \, N_{\text{elems},\Delta} \, , \tag{3.87}$$

$$\text{from Eqs. above} \; : \; N_{\text{edges}} \simeq 3 \, N_{\text{nodes}} \, . \tag{3.88}$$

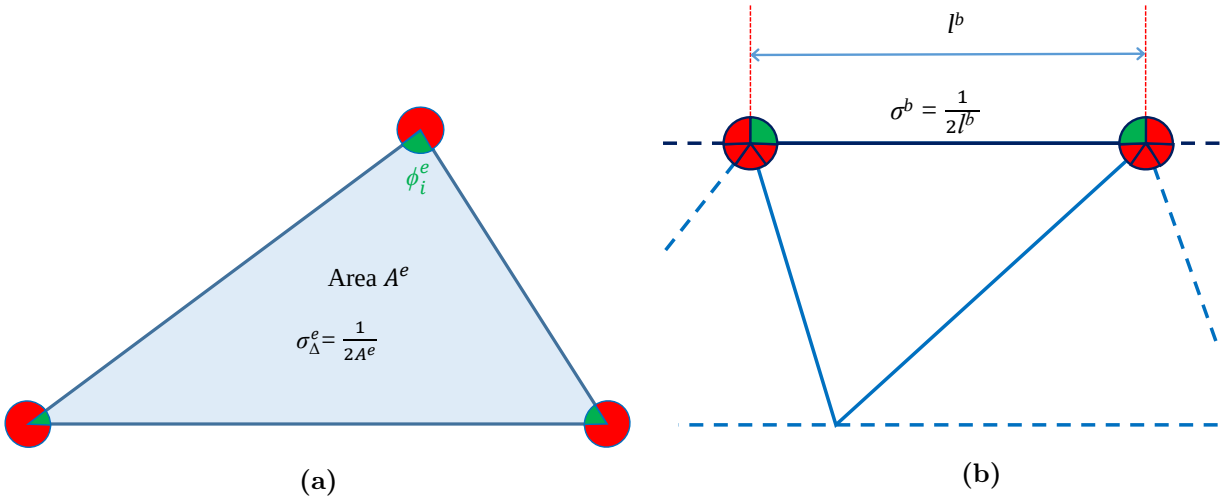


Figure 3.24: Definition of nodal density. Fractions of nodes assigned to the triangle (**a**) or to the boundary edge (**b**), are depicted by the green disk fragments, showing that there is a total of half a node in each.

[11] including cells for 3D meshes.

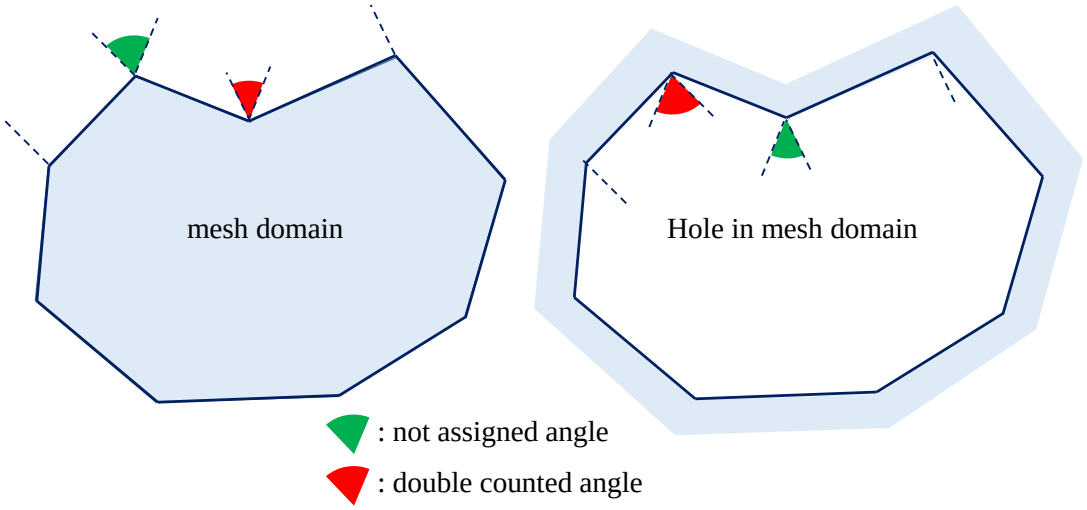


Figure 3.25: Illustration of the difference, in terms of the boundary angle count, between the exterior (left) and interior boundary (right) of a mesh. The sum of all the "unassigned" angles at convex locations (green) minus the sum of the "double counted" angles at concave locations (red) equals 2π for an external boundary (left) and equals -2π for an internal boundary (right).

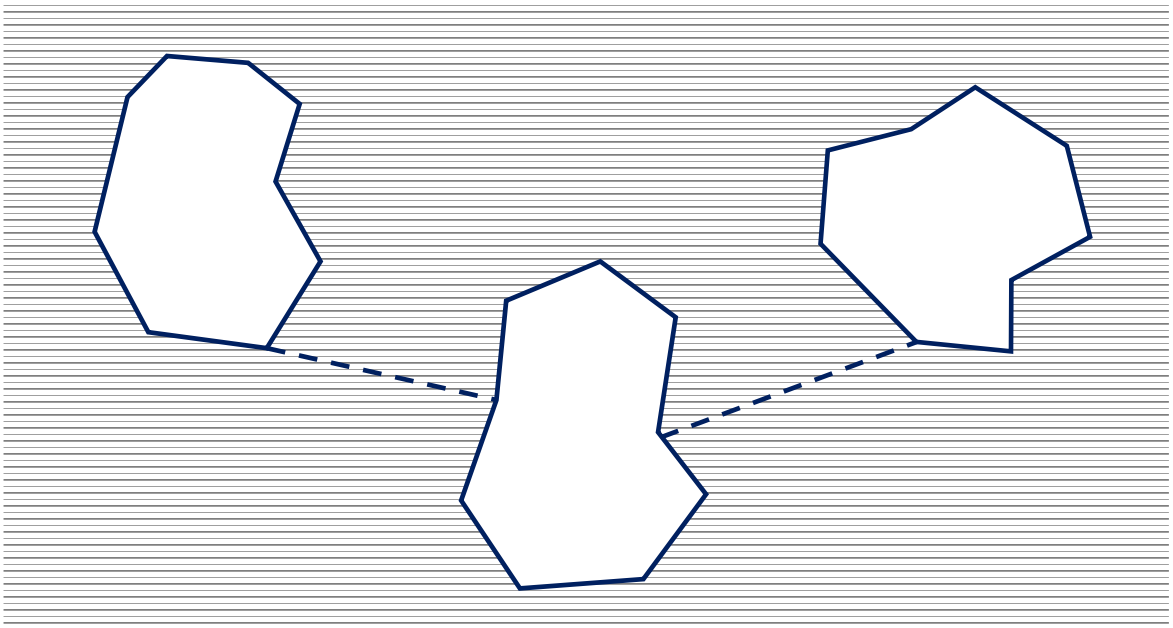


Figure 3.26: Illustration of a valid graph (i.e., for which the Euler formula given by Eq. (3.79) can be used) built by connecting $N_{\text{dom}} = 3$ distinct mesh domains (of which only the exterior boundary is represented by dark blue lines) using $N_{\text{dom}} - 1 = 2$ fictitious edges (dashed dark blue lines). Finally, the surrounding infinite part of the plane (hatched with blue lines), excluding the mesh domains, count as one face of the graph.

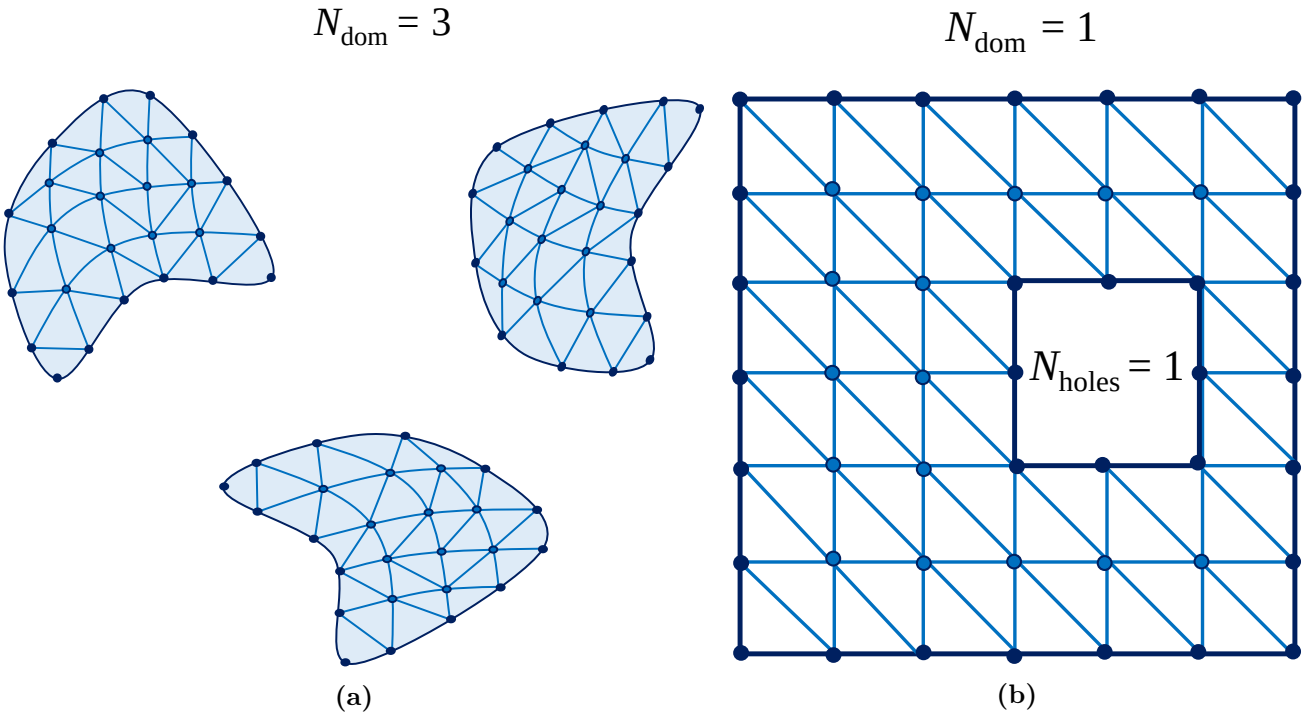


Figure 3.27: Illustration of the relations (3.78) and (3.83). (**a**) Schematic of three triangular meshed regions: $N_{\text{nodes}} = 78$, $N_{\text{edges}} = 180$, $N_{\text{b.edges}} = 45$, $N_{\text{i.edges}} = 135$, $N_{\text{elem},\Delta} = 105$, $N_{\text{holes}} = 0$ and $N_{\text{dom}} = 3$ (the relations are valid for each of the 3 meshes taken individually, dividing each quantities by 3). (**b**) Mesh with a hole: $N_{\text{nodes}} = 48$, $N_{\text{edges}} = 112$, $N_{\text{b.edges}} = 32$, $N_{\text{i.edges}} = 32$, $N_{\text{elem},\Delta} = 64$, $N_{\text{holes}} = 1$ and $N_{\text{dom}} = 1$.

3.4.2 Three dimensional nodal fractions and nodal densities

The generalization of the concept of nodal density to three dimensions consists in interpreting the ratio of solid angles to 4π as nodal fractions. Specifically, for a node i shared by several elements, a fraction $\Omega_i^e/(4\pi)$ of that node can be assigned to a specific element e, where Ω_i^e is the internal angle of this element e at that node i (see Fig. 3.28). Considering the other nodes of this element, the total nodal fraction f_N^e inside this element is

$$f_N^e = \frac{1}{4\pi} \sum_{i=1}^{N_{\text{nodes}}^e} \Omega_i^e \,, \tag{3.89}$$

which is analogous to Eq. (3.71) for the two dimensional case. Like in the two-dimensional case, Eq. (3.71) can be particularized for a given polyhedron e using a relation between the internal angles

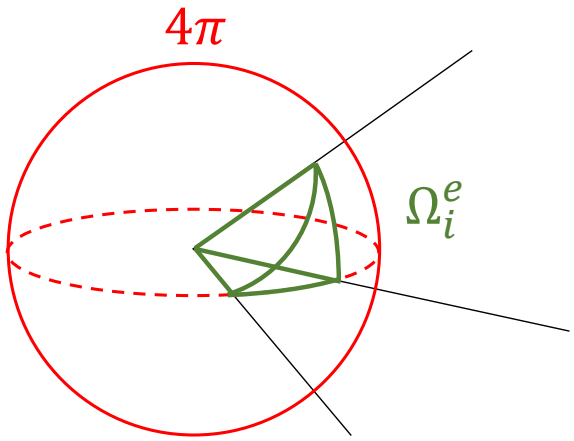


Figure 3.28: Illustration of the nodal fraction at a polyhedron vertex (minimally involving 3 edges here). The solid angle at the vertex (green) is a fraction of 4π (red), corresponding to a fraction of a node at the vertex.

and the number N^e_{facets} of polyhedron facets.[12] The relation is known as the Gram-Euler theorem [69] and writes as

$$2\sum_{i=1}^{N^e_{\text{edges}}}\phi^e_i-\sum_{i=1}^{N^e_{\text{nodes}}}\Omega^e_i=2\pi(N^e_{\text{facets}}-2)\,. \tag{3.90}$$

where ϕ_i is the internal dihedral angles at edge i, Ω_i the internal solid angles at nodes i, and N^e_{edges} and N^e_{nodes} are respectively the number of edges and nodes of the polyhedron. Note that a generalization of this result to higher dimensions has been later discovered [70, 71].

By opposition to the relation in two dimensions, Eq. (3.90) involves 2 types of angles, such that the solid angle sum is not constant but depends on the sum of the dihedral angles. As a conclusion, the 3D nodal fraction given by Eq. (3.89), unlike in the 2D case, depend on the shape of the element(s). It is given by

$$\begin{aligned} f^e_{N,3D}&=\frac{1}{4\pi}\sum_{i=1}^{N^e_{\text{nodes}}}\Omega^e_i\\ &=\frac{1}{4\pi}\left(2\sum_{i=1}^{N^e_{\text{edges}}}\phi^e_i-2\pi N^e_{\text{facets}}+4\pi\right)\\ &=\frac{1}{2\pi}\left(\sum_{i=1}^{N^e_{\text{edges}}}\phi^e_i\right)-\frac{N^e_{\text{facets}}}{2}+1\,. \end{aligned} \tag{3.91}$$

Corollary of the Gram-Euler formula

There is a nice corollary to the Gram-Euler theorem, Eq. (3.90), giving a similar relation between external polyhedron angles and the number of facets. It is first shown that, while the interior sum of the dihedral angles and the solid angles does not directly reflect the number of edges and nodes, it is the case for the total sum. Specifically, one has the following relations,

$$\sum_{\text{s.angles}}\Omega^e=\sum_{i=1}^{N^e_{\text{nodes}}}\Omega^e_i+\sum_{i=1}^{N^e_{\text{nodes}}}\Omega^{e\prime}_i=\sum_{i=1}^{N^e_{\text{nodes}}}(\Omega^e_i+\Omega^{e\prime}_i)=4\pi N_{\text{nodes}}\,, \tag{3.92}$$

$$\sum_{\text{d.angles}}\phi^e=\sum_{i=1}^{N^e_{\text{edges}}}\phi^e_i+\sum_{i=1}^{N^e_{\text{edges}}}\phi^{e\prime}_i=\sum_{i=1}^{N^e_{\text{edges}}}(\phi_i+\phi^{e\prime}_i)=2\pi N_{\text{edges}}\,, \tag{3.93}$$

where

$$\sum_{\text{s.angles}}\Omega^e\quad\text{and}\quad\sum_{\text{d.angles}}\phi^e$$

are respectively the sums over all solid and dihedral angles of the element and $\Omega^{e\prime}_i$ and $\phi^{e\prime}_i$ are respectively the exterior solid angle at nodes i and the exterior dihedral angle at edges i (one obviously has $\phi^e_i+\phi^{e\prime}_i=2\pi$ and $\Omega^e_i+\Omega^{e\prime}_i=4\pi$).

One can therefore write

$$\begin{aligned} 2\sum_{i=1}^{N^e_{\text{edges}}}\phi^{e\prime}_i-\sum_{i=1}^{N^e_{\text{nodes}}}\Omega^{e\prime}_i&=\left(2\sum_{\text{d.angles}}\phi^e-\sum_{\text{s.angles}}\Omega^e\right)-\left(2\sum_{i=0}^{N^e_{\text{edges}}}\phi^e_i-\sum_{i=1}^{N^e_{\text{nodes}}}\Omega^e_i\right)\\ &=4\pi\left(N^e_{\text{edges}}-N^e_{\text{nodes}}-\frac{N^e_{\text{facets}}}{2}+1\right)\\ &=4\pi\left(\frac{N^e_{\text{facets}}}{2}-1\right)\,. \end{aligned} \tag{3.94}$$

[12] Facets are edges and faces, respectively for polygons and polyhedra.

The second equality is obtained by replacing the term in the first parenthesis by Eqs. (3.92) and (3.93), and the term in the second parenthesis using Gram Euler Theorem, Eq. (3.90). The second equality is obtained using the Euler characteristic equation $N^e_{\text{edges}} - N^e_{\text{nodes}} = N^e_{\text{facets}} - 2$. Equation (3.94) provides the same result as for the interior angles (Eq. (3.90)) and is useful to study 3D meshes, where the interior and exterior boundaries are considered as polyhedra.

For 2D meshes, some relations have been found between the interior and exterior edges of a mesh, regarding the total number of triangular faces (see Eq. (3.83)). Similarly, for 3D meshes, relations between the interior and exterior faces can be derived. The practical application of such relations is the knowledge of the relative importance, in count, of elements adjacent to boundaries, as the mesh adaptation algorithm involves more operations on those elements. For instance, suppose that one has a mesh separated in N_{dom} domains, and that there are a total number N_{holes} of polygonal holes (how the holes are distributed among the different domains does not matter). The domain and the holes therefore consist in $N_{\text{dom}} + N_{\text{holes}}$ polyhedra. For the domain and hole boundaries, the focus is on the total solid angle and dihedral angle sums, respectively on the exterior and the interior of the polyhedra defined by their boundary edges. Equation (3.94) has to be applied $N_{\text{dom}} + N_{\text{holes}}$ times[13], which amounts to replace the term -1 by $-N_{\text{dom}} - N_{\text{holes}}$ in this equation. Considering $N_{\text{b.nodes}}$, $N_{\text{b.edges}}$, and $N_{\text{b.facets}}$, i.e., respectively the number of boundary nodes, edges and facets, one has

$$2 \sum_{i=1}^{N_{\text{b.edges}}} \phi_i^{b\prime} - \sum_{i=1}^{N_{\text{b.nodes}}} \Omega_i^{b\prime} = 4\pi \left(\frac{N_{\text{b.facets}}}{2} - N_{\text{dom}} - N_{\text{holes}} \right) , \tag{3.95}$$

where $\phi_i^{b\prime}$ is the exterior dihedral angle at the boundary edge i and $\Omega_i^{b\prime}$ is the exterior solid angle at the boundary node i. Once these contributions have been taken into account, the remaining part consist in the contribution of the N_{elems} internal cells. The contribution of one element being given by Eq. (3.90), the following relation holds for the total contribution of the mesh elements (assuming each mesh element all have the same number of facets)

$$\sum_{e \in \mathcal{E}} \left(2 \sum_{i=1}^{N^e_{\text{edges}}} \phi_i^e - \sum_{i=1}^{N^e_{\text{nodes}}} \Omega_i^e \right) = 4\pi \, N_{\text{elems}} \left(\frac{N^e_{\text{facets}}}{2} - 1 \right) , \tag{3.96}$$

$$\overset{\text{if } \Delta \text{ el.}}{=} 4\pi \, N_{\text{elems},\Delta} \, . \tag{3.97}$$

The second equality, Eq. (3.101) corresponds to the specific case of tetrahedral elements. On the other hand, the global sum gives the differences between the total number of vertices and edges, through Eqs. (3.92) and (3.93). One can thus write

$$\begin{aligned} 4\pi(N_{\text{edges}} - N_{\text{nodes}}) &= 2 \sum_{\text{d.angles}} \phi^e - \sum_{\text{s.angles}} \Omega^e \\ &= \sum_{e \in \mathcal{E}} \left(2 \sum_{i=1}^{N^e_{\text{edges}}} \phi_i^e - \sum_{i=1}^{N^e_{\text{nodes}}} \Omega_i^e \right) + \left(2 \sum_{i=1}^{N_{\text{b.edges}}} \phi_i^{b\prime} - \sum_{i=1}^{N_{\text{b.nodes}}} \Omega_i^{b\prime} \right) , \qquad (3.98) \\ &= 4\pi \, N_{\text{elems},\Delta} + 4\pi \left(\frac{N_{\text{b.facets}}}{2} - N_{\text{dom}} - N_{\text{holes}} \right) \\ &= 4\pi \left(N_{\text{elems},\Delta} + \frac{N_{\text{b.facets}}}{2} - N_{\text{dom}} - N_{\text{holes}} \right) , \qquad (3.99) \end{aligned}$$

where the second equality is obtained replacing the first and the second terms on the right hand side, by Eq. (3.97) and Eq. (3.95), respectively. It is the 3D equivalent of Eq. (3.78). Furthermore, the Euler identity for 3 dimensional CW-complexes can be used [68], similarly to what has been done in Section 3.4.1 for the 2D case[14]. For 3D graphs, this identity writes

$$N_{\text{nodes,g}} - N_{\text{edges,g}} + N_{\text{facets,g}} - N_{\text{cells,g}} = 0 \, , \tag{3.100}$$

[13] The same formula holds for the interior and the exterior.

[14] The planar graphs used in 2D are two dimensional CW-complexes

where $N_{\text{nodes,g}}, N_{\text{edges,g}}, N_{\text{facets,g}}, N_{\text{cells,g}}$ are respectively the number of vertices (nodes), edges, faces, and cells of the 3D graph. For the formula to be valid, the mesh should be connected into one single part (such that a path always exist, following edges, between two arbitrary chosen vertices). As in the present example, if N_{dom} separated domains are considered, $N_{\text{dom}} - 1$ additional fictive edges are needed to make the graph connected (see Fig. 3.26 for an example in 2D), such that $N_{\text{edges,g}}$, in the Euler formula, is replaced by $(N_{\text{edges}} + N_{\text{dom}} - 1)$. On the other hand, the formula considers $N_{\text{cells,g}}$ as all the cells that are filling the space, including the holes in the mesh as well as the space surrounding the mesh (again, the situation is completely analogous to the 2D case, as shown above). For this reason, it should be replaced by $(N_{\text{elems}} + N_{\text{holes}} + 1)$. Finally, the Euler equation writes as

$$N_{\text{nodes}} - N_{\text{edges}} + N_{\text{facets}} - N_{\text{elems}} = N_{\text{dom}} + N_{\text{holes}} \,. \tag{3.101}$$

Combining this equation with Eq. (3.99) (and therefore replacing N_{elems} by $N_{\text{elems},\Delta}$ as we are in the special case of tetrahedral elements), dividing by 4π and rearranging the terms gives

$$N_{\text{facets}} = 2N_{\text{elems},\Delta} + \frac{N_{\text{b.facets}}}{2} \,, \tag{3.102}$$

where N_{dom} and N_{holes} have cancelled out. Now, rewriting N_{facets} as the sum of all internal and boundary contributions,

$$N_{\text{facets}} = N_{\text{i.facets}} + N_{\text{b.facets}} \,, \tag{3.103}$$

one has

$$N_{\text{i.facets}} = 2N_{\text{elems},\Delta} - \frac{N_{\text{b.facets}}}{2} \,, \tag{3.104}$$

which is the 3D equivalent of Eq. (3.83). Again, this relation could have been obtained by considering that there are two faces ($4 \times \frac{1}{2}$) per cell as each face is shared between two adjacent tetrahedra and that they have four faces. The missing faces in the total count are the boundary faces and one half of them have to be added in the count since they have been counted only one half, as they belong only to one tetrahedron. It directly gives Eq. (3.102), which then could have been used with Eq. (3.99) to demonstrate the Euler invariant given by Eq. (3.100).

As already mentioned in Section 3.4.1 for the 2D case, it should be emphasized that relations similar to Eq. (3.104) can be useful, in the context of meshes, to evaluate the complexity of the remeshing algorithm.

Particular case of a large mesh

If large meshes are used, such that the contribution of the boundary may be neglected, approximate one-to-one relations between mesh entities can be found. One should first note that, despite the fact that the nodal fraction f_N^e is not constant among the elements, the following approximation can be made using an averaged value $\overline{f_N}$:

$$\frac{1}{4\pi} \sum_{\text{s.angles}} \Omega^e \simeq \sum_{e \in \mathcal{E}} \frac{1}{4\pi} \sum_{i=1}^{N^e_{\text{nodes}}} \Omega^e_i \tag{3.105}$$

$$\Longleftrightarrow$$

$$N_{\text{nodes}} \simeq \overline{f_N} N_{\text{elems}} \,. \tag{3.106}$$

The same procedure can be performed for the edges by defining an elemental edge fraction f_E^e whose average is

$$\frac{1}{2\pi} \sum_{\text{d.angles}} \phi^e \simeq \sum_{e \in \mathcal{E}} \frac{1}{2\pi} \sum_{i=1}^{N^e_{\text{edges}}} \phi^e_i \tag{3.107}$$

$$\Longleftrightarrow$$

$$N_{\text{edges}} \simeq \overline{f_E} N_{\text{elems}} \,. \tag{3.108}$$

A relation between $\overline{f_N}$ and $\overline{f_E}$ can then be obtained considering Eq. (3.97), which gives

$$\frac{1}{4\pi}\sum_{e\in\mathcal{E}}\left(2\sum_{i=1}^{N^e_{\text{edges}}}\phi_i^e-\sum_{i=1}^{N^e_{\text{nodes}}}\Omega_i^e\right)=N_{\text{elems}}\left(\frac{N^e_{\text{facets}}}{2}-1\right) \tag{3.109}$$

$$\Longleftrightarrow$$

$$N_{\text{elems}}\left(\overline{f_E}-\overline{f_N}\right)=N_{\text{elems}}\left(\frac{N^e_{\text{facets}}}{2}-1\right) \tag{3.110}$$

$$\Longleftrightarrow$$

$$\left(\overline{f_E}-f_N\right)=\left(\frac{N^e_{\text{facets}}}{2}-1\right). \tag{3.111}$$

In the particular case of tetrahedral meshes ($N^e_{\text{facets}}=4$), this gives $\overline{f_E}=\overline{f_N}+1$. In this case, one has the following one-to-one relations between the different mesh entities:

$$\text{from Eq. (3.106)}: N_{\text{nodes}}\simeq\overline{f_N}\,N_{\text{elems},\Delta}\,, \tag{3.112}$$

$$\text{from Eq. (3.108)}: N_{\text{edges}}\simeq\left(\overline{f_N}+1\right)N_{\text{elems},\Delta}\,, \tag{3.113}$$

$$\text{from Eq. (3.104)}: N_{\text{facets}}\simeq 2\,N_{\text{elems},\Delta} \tag{3.114}$$

$$\text{from Eqs. above}: N_{\text{edges}}\simeq\left(\frac{\overline{f_N}+1}{\overline{f_N}}\right)N_{\text{nodes}}\,. \tag{3.115}$$

$$\text{from Eqs. above}: N_{\text{nodes}}\simeq\frac{\overline{f_N}}{2}N_{\text{facets}} \tag{3.116}$$

$$\text{from Eqs. above}: N_{\text{edges}}\simeq\left(\frac{\overline{f_N}+1}{2\overline{f_N}}\right)N_{\text{facets}} \tag{3.117}$$

To estimate $\overline{f_N}$, a 3D tetrahedral mesh has been build with gmsh inside a cube of unit side length with 50 subdivisions, using a Delaunay building procedure.[15] This leads to a mesh containing ~ 98000 nodes and ~ 590000 elements suggesting $\overline{f_N}\simeq\frac{98000}{590000}\simeq\frac{1}{6}$. Note that a theoretical estimation of $\overline{f_N}$ is given in the next paragraph.

3.4.3 Application of the concept of nodal density to mesh adaptation of unstructured tetrahedral meshes

The concept of nodal density in 3D can be used to ensure good mesh quality so as to reduce the space discretization error. In the present context, the distortion of the mesh between two time steps degrades its quality and the cloud of nodes need to be remeshed using 3D Delaunay triangulation. Moreover, when adding or removing nodes for mesh adaptation, the spatial variation of the node distribution should be sufficiently smooth to ensure a high quality Delaunay triangulation. The goal of this section is therefore to clarify how to properly add and remove nodes in 3D meshes by leveraging the concept of nodal density.

Rather intuitively, the more evenly spaced from each other the nodes are, the better the element quality is, irrespective of the type of quality measure used.[16] A notable exception is the specific case of slivers, which are discussed later.

As aforementioned, the nodal fraction f_N^τ of a tetrahedron τ involves the interior solid angles sum:

$$f_N^\tau=\frac{\sum_{i=1}^4\Omega_i^\tau}{4\pi}\,. \tag{3.118}$$

Again, unlike the two-dimensional case, the sum of all the interior solid angles Ω_i is not fixed for a given polyhedron type but depends on its specific shape. In particular, it is related to the sum of the dihedral

[15] `https://gmsh.info/`

[16] Several ways can be used to quantify the element quality, which can be found in `https://coreform.com/cubit_help/mesh_generation/mesh_quality_assessment/tetrahedral_metrics.htm`.

angles ϕ_i through the Gram-Euler formula (Eq. (3.90)), which, in the particular case of a tetrahedron, reads

$$2\sum_{i=1}^{6}\phi_i^\tau - \sum_{i=1}^{4}\Omega_i^\tau = 4\pi\,. \tag{3.119}$$

The fraction of nodes f_N^τ inside the tetrahedron can therefore be expressed as

$$f_N^\tau = \frac{\sum_{i=1}^{6}\phi_i^\tau}{2\pi} - 1\,. \tag{3.120}$$

Consequently, either the solid angles or the dihedral angles can be used to compute f_N^τ, depending on which approach (Eq. (3.118) or (3.120)) is the most efficient. On the one hand, solid angles between three edges of unitary director vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ can be computed with the formula [72]

$$\Omega(\mathbf{a},\mathbf{b},\mathbf{c}) = 2\arctan\left(\frac{|\,\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,|}{1+(\mathbf{a}\cdot\mathbf{b})+(\mathbf{a}\cdot\mathbf{c})+(\mathbf{b}\cdot\mathbf{c})}\right)\,, \tag{3.121}$$

where $|\,\mathbf{a}\,\mathbf{b}\,\mathbf{c}\,|$ is the scalar triple product of vectors $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$. On the other hand, the dihedral angle $\phi_{i,j}$ between the faces with normals $\mathbf{n}_i$ and $\mathbf{n}_j$ is given by[17]

$$\phi_{i,j} = \pi - \arccos(\mathbf{n}_i\cdot\mathbf{n}_j)\,. \tag{3.122}$$

Once f_N^τ is computed, the nodal density of a tetrahedron element is given by

$$\sigma_{3D}^\tau \triangleq \frac{f_N^\tau}{V^\tau}\,, \tag{3.123}$$

where V^τ is the tetrahedron volume. This nodal density thus represents an ideal indicator to determine whether the mesh should be locally refined, locally coarsened or remain unchanged. This can be done by comparing the local nodal density to a local target nodal density σ^* and introducing threshold values for adding or removing nodes.

Target mesh size and target nodal density.

The idea behind the definition of the target mesh size in 3D is very similar to that in 2D. While in 2D, the target mesh size was defined as the square root of the target surface area, here, in 3D, it should be defined as the cubic root of the "ideal tetrahedron" target volume V_{ideal}^τ:

$$L^* = \sqrt[3]{V_{\text{ideal}}^\tau}\,. \tag{3.124}$$

An ideal tetrahedron should be understood as a tetrahedron with a nodal fraction f_N^τ corresponding to some average value $\overline{f_N}$, such that the target nodal density σ^* is given by

$$\sigma^* = \frac{\overline{f_N}}{L^{*3}}\,. \tag{3.125}$$

As aforementioned, statistics performed on elements of a large mesh suggest $\overline{f_N} = \frac{1}{6}$. Another estimation can be performed using particular sets of space filling tetrahedra, i.e., tetrahedra that can be put next to each other to fill the entire space. To define such reference tetrahedron, two particular cases of tetrahedra filling a cube are chosen, as it is trivially understood that cubes can fill the space.

The first example is illustrated in the left image of Fig. 3.29. At each face F_i of the cube $ABCDEFGH$, a vertex P_i is added, which counts as 1/2 since it is shared among two cubes. At the center of the cube, a vertex O is added and counts as 1 since it is entirely contained in the cube. Finally, the original

[17] It seems at first glance that the second option has a smaller complexity and may be the method of choice to compute the nodal fraction, since it involves only six scalar products, compared to twelve for the first option (three at the denominator times four vertices), without considering the other mathematical operations. Moreover, face normals are usually readily available.

vertices of the cube (A, B, C,...) are shared among 8 cubes in total, such that their sum counts as 1. The total number of nodes is therefore $1 + 6 \times \frac{1}{2} + 1 = 5$. It can be seen that, with the additional vertices (O, P_1, P_2,...), the cubes can be divided into 24 similar tetrahedra. For instance, the pyramids $OFGHE$ with its base on the face $GHEF$ is divided into 4 tetrahedra containing the edge OP_1. Since there is such a pyramid adjacent to each face, there are 4×6 tetrahedra in total, leading to a fraction of nodes per tetrahedron

$$f_N = \frac{5}{24} \simeq 0.2\,. \tag{3.126}$$

The second example is simpler and consists in observing that the cube can be divided in 4 similar tetrahedra ($AGEF$, $ABCG$, $CHEG$, $ACDE$) and one larger tetrahedron ($ACEG$), giving a total of 5 tetrahedra for 1 node (the sum of all the solid angles being 4π). It yields the average nodal fraction

$$\overline{f_N} = \frac{1}{5} = 0.2\,, \tag{3.127}$$

which is almost identical to the first result. Note that it is a bit higher than the nodal fraction of a regular tetrahedron, which is given by[18] $\arccos(23/27)/\pi \simeq 0.175$.

Despite the fact that this value is slightly different than the one previously obtained through statistics ($\overline{f_N} = 1/6$), it is suggested to use it as the reference value[19], such that one has

$$\overline{f_N} = 0.2 \tag{3.128}$$

and

$$\sigma^* = \frac{0.2}{L^{*3}}\,. \tag{3.129}$$

Note that in 2D, these values are respectively $1/2$ and $1/(2L^{*2})$.

Once this target nodal density has been defined, higher and lower bounds, beyond which the mesh is respectively coarsened or refined, have to be defined. The exact value of these thresholds depends on the specific refinement and coarsening process. Therefore, the possible extension to 3D of the node addition and removal process and the generalization of the boundary recognition algorithm are first discussed. Upper and lower bounds for the nodal density are then proposed as an example. They are also applied to the special case of slivers, that are badly shaped tetrahedra that cannot be eliminated by the 3D Delaunay triangulation.

3.4.4 3D generalization of the mesh refinement, coarsening, and boundary recognition algorithms

A possible 3D generalization of the 2D algorithms described in the beginning of this chapter is briefly described. The generalization of the mesh refinement and mesh coarsening steps are first discussed, followed by that of the boundary recognition algorithm

The geometric and solution-based criteria introduced at the beginning of this chapter for imposing the target mesh size can be readily used in 3D. The idea is then to modify the local nodal density by locally adding or removing nodes, such that this nodal density remains close to the target value $0.2/L^{*3}$, avoiding too large changes of the nodal density between adjacent elements. Therefore, the algorithm for adding and removing nodes should be designed such that the local modification of the nodal density remains limited.

Node addition

For mesh refinement, the general idea would be to add one node that affect several elements rather than a single element, such that the added node impacts a larger portion of the space, thereby limiting the resulting increase in nodal density. In two dimensions, the surface area considered is that of two adjacent triangles and the node is added in the middle of their common edge if a given area threshold

[18] This value could also be used as a reference as it is between our 2 estimations 1/6 and 1/5, despite the fact that regular tetrahedron cannot fill the space on themselves

[19] Note that slightly different values (for instance 1/6 or 0.175) could also be considered.

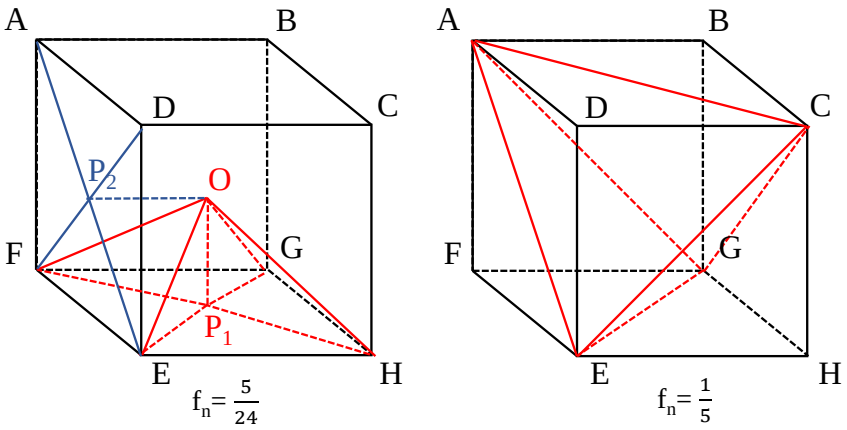


Figure 3.29: Left : Illustration of the partition of a cube into 24 similar tetrahedra (only the four associated with the face EFGH are shown in red). Right: Illustration of the partition of a cube into four similar trirectangular tetrahedra (in red) at four of the cube corners (B, D, F, H), and one central regular tetrahedron.

is exceeded. Here, one choice could be to add a node on the common face of two tetrahedra, in which case the volume impacted is the sum of the volume of the two tetrahedra. An alternative option could be to add a node on a common edge of several tetrahedra, in which case the volume impacted is the sum of the volumes of all adjacent elements. As it involves a larger volume, the second method has a smaller impact on the nodal density increase; it is also computationally more expensive as it requires computing a larger number of solid angle sums. In both cases, the local nodal density σ_l before node addition is evaluated as

$$\sigma_\mathrm{l} = \frac{\sum_{i=1}^{N_\mathrm{ref.}} \sigma_i \, V_i}{\sum_{i=1}^{N_\mathrm{ref.}} V_i} = \frac{\sum_{i=1}^{N_\mathrm{ref.}} f_{N,i}}{\sum_{i=1}^{N_\mathrm{ref.}} V_i} \,, \tag{3.130}$$

where N_ref is the number of elements impacted. As one node is added in the compounded volume, the increment of nodal density achieved by such a procedure is

$$\Delta\sigma = \left(\sum_{i=1}^{N_\mathrm{ref.}} V_i \right)^{-1} \,, \tag{3.131}$$

such that the nodal density after the refinement process, σ_r, is given by

$$\sigma_\mathrm{r} = \sigma_\mathrm{l} + \Delta\sigma \,. \tag{3.132}$$

For convenience, the lower threshold value σ_th^-, to which σ_l could be compared, can be defined such that a node is only added if the resulting nodal density σ_r does not exceed the target value σ^* given by Eq. (3.129):

$$\sigma^* \geq \sigma_\mathrm{th}^- + \Delta\sigma \,, \tag{3.133}$$

or equivalently,

$$\sigma_\mathrm{th}^- \leq \sigma^* - \Delta\sigma \,. \tag{3.134}$$

It should be emphasize that, unlike in 2D, the above found for σ_th^- is case-dependent as $\Delta\sigma$ depends on the volumes of the elements considered. In practice, a node should thus be added if $\sigma_l \leq \sigma_\mathrm{th}^-$ with the threshold

$$\sigma_\mathrm{th}^- = k_\mathrm{r}(\sigma^* - \Delta\,\sigma) \,, \tag{3.135}$$

where $k_\mathrm{r} \leq 1$ is a constant to be determined based on the practical application of the proposed algorithm.

Node removal

Node removal is more challenging. While for the addition of nodes, the volume impacted by the new inserted node is well defined and well separated, this is not the case for the removal of nodes as it impacts the surrounding elements. It is assumed that the 3D algorithm would follow a similar approach to the 2D cell collapsing described in Section 3.2.2.2. When the nodal density of a tetrahedron is higher than a given threshold, this element is tagged to be collapsed, which means that each of its nodes are removed and replaced by a single node at its center. Afterwards, each first and secondary neighbours are tagged to prevent them from being collapsed (as already mentioned in Section 3.2.2.2, collapsing all the elements in some region would lead to mesh refinement and not mesh coarsening, which also holds in 3D). The removed element and its first and secondary neighbours correspond to a total number of elements

$$N_{\text{coars.}} = 1 + (\mathcal{D}+1)^2\,, \tag{3.136}$$

where $\mathcal{D}$ is the number of dimensions of the mesh (2 or 3), which gives respectively 10 and 17 for 2D and 3D meshes, respectively. The local nodal fraction $f_{N,\text{l}}$ for the compounded volume of these elements is

$$f_{N,\text{l}} = \sum_{i=1}^{N_{\text{coars.}}} f_{N,i} \simeq N_{\text{coars.}}\,\overline{f_N}\,, \tag{3.137}$$

where it has been assumed in the last equality that $f_{N,i}$ equals $\overline{f_N}$ on average (0.5 in 2D and 0.2 in 3D). As illustrated in Fig. 3.30, once the central element is collapsed and replaced by a node as its center, it induces a decrease of the nodal fraction in that total volume

$$\Delta f_N = \frac{1}{(\mathcal{D}-1)2\pi}\left(\sum_{i=1}^{\mathcal{D}+1}\sum_{j=1}^{\mathcal{D}^2+1}\Omega_{ij}^{\text{Del}}\right) - 1\,, \tag{3.138}$$

where Ω_{ij}^{Del} is the solid angle (the regular angle in 2D) at node i of element j that shares this node and is either the collapsed element, one of its direct neighbors or one of the elements adjacent to these neighbors. They are $\mathcal{D}^2+1$ such elements for each of the $\mathcal{D}+1$ deleted nodes. The -1 in the above expression corresponds to the contribution of the new node added at the center of the collapsed element. To simplify this expression, it is assumed that, on average, a solid angle (regular angle in 2D) equals $(\mathcal{D}-1)2\pi\,\frac{\overline{f_N}}{\mathcal{D}+1}$ because there are $\mathcal{D}$+1 vertices in a $\mathcal{D}$-simplex (triangles and tetrahedra are respectively 2-simplices and 3-simplices). Considering the $(\mathcal{D}+1)(\mathcal{D}^2+1)$ solid angles, Δf_N can be approximated as

$$\Delta f_N \simeq (\mathcal{D}^2+1)\,\overline{f_N} - 1\,. \tag{3.139}$$

Therefore, the nodal fraction $f_{N,\text{c}}$ after the coarsening process is

$$f_{N,\text{c}} \simeq f_{N,\text{l}} - \Delta\, f_N = \overline{f_N}\left(N_{\text{coars.}} - (\mathcal{D}^2+1)\right) + 1 = \overline{f_N}(2\mathcal{D}+1) + 1\,, \tag{3.140}$$

where Eq. (3.136) has been used to obtain the last equality. For the same compounded volume, the ratio between the nodal densities is equal to the ratio of the nodal fractions,

$$\frac{\sigma_{\text{c}}}{\sigma_{\text{l}}} = \frac{f_{N,\text{c}}}{f_{N,\text{l}}} \simeq \frac{\overline{f_N}(2\mathcal{D}+1)+1}{N_{\text{coars.}}\,\overline{f_N}} = \frac{1}{1+(\mathcal{D}+1)^2}\left((2\mathcal{D}+1) + \frac{1}{\overline{f_N}}\right)\,, \tag{3.141}$$

which gives respectively 7/10 and 12/17 for the 2D and 3D[20] cases. These values are close to each other such that a value of 0.7 could be used in both cases. Practically speaking, it means that the algorithm provides, as a first approximation, a reduction of the nodal density by a factor 0.7,

$$\sigma_{\text{c}} = 0.7\sigma_{\text{l}}\,. \tag{3.142}$$

The upper threshold of the nodal density before mesh coarsening is chosen by imposing that the coarsened value σ_{c} should be higher or equal to the target nodal density σ^*, so as to avoid too similar

[20] 13/17 for $\overline{f_N} = 1/6$.

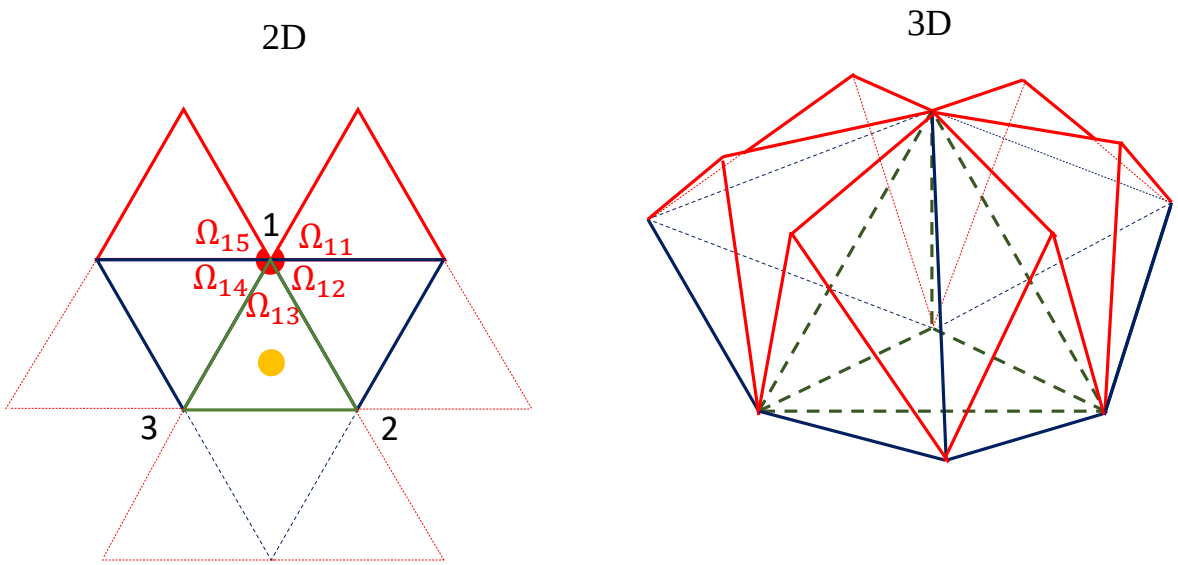


Figure 3.30: Illustration of the solid angle (planar angles in $2D$) count around each deleted vertex, among those of the tagged elements plus the central one, both in $2D$ (left) and $3D$ (right). On the left image, the dashed lines correspond to the elements involved in the solid angle counts around vertex 2 and 3. Their 3D equivalent are not represented on the right for visibility. On the right image, the dashed lines are the edges that are not visible in the foreground. The collapsed element is only delimited by green lines and its direct adjacent neighbours by $(2\mathcal{D}-3)$ green lines and $\mathcal{D}$ blue lines ($\mathcal{D}$ being the mesh dimensions, i.e., 2 or 3). The other elements adjacent to these direct neighbors are delimited by $\mathcal{D}$ red lines, $\mathcal{D}-1$ blue lines and $(\mathcal{D}-2)$ green lines. It can be easily verified that the total number of solid angles equals $(\mathcal{D}+1)(\mathcal{D}^2+1)$.

nodal densities after mesh refinement and mesh coarsening (a situation that could lead to node addition and node removal cycles in a given region of the space, and a resulting large numerical dissipation). This condition writes

$$\sigma_{\text{th}}^{+} \geq \frac{\sigma^*}{0.7} = 1.43\sigma^* \,. \tag{3.143}$$

Finally, a coarsening constant $k_{\text{c}} > 1$ can be introduced to exactly determine the threshold so that the mesh is coarsened if $\sigma_{\text{l}} > \sigma_{\text{th}}^{+}$ with

$$\sigma_{\text{th}}^{+} = 1.43 k_{\text{c}} \sigma^* \,. \tag{3.144}$$

The constant k_{c} should be determined based on the practical application of the proposed algorithm.

Remarks

The proposed approach is very theoretical and has not been implemented and tested yet. In particular, uncertainties still remain concerning the computation of the nodal densities, especially in 3D and for the coarsening algorithm. The practical application of the algorithm is therefore required to determine adequate values for the thresholds k_{r} and k_{c}. Additionally, another value for $f_{N\text{ref}}$ may be more adapted than 0.2 (for instance, motivations to use 1/6 also exist, as previously discussed), and may require some statistics applied to tetrahedral meshes. Nevertheless, the proposed algorithm is based on the concept of nodal density, which is a solid tool to derive relations between the number of different mesh entities (i.e., nodes, edges, facets, and cells). It is therefore hoped that it will be used in future work involving three dimensional unstructured meshes. To the author best knowledge, the proposed idea of using the fractional number per elements based on the solid angles has not yet been proposed in the literature in the context of mesh-based numerical simulations, and is a novel concept. It aims to provide a general framework for metric-based mesh adaptation, where the accurate measure of the local nodal density should enables finding a good trade-off between mesh quality and numerical dissipation (see Section 3.2.2.2 for an example in 2D).

3D boundary recognition algorithm

The extension of the boundary recognition algorithm to 3D would be very similar too and follow the same philosophy as that in 2D (see Fig. 3.17). The three types of tetrahedra would be classified as follows. The first category would include tetrahedra with two or fewer nodes on a boundary and would be automatically kept in the triangulation similarly to triangles having at most one boundary node in the 2D case. The next category would encompass tetrahedra having exactly three nodes (i.e., one face) on a boundary; they would be considered a priori as fluid elements. Finally, tetrahedra having four nodes on the boundary would a priori be considered as empty space. The following criteria for these last two categories of elements would be very similar to those in 2D. First, the α-shape criterion would be defined by Eq. (3.34) where the target mesh size L^*_{elem} would be prescribed by some geometrical and solution-based relations.

Afterwards, the elements having 3 nodes on a boundary should be removed if the α-shape criterion is violated, if they have their largest face on the boundary, and if they reach the minimal mesh resolution (similarly to the largest edge criterion in 2D). Moreover, in the case where the last criterion (minimal mesh resolution) is not satisfied, a node would be added in the middle of the boundary faces that are stretched to keep refining the free surfaces without inducing any volume loss.

Lastly, as in 2D, the elements having their four nodes on a boundary could be considered as fluid if they match the resolution of their neighbours, and they could be used as collision anticipation elements otherwise. In summary, every box in Fig. 3.17 could be adapted to a "3D equivalent".

Sliver elements

A last important point to mention about 3D triangulation is the possible presence of badly shaped elements called slivers. These elements lead to inaccurate results and can cause stability problems in the case of artificially compressible schemes [73]; therefore, they should be cleaned out from the triangulation. Slivers consist in tetrahedra having all their nodes close to the equator of their circumscribed sphere such that they have a "flat" shape, as illustrated in Fig. 3.31. As it can be seen in this figure, they are characterized by some very small dihedral angles and a small volume, which can be used as criterion to detect them. The presence of slivers is actually an interesting result illustrating well the concept of nodal density. In particular, even if the nodes are quasi evenly distributed, resulting in a smooth variation of the nodal density, very flat elements can exist as their small volume can be compensated by a small solid angle sum (see Eq. (3.123)). This is the case for slivers of type 2 as illustrated by the right image of Fig. 3.31, while slivers of type 1 (left part of the figure) are removed by the proposed coarsening algorithm because of the higher nodal density (see the proof below, summarized by Fig. 3.32b). Because the nodal density of type 2 slivers is close to that of their neighbors, they should not be eliminated by adding or removing nodes (operations that should only apply when σ_{l} deviates from σ^*), but rather by rearranging locally the disposition of the nodes. Different ways of doing it could be considered but their discussion would go beyond the scope of this thesis. For some literature on the subject, the reader is referred for instance to Meduri et al. [73] who have proposed to inflate the slivers by solving a local elastic system based on prescribed stiffness constants for the different edges. A summary of various methods to remove slivers can also be found in Tournois et al. [74]. Another possibility, as recently proposed in the context of the X-mesh method [14] (see Chapter 1), is to keep the slivers in the triangulation instead of removing them, making sure that, through special treatments, they do not deteriorate the local accuracy of the solution.

This section concludes with an example illustrating how the coarsening algorithm is able to eliminate slivers of type 1. By definition slivers are tetrahedra whose four nodes are close to a great circle of the circumscribed sphere. To represent this, the nodes A, B and C of the tetrahedron are assumed to lie on a great circle of radius 1, as illustrated in Fig. 3.32a. Tetrahedra having numerous degrees of freedom, the exhaustive analysis of all of them would be tedious. For simplicity, it is assumed that A, B, C form

an equilateral triangle with coordinates

$$A = (0, 1, 0)\,, \tag{3.145}$$

$$B = (\frac{\sqrt{3}}{2}, \frac{-1}{2}, 0)\,, \tag{3.146}$$

$$C = (\frac{-\sqrt{3}}{2}, \frac{-1}{2}, 0)\,. \tag{3.147}$$

The only remaining degrees of freedom are the angles ϕ and θ of point D lying on the circumscribed sphere of radius 1:

$$D = (\cos\theta\cos\phi, \cos\theta\sin\phi, \sin\theta)\,. \tag{3.148}$$

With this setup, the nodal density defined by Eq. (3.123) is a function of ϕ and θ where the solid angles can be obtained from Eq. (3.121). The maximum volume $V_{\max}$ is obtained when point D is at its maximum height (the element volume is independent of ϕ as its height, the z-coordinate of point D, remains unchanged as ϕ varies), which corresponds to $\theta = \pi/2$. For the given numerical values, one has $V_{\max} \simeq 0.433$. As the mesh is assumed Delaunay, there is no other nodes in the circumscribed sphere, such that the volume of the adjacent elements is assumed to be of the order of $V_{\max}$. This maximum volume can thus be used as a reference volume to define the target mesh size,

$$L^{*3} = V_{\max} \simeq 0.433\,, \tag{3.149}$$

which corresponds to about 10% of the sphere volume. The corresponding reference nodal density is therefore equal to 0.462 from Eq. (3.129) with the numerical value given by Eq. (3.149). Taking for instance $k_c = 1$, the upper threshold value of the nodal density for mesh coarsening, σ^+_{th}, is computed according to Eq. (3.144) leading to $\sigma^+_{\text{th}} = 0.66$.

The different nodal densities for the different possible tetrahedra ABCD are plotted in terms of ϕ for different θ levels in Fig. 3.32b, where the upper bound $\sigma^+_{\text{th}} = 0.66$ is indicated by a black dashed line. The proposed approach enables to eliminate slivers of type 1 in Fig. 3.31, provided that the angle θ is less than 40°. As illustrated in Fig. 3.32b, the nodal density increases each times point D passes above one of the three other nodes A, C and B, respectively at angles $\phi = \frac{\pi}{2}, \phi = \frac{7\pi}{6}$ and $\phi = \frac{11\pi}{6}$, such that the nodal density becomes larger than the thresholds for $\theta < 40°$. Moreover, the algorithm does not remove the elements with $\theta > 40°$, but those are not really flat and cannot be considered as slivers. Consequently, they should not be removed.

Finally, it should be emphasized that, in this analysis, the volume of the sliver elements (for instance with $\theta < 40°$) is assumed to be significantly smaller than L^{*3}_{elem}. This assumption is expected to hold far from boundaries, similarly to the 2D case. Close to boundaries, special treatment analogous to Eq. (3.32) could be considered.

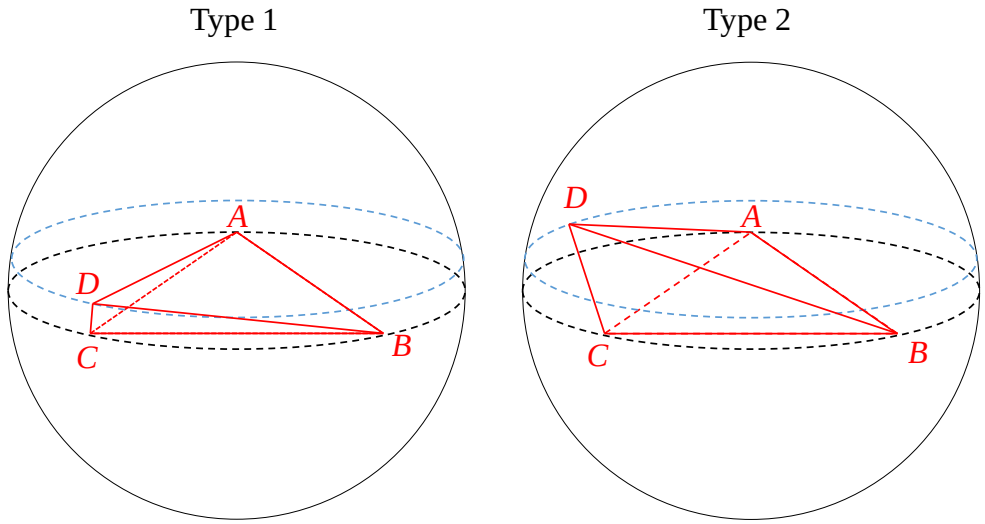
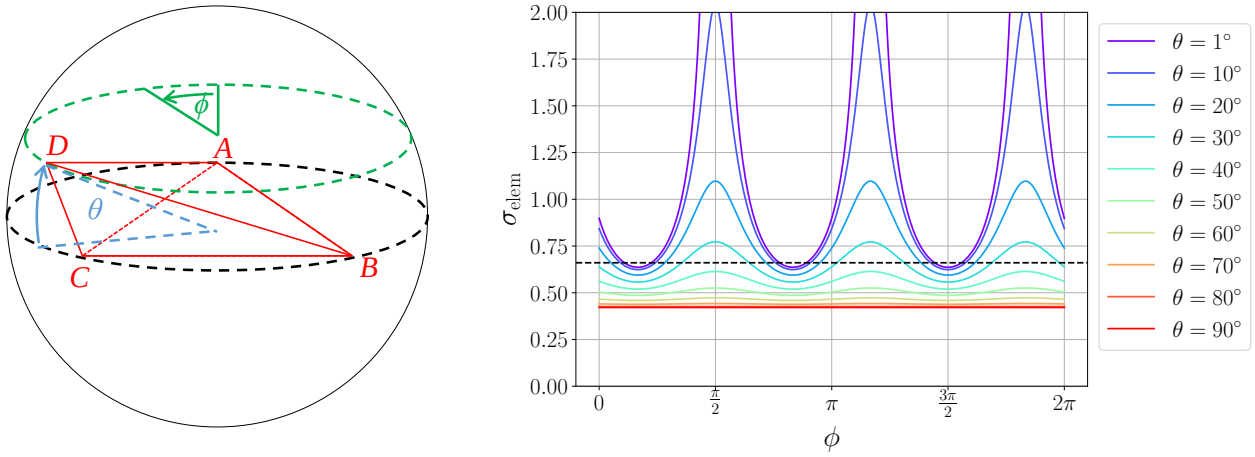


Figure 3.31: Illustration of the two types of slivers, i.e., tetrahedra with all vertices close to a great circle of the circumscribed sphere (black dashed line). The first type of sliver, on the left, is characterized by two vertices close to each other (C and D), and is therefore characterized by a high nodal density. The second type of slivers is characterized by well-spaced vertices and is therefore associated with a small nodal density.



(a) **(b)**

Figure 3.32: (**a**) Illustration of the remaining degrees of freedom ϕ and θ of vertex D, the points A, B and C being set on a great circle of the sphere, and forming an equilateral triangle. (**b**) Variation of the nodal density as the node D turns along the longitude ϕ for different latitudes θ, as illustrated in (**a**). The nodal density increases at angles ϕ of $\pi/2$, $7\pi/6$ and $11\pi/6$, i.e., when point D is closest to points A, C and B, respectively. The black dashed line in (**b**) represents the nodal density threshold, $\sigma_{\text{th}}^{+} = 0.66$, for node removal.

Chapter 4

Validation of the 2D mesh adaptation algorithm

The goal of this chapter is to illustrate the advantages of the new mesh adaptation algorithm detailed in Chapter 3 and to validate it. In particular, six two-dimensional test cases are considered, of which three focus on bounded flows to demonstrate the mesh adaptation capabilities, and three deal with free-surface flows to highlight the improved free-surface description and mass conservation properties of the new algorithm. The validation relies on comparisons with the classical PFEM and with results from the literature.

Flows in bounded domain

First of all, the focus is solely on the capability of the new algorithm to provide sufficient spatial resolution in regions where the flow is rapidly changing, while keeping a limited increase of the total number of nodes. As this goal is independent of the accurate representation of free surfaces and the limitation of mass conservation errors coming from the boundary recognition step, such validation can be performed using confined domains whose external boundaries do not change in time. The advantage is that such flows are widely covered by the literature, because they can be easily simulated with classical CFD methods such as the finite difference or the finite volume methods (to cite only those), within an Eulerian formalism. The following cases are considered:

- the flow around a static cylinder at low Reynolds number ($Re_D \leq 200$) and around a rotating/oscillating cylinder at $Re_D = 200$;
- the flow around an impulsively started cylinder at $Re_D = 9500$;
- the lid-driven cavity flow at $Re = 100$ and $Re = 400$.

The first two cases are described in Section 4.1 and the last one in Section 4.2.

Free-surface flows

The mesh adaptation algorithm has also been developed in order to reduce the mass conservation errors inherent to the boundary recognition technique of the PFEM, and to provide a more accurate description of free surfaces. To assess the performance of the algorithm regarding these aspects, some free-surface flows are simulated, for which the literature is scarcer than for flows in a fixed domain. The following cases are considered:

- the sloshing of water occurring in an oscillating reservoir;
- a 2D viscous drop falling into a bath of the same fluid;
- an impulsively started cylinder going out of a bath at constant velocity.

The first goal of these three test cases is to quantify the reduction of the mass conservation error achieved by the new algorithm. The second goal is to demonstrate how the new algorithm enables an accurate description of the free surfaces. Both features are evaluated through comparisons with results from the literature, as well as results obtained with the classical PFEM algorithm on meshes of uniform element size.

Finally, for the last test case, an additional goal is to demonstrate why the mesh adaptation is essential for our application, and to assess the performance of the new algorithm in terms of CPU time. These three test cases are respectively discussed in Sections 4.3, 4.4 and 4.5.

4.1 The flow around a cylinder

The case of a flow around a cylinder is a classical CFD case in the literature and has been the subject of many publications over the past decades. As the amount of literature on this subject is large, only a limited number of references have been considered, mainly at low Reynolds number ($Re \leq 200$). Below this limit, the behavior of the flow can be separated into two regimes (for a static cylinder), as illustrated in Fig. 4.1. The first regime (Fig. 4.1a) is observed for a Reynolds number $Re \leq 40$ and is characterized by a steady symmetrical wake behind the cylinder, while the second regime (Fig. 4.1b) features a regular oscillatory wake made of vortices, called the Von Karmann Vortex street.

Taking into account these aspects, the following cases are considered:

- the flow around a static cylinder for a Reynolds number ranging from 20 to 200: both steady and unsteady regimes are investigated in Section 4.1.1;
- the flow around a rotating cylinder, for two different rotation speeds, at a Reynolds number of 200: with this test case, the simulation of the flow around a moving body takes advantage of the Lagrangian nature of the method (Section 4.1.2);
- the flow around a cylinder oscillating perpendicularly to the flow, at two different oscillation frequencies and amplitudes (Section 4.1.3): here again, the Lagrangian nature of the PFEM is leveraged;
- the flow around an impulsively started cylinder at the larger Reynolds number $Re_D = 9500$ (Section 4.1.4): this case assesses the maximum capacity of the mesh adaptation method for the study of the cylinder wake during the transient regime, where a very fine mesh is required to obtain accurate results.

Numerical setup

The numerical setup is the same for each simulation, except for the last one (impulsively started cylinder at $Re_D = 9500$, discussed in Section 4.1.4), and is summarized in Fig. 4.2. All physical quantities reported below are made non-dimensional using the cylinder diameter D, the free-stream velocity U_∞ and the fluid density ρ. The computational domain is rectangular and the flow goes from left to right. These test cases feature a flow that enters and leaves the computational domain at all far field boundaries. Because inlet/outlet boundary conditions are not straightforward in Lagrangian methods, the upper and lower boundaries are here modeled as slip walls, i.e., with zero normal velocity, to prevent an in- or outflow at these two boundaries. The left and right boundaries are on the other hand inlet and outlet boundaries, respectively. A uniform velocity profile is imposed at the inlet, a constant pressure at the outlet and a no-slip condition at the cylinder surface. A description of the actual implementation of such boundary conditions can be found in Cerquaglia [75].

A relatively large computational domain is chosen, with $-4.5 \leq x \leq 75.5$ in the streamwise direction and $-15 \leq y \leq 15$ in the flow normal direction (length $L = 80D$ and width $W = 30D$), where the origin is located at the cylinder center. This allows minimizing the effect of boundary conditions and the blockage effect induced by the slip walls (the blockage is here $B = 1/30$). Furthermore, it provides the opportunity to clearly demonstrate the advantage of a non-uniform mesh, as relatively large elements can be used away from the cylinder and its wake.

Overall, the non-dimensional target mesh size ranges from $L^*_{\min} = 0.01$ to $L^*_{\max} = 1.1$, corresponding to a ratio of 110 between smallest and largest elements. Two geometric refinement criteria are combined. First, L^*_1 is increased linearly from $L^*_{\min}$ at the cylinder surface to $L^*_{\max}$ at a distance $d = 15$ from the cylinder center. The second geometric criterion imposes a target mesh size L^*_2 with an intermediate value L^*_{med} of 0.11 for $|y| \leq 2$ (i.e., in the wake region). For $|y| \geq 2$, L^*_2 increases linearly in y to $L^*_{\max}$ at the upper and lower boundary. Additionally, a solution-based criterion provides a target mesh size L^*_3 that varies between $L^*_{\min}$ where non-dimensional gradients are larger than 4 and 0.14 where gradients are smaller than 0.012 according to Eq. (3.23) with $\beta = 1/3$. This criterion is however only applied in the region $|y| \leq 2$ to ensure a good resolution of the Karman vortex street. Finally, the prescribed target mesh size is obtained as the minimum over the three criteria: $L^* = \min\{L^*_1, L^*_2, L^*_3\}$. Examples of meshes using only the two geometric or all three criteria are shown in Fig. 3.4 (Chapter 3). It is interesting to note that a uniform mesh with a mesh size of 0.01 everywhere would increase the number of elements by a factor between 10 and 60. At last, the non-dimensional time step is $\Delta t = L^*_{\min}$.

4.1.1 Static cylinder ($Re \leq 200$)

Many aspects of the flow around a cylinder have been investigated experimentally and numerically. For instance, the variation of the drag coefficient C_D at low Reynolds number, with respect to Re, has been studied numerically by Henderson [76] and Sheard et al. [77], and experimentally by Wieselberger [78]. For $Re > 40$, the variation of the average value of the root mean square of the lift coefficient, with respect to Re, has been studied numerically by Norberg [79]. Moreover, the variation of the Strouhal number St, with respect to Re, has been studied by Henderson [80], Williamson [81], Karniadakis and Keddar (private communication in [81]) and Zhang et al. [82]. As a reminder, the Strouhal number St is the non-dimensional value of the shedding frequency f : $St = f_D/U_\infty$. A last feature of interest is the length of the separation bubble for the stable wakes at $Re < 40$. It has been studied experimentally by Taneda [83], Grove et al. [84], Coutanceau and Bouard [85] and Acrivos et al. [86], and numerically by Sen et al. [87] and Keller and Takami [88], for different values of the blockage ratio $B = D/W$.

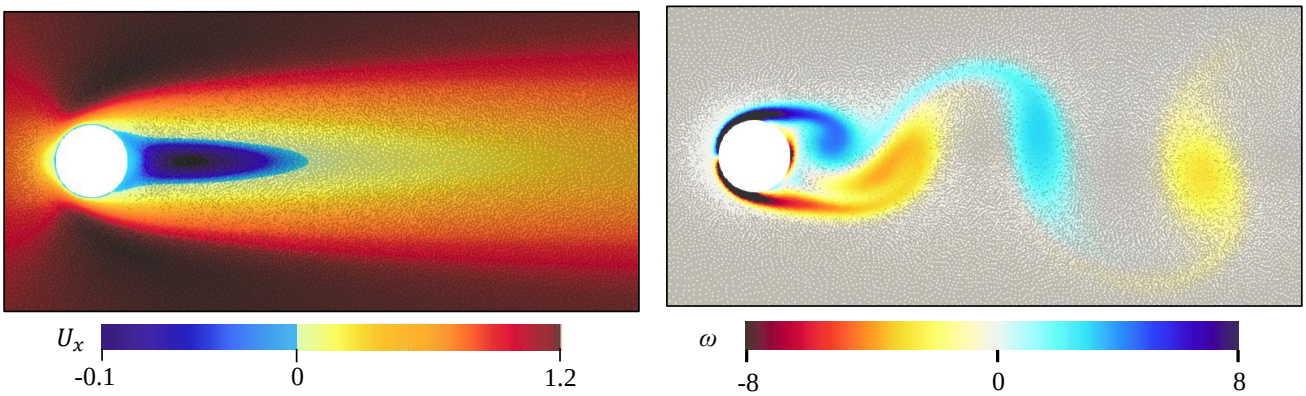
Thanks to the use of the new mesh adaptation techniques, as described above in the numerical setup, good results regarding the literature can be recovered. The aerodynamic coefficients are first analyzed. The steady or mean (for unsteady cases) drag coefficient C_D is compared for different values of Re_D ranging between 20 and 200 with references from the literature in Fig. 4.3. Because the steady or mean lift vanishes by symmetry, the root-mean-square (rms) of the lift coefficient is reported in Fig. 4.4 and compared with the numerical results of Norberg [79]. A very good match is observed in both cases between the present results and the reference data. This excellent agreement is further confirmed by the comparison of the recirculation bubble length L_{rec} for steady cases with similar blockage B in Fig. 4.5, and the Strouhal number for unsteady cases in Fig. 4.6.

The simulation at $Re_D = 200$ has also been repeated using only the geometric refinement criteria, L^*_1 and L^*_2. The same aerodynamic coefficients are found as for the simulation using the solution-based criterion, but a 5% lower shedding frequency is observed, owing to the coarser mesh in the wake.

4.1.2 Rotating cylinder ($Re_D = 200$)

To demonstrate the applicability of the method to moving bodies, the case of a two-dimensional laminar flow around a rotating cylinder at $Re_D = 200$ is briefly presented for two values of the cylinder rotation velocity. It is well-known that a sufficiently large rotation velocity Ω can suppress the vortex shedding process. More precisely, if the non-dimensional rotation velocity $\omega = \Omega D/(2U_\infty) \gtrapprox 2$, a steady but asymmetric solution is obtained [90, 91]. At lower values of ω, vortex shedding occurs but the wake is displaced vertically and the solution is also asymmetric. This test case is interesting because the position of the wake is not known a priori, such that a purely geometric refinement criterion might not provide an optimal refinement. The use of a solution-based refinement criterion overcomes this difficulty, as shown in two examples, one at a low ($\omega = 1.5$) and one at a high ($\omega = 2.07$) rotational speed.

The same computational domain and mesh characteristics are used as for the static case. The only difference is for the solution-based refinement criterion, which uses Eq. (3.23) with $\beta = 1$. Moreover, the target mesh size L^*_3 varies between $L^*_{\min}$ where relative gradients are larger than 0.8 and 0.2 where relative gradients are smaller than 0.05. In order to avoid too high distortion of the mesh elements above



(a) $Re_D = 40$ **(b)** $Re_D = 200$

Figure 4.1: Two-dimensional flow around a static cylinder. (**a**) Contour of the streamwise velocity component for the the steady case at $Re_D = 40$. The blue zone indicates the region where the streamwise velocity is negative and illustrates the extent of the recirculation bubble. (**b**) Contour of the vorticity for the unsteady case at $Re_D = 200$ showing the periodic Karman vortex street. In both cases, only a short portion of the entire computational domain is shown.

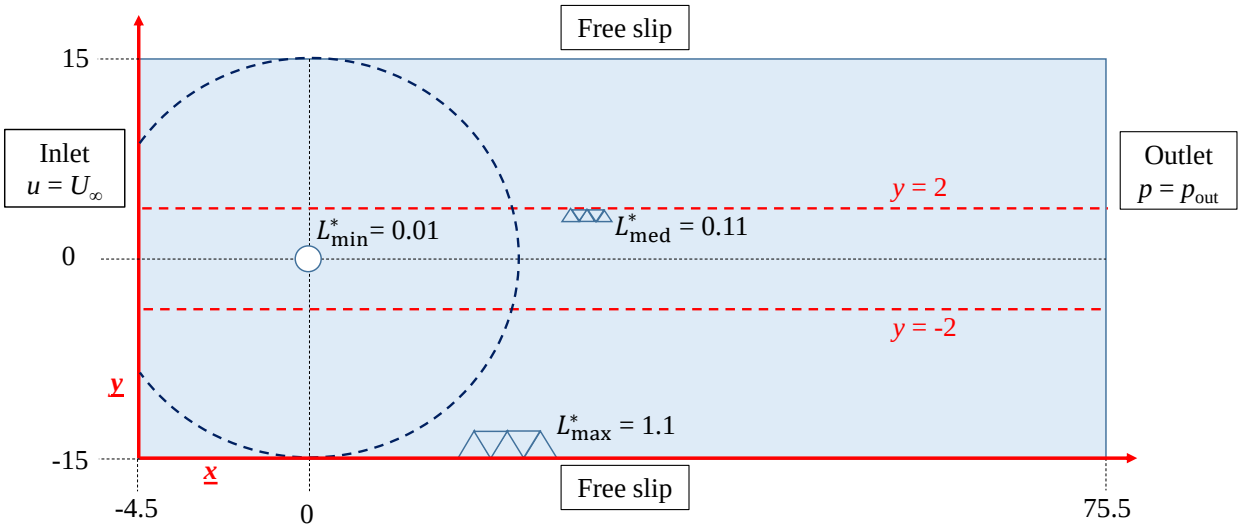


Figure 4.2: Numerical setup for the flow around a cylinder. The boundary conditions are indicated in the framed text near the four external boundaries. The dashed blue line is the limit of the first geometrical refinement criterion, increasing the value L_1^* from L^*_{min} at the cylinder surface to L^*_{max} at a distance of 15 from the cylinder center at $(0, 0)$. The dashed red lines are the limits for the second refinement criterion between which L_2^* is equal to the intermediate value L^*_{med}, while L_2^* increases from L^*_{med} to L^*_{max} outside of this intermediate region.

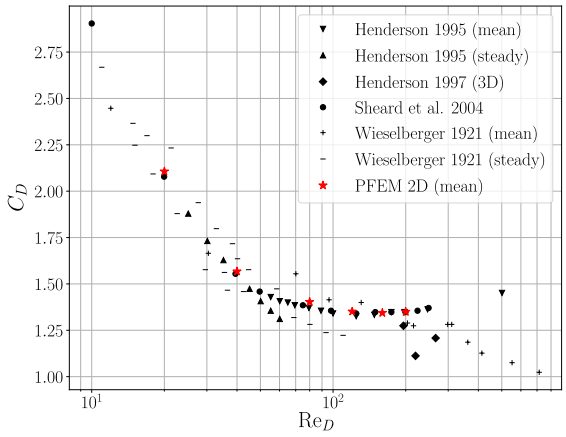


Figure 4.3: Steady or mean (if unsteady) drag coefficient as a function of the Reynolds number for the laminar flow around a static cylinder. The present results (red stars) are compared with the experimental results of Wieselberger [78] and the numerical results of Henderson [76, 89] and Sheard et al. [77].

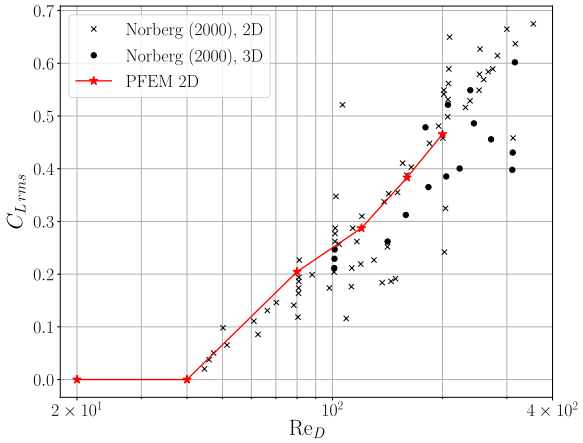


Figure 4.4: Root-mean-square (rms) lift coefficient as a function of the Reynolds number for the laminar flow around a static cylinder. The present results (red stars) are compared with the numerical results of Norberg [79]. The first two PFEM data points correspond to the steady cases without vortex shedding.

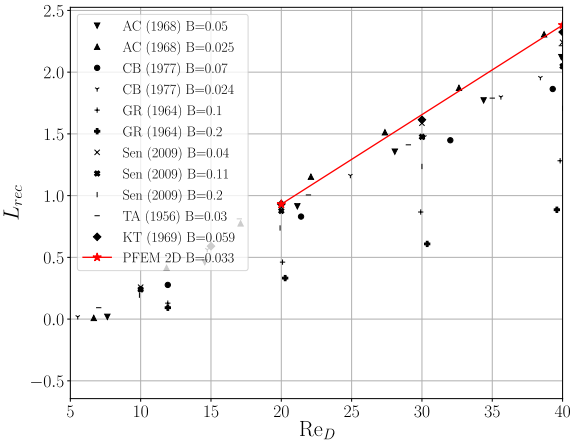


Figure 4.5: Non-dimensional length of the recirculation bubble behind the static cylinder for $Re_D \lessapprox 40$. The present results (red stars, $B = 0.033$) are compared with the experimental results of Taneda (TA) [83], Grove et al. (GR) [84], Coutanceau and Bouard (CB) [85] and Acrivos et al. (AC) [86], and to the numerical results of Sen et al. [87] and Keller and Takami (KT) [88], for different values of the blockage B.

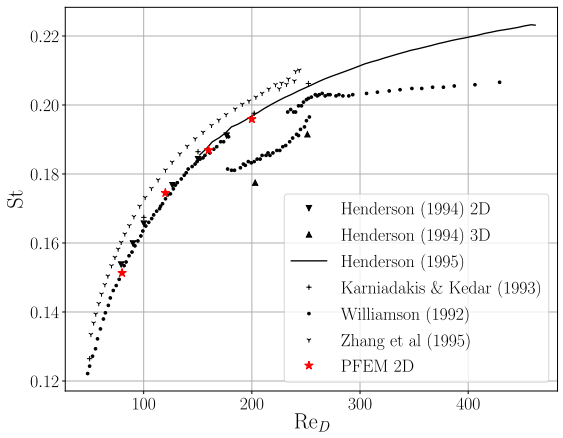


Figure 4.6: Strouhal number of the vortex shedding in the wake of a static cylinder for different Reynolds numbers. The present results (red stars) are compared with experimental results of Zhang et al. [82] and Williamson [81] and numerical results of Henderson (2D and 3D) [80] and Karniadakis and Kedar (taken from Williamson [81]).

the cylinder, where the flow velocity is increased by the cylinder rotation, the time step is reduced by a factor of 2.5 and 3, respectively, with respect to the static case.

The resulting vorticity field is shown in Fig. 4.7 for the two cases. As expected, a Karman vortex street is observed at low ω but displaced above the horizontal centerline. At larger ω vortex shedding is fully suppressed and the wake is asymmetric. Because of the flow asymmetry induced by the cylinder rotation, the lift does not vanish. The (mean) lift coefficient C_L obtained for the two cases is summarized and compared to numerical references from the literature in Table 4.1. Again good agreement is obtained.

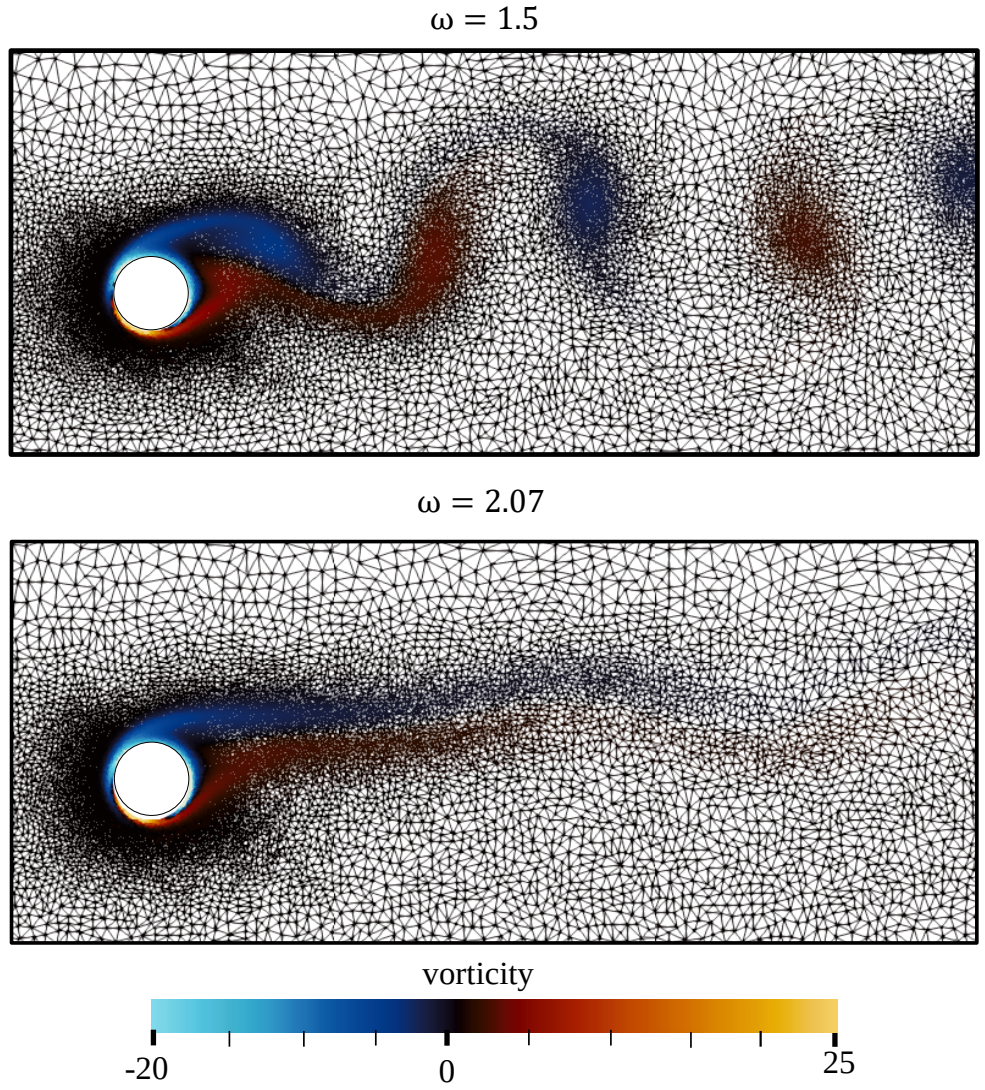


Figure 4.7: Vorticity contour for the laminar flow around a rotating cylinder at a non-dimensional rotation speed $\omega = 1.5$ (top) and $\omega = 2.07$ (bottom). The suppression of the shedding process is clearly visible at large ω. The wireframe representation illustrates the effect of the adaptive mesh refinement near the cylinder and around the vortices and the wake.

	$\omega = 1.50$	$\omega = 2.07$
Wang et al. [91]	-3.93	-5.71
Mittal et al. [90]	-3.90	-5.70
PFEM (GEO + SOL)	-3.94	-5.66

Table 4.1: Mean lift coefficient C_L of the rotating cylinder obtained for two different values of the non-dimensional cylinder rotation speed ω and comparison with numerical results from the literature. GEO and SOL indicate geometric and solution-based mesh refinement, respectively.

4.1.3 Transversely oscillating cylinder ($Re_D = 200$)

To further illustrate the use of the proposed mesh refinement algorithm for moving bodies, the two-dimensional laminar flow around a cylinder undergoing a forced vertical sinusoidal oscillation at $Re_D = 200$ is investigated. The objective is to reproduce the lock-in phenomenon that occurs when the cylinder oscillation has a sufficiently large amplitude and a frequency close to the natural vortex shedding frequency. During lock-in, the shedding frequency departs from the natural shedding frequency and synchronizes with the forcing frequency. Moreover, the characteristics of the wake are fundamentally altered (see e.g. Kumar et al. [92]).

Three cases taken from Wang et al. [91] are considered here: (a) one with sufficient forced oscillation amplitude A_e and adequate frequency to observe lock-in, (b) one with sufficient amplitude but a forcing

frequency f_e too different from the natural (i.e., for a static cylinder) shedding frequency f_S that lock-in should not take place, and finally (c) a case with an adequate forcing frequency but a low oscillation amplitude A_e preventing again the appearance of the lock-in phenomenon. The corresponding parameters are summarized in Table 4.2. The computational domain and mesh parameters are the same as for the static cylinder case.

Case	f_e/f_S	A_e	Lock-in
(a)	0.85	0.2	yes
(b)	0.75	0.2	no
(c)	0.85	0.1	no

Table 4.2: Parameters for the vertically oscillating cylinder, where f_e is the forcing frequency, f_S the natural shedding frequency and A_e the forcing amplitude. Only the first case should lead to the lock-in phenomenon.

The time evolution of the lift and drag coefficients is shown in the left column of Fig. 4.8 for the three cases. The vertical position $y_c(t)$ of the cylinder center is also reported for reference. Moreover, the power spectrum of the lift oscillation is also illustrated in the right column of that figure. Unlike the second and third cases (Fig. 4.8b,c) where several frequencies are clearly distinguishable, case (a) exhibits a unique frequency (Fig. 4.8a). The Fourier analysis shown in the right column confirms that this single frequency corresponds to the forcing frequency f_e. In other words, the shedding frequency has synchronized with the forcing frequency, there is a lock-in phenomenon. For cases (b) and (c), the lock-in does not occur and the shedding frequency remains distinct from the forcing frequency, leading to an amplitude modulation of the fluctuating aerodynamic coefficients. The Fourier analysis shows that the dominant frequency is f_S but contributions at f_e and other frequencies are also present. The frequency f_e is too apart from f_S (case (b)) or the oscillation amplitude A_e is too low (case (c)) to trigger the lock-in phenomenon, in agreement with Wang et al. [91] (see Figures 21 (b), 21 (a) and 24 (b) for the aerodynamics coefficients of the cases in table 4.2 and figures 23 (b), 23 (a) and 26 (b) for the power spectra of their lift coefficient). The corresponding wakes are shown in Fig. 4.9. A regular periodic vortex shedding is observed for the lock-in case (Fig. 4.9a), while clear variations are seen in the two other cases.

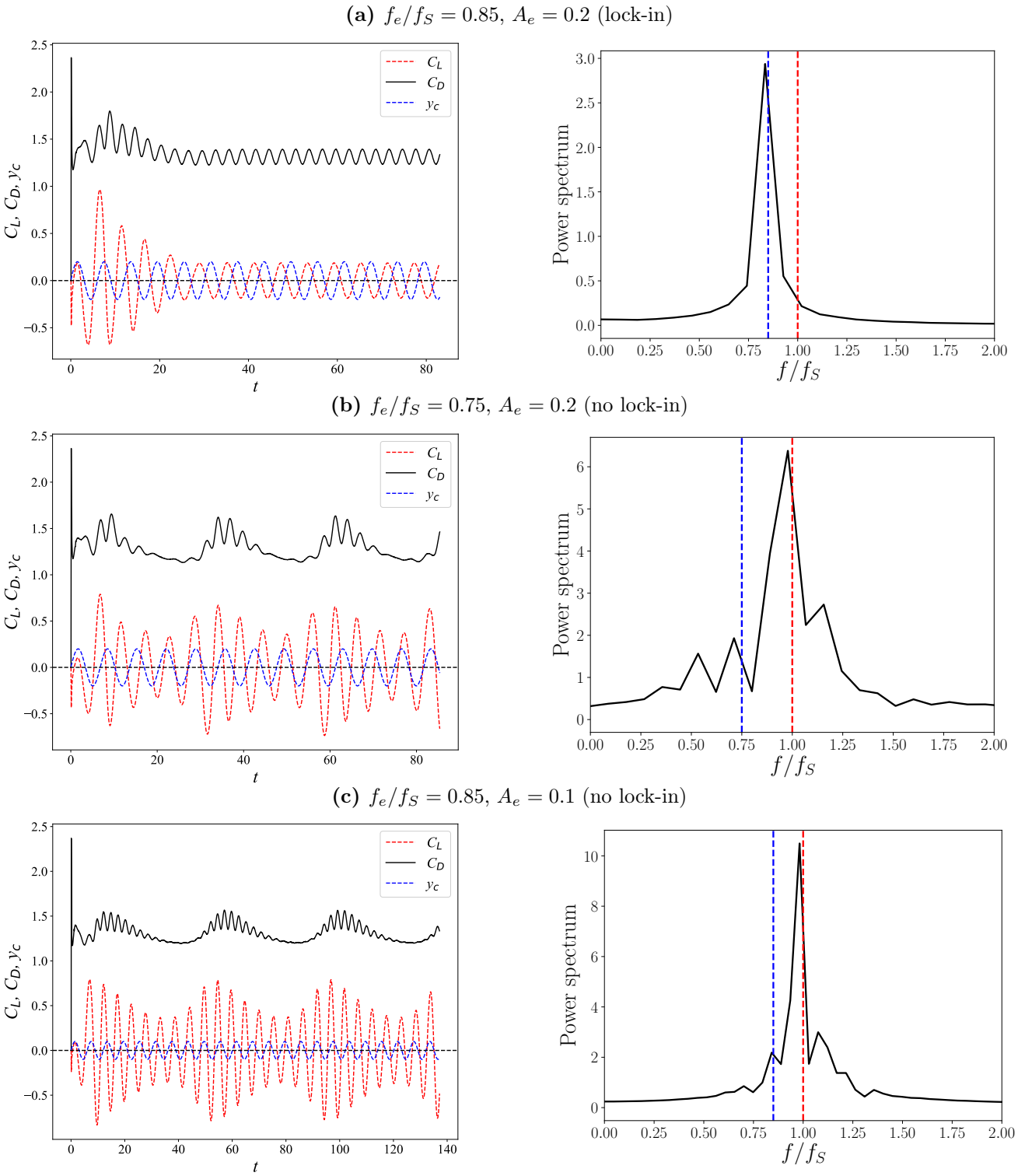
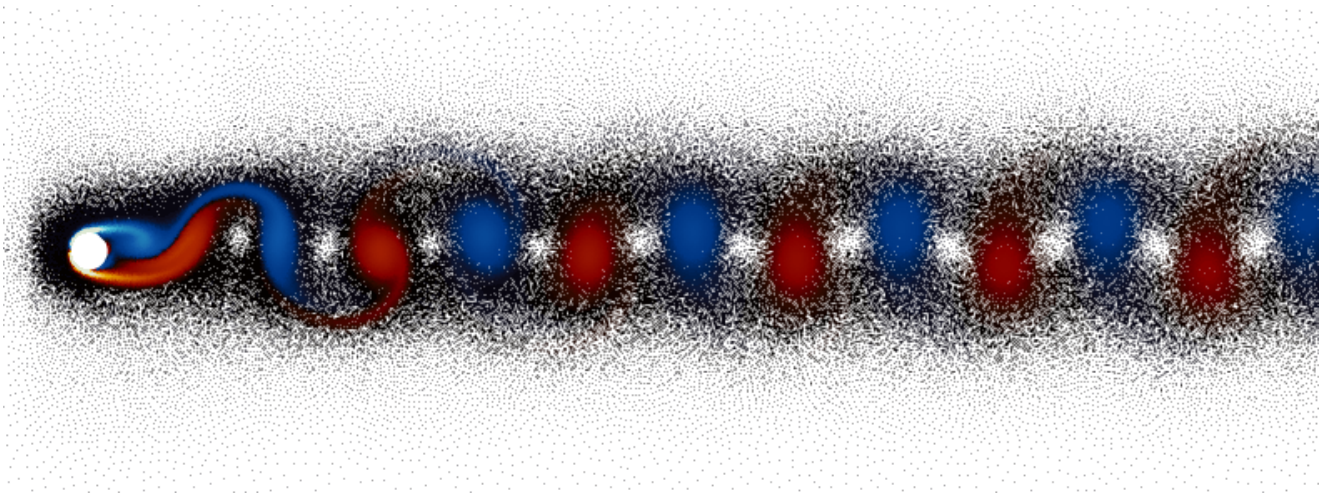
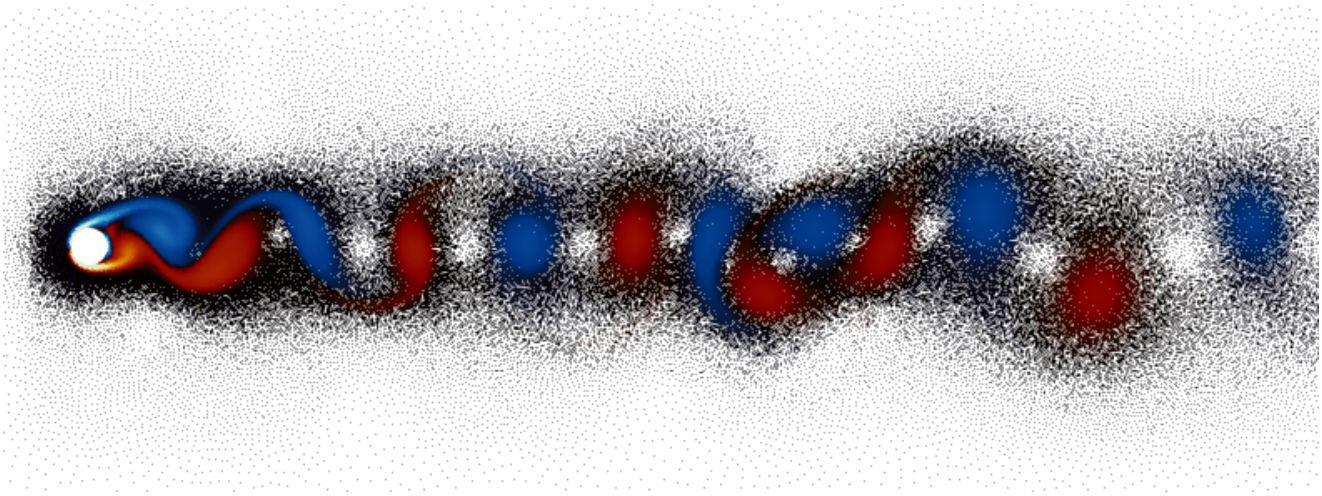


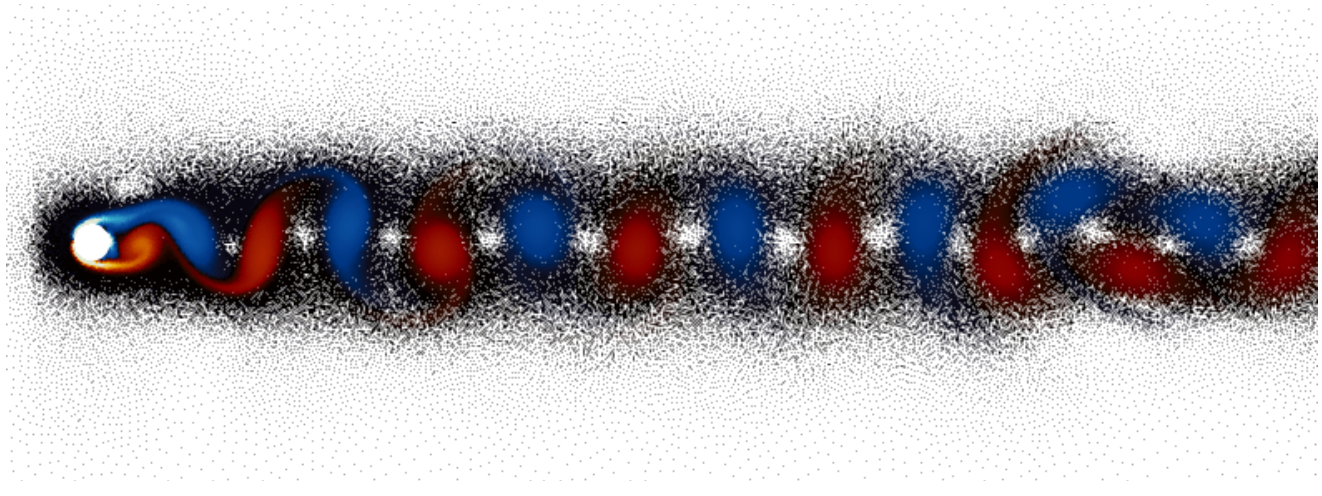
Figure 4.8: Left: lift and drag coefficients, C_L and C_D, and vertical position of the cylinder y_c as a function of the time for the flow around a vertically oscillating cylinder (to be compared with Figures 21 b), 21 a) and 24 b) of Wang et al. [91]). Right column: power spectra of the lift coefficient (to be compared with Figures 23 b), 23 a) and 26 b) of Wang et al. [91]). The dashed red line corresponds to the vortex shedding frequency f_S and the dashed blue line to the excitation frequency f_e, both being normalized by f_S. For each case, the corresponding oscillation amplitude A_e and the ratio of the forcing frequency f_e to the natural shedding frequency f_S is reported.



(a) $f_e/f_S = 0.85$, $A_e = 0.2$ (lock-in)



(b) $f_e/f_S = 0.75$, $A_e = 0.2$ (no lock-in)



(c) $f_e/f_S = 0.85$, $A_e = 0.1$ (no lock-in)

Figure 4.9: Vorticity contour in the wake of a vertically oscillating cylinder for different values of the cylinder oscillation frequency f_e and amplitude A_e. Only case **(a)** shows the lock-in phenomenon. The dots correspond to the mesh nodes.

4.1.4 Impulsively started cylinder ($Re_D = 9500$)

Previous cases only involved a limited Reynolds number ($Re_D \leq 200$) for which the size of the boundary layer was large enough that a limited resolution is sufficient to correctly capture the relevant physics. Therefore, they are not the most challenging ones in terms of required spatial resolution. The transient flow around an impulsively started cylinder at $Re_D = 9500$ is now considered. The goal is to evaluate different aspects of the mesh adaptation algorithm:

- The first objective is to compare two different solution-based strategies to define the target mesh size L^*, using either the velocity gradient $||\nabla \mathbf{U}||$ or its normalized version given by Eq. (3.24), together with Eq. (3.23), and determine how they are able to represent the interesting features of the transient wake. These different mesh refinement strategies are respectively referred to as SOL1 and SOL2, and their characteristics are summarized in Table 4.3.

- Secondly, it is shown that $||\nabla \mathbf{U}||$ or its normalized version given by Eq. (3.24) have both a dependency on some negative power of the distance r, and that a clever choice of the parameter β in Eq. (3.23) is able to provide a smooth progression of the target mesh size, therefore making the use of geometry-based mesh refinement unnecessary.

- The accuracy of the results obtained with the PFEM is evaluated by comparing them with those obtained by Koumoutsakos et al. [93] at the same Reynolds number and based on a vortex method. In particular, the accuracy of the solution is assessed by the vorticity contour at different times and by the time evolution of the drag coefficient.

Simulation setup

All quantities are here non-dimensionalized using the cylinder radius a, velocity U_∞ and density ρ. The main characteristics of the simulations, including domain size, boundary conditions and target mesh sizes, are summarized in Fig. 4.10a. The mesh after one iteration with the SOL2 mesh adaptation strategy (SOL1 leads to a very similar mesh) is shown in Fig. 4.10b. The parameters for mesh adaptation are summarized in Table 4.3. The simulation is performed in a frame of reference that is fixed with the cylinder and the flow goes from left to right. At $t = 0$, a uniform velocity U is prescribed everywhere (except at the cylinder surface). For $t > 0$, a uniform velocity is imposed at the inlet.

	SOL1	SOL2
L^*_{min}	0.00175	
L^*_{max}	1.4	
$\|\|\nabla \mathbf{u}\|\|_{\text{min}}$	0.0042	0.125
$\|\|\nabla \mathbf{u}\|\|_{\text{max}}$	10	5
β	1/3	1
$(\mathbf{U}_\infty, U_\epsilon)$	–	$((1,0)\,,\,0.01)$

Table 4.3: Target mesh size parameters of Eqs. (3.23) and (3.24) for the mesh refinement of the flow around an impulsively started cylinder at $Re_D = 9500$.

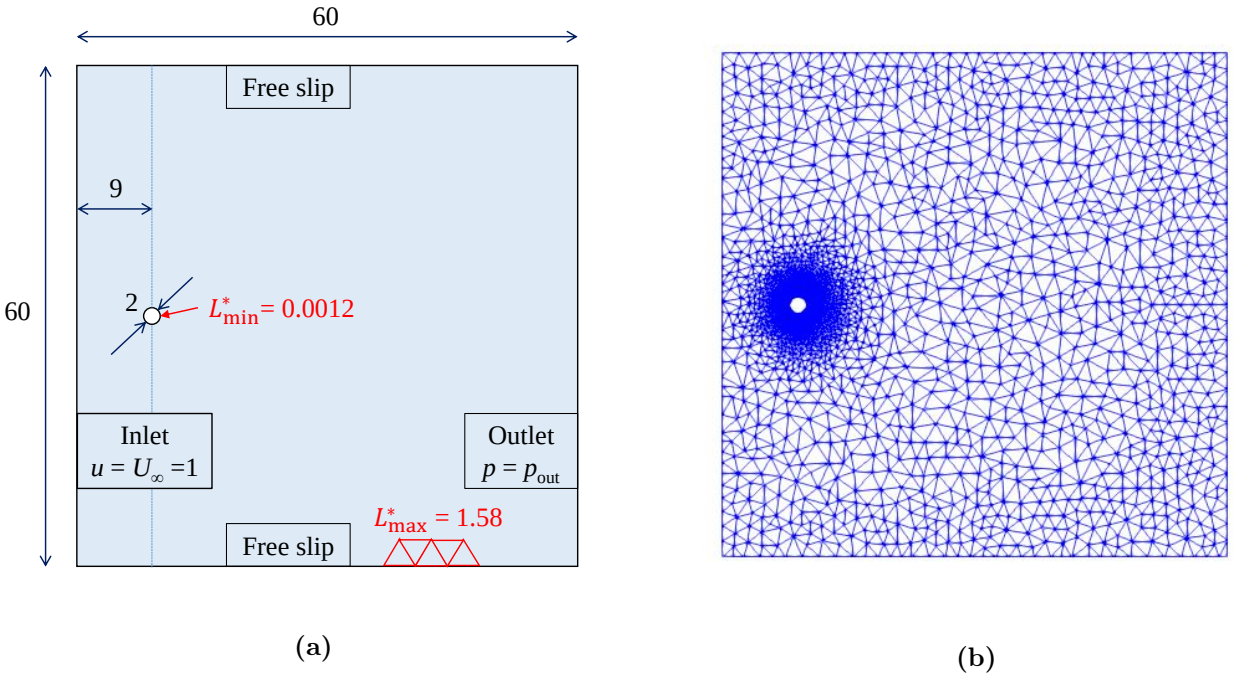


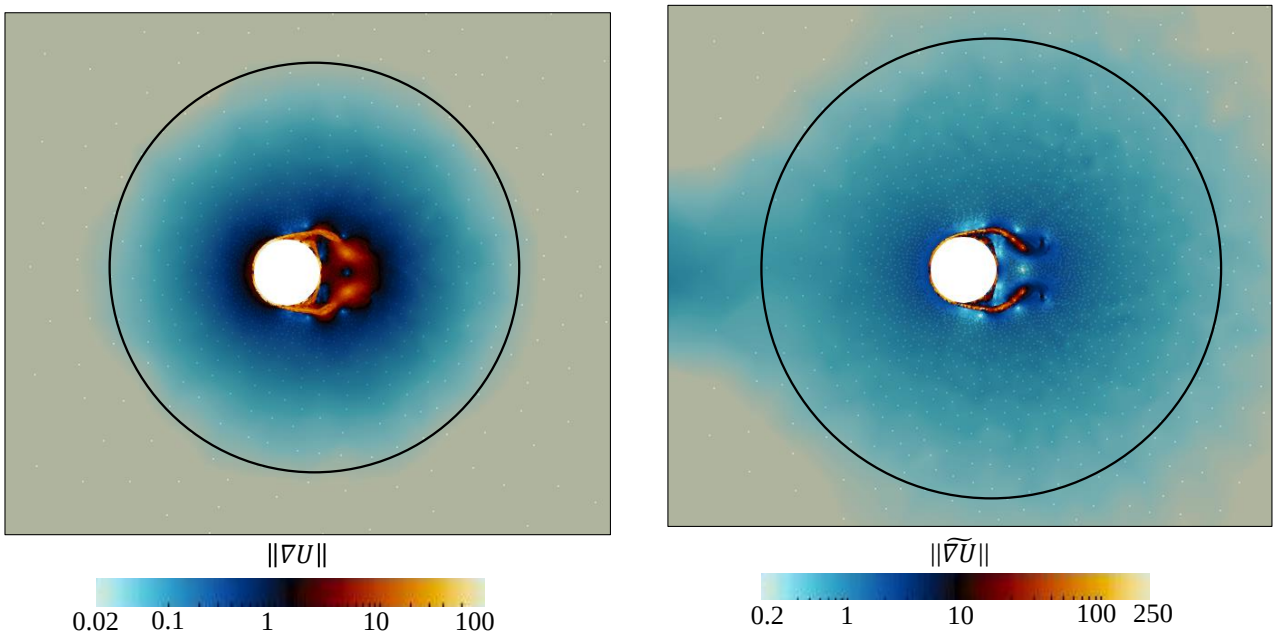
Figure 4.10: **(a)** Setup of the simulation for the flow around an impulsively started cylinder at $Re_D = 9500$. **(b)** Mesh wireframe after one iteration using the SOL2 mesh adaptation (note that the SOL1 gives a similar mesh).

Results

In a first step, the two mesh adaptation strategies are analyzed and compared. Figure 4.11 shows the contour plots of $||\nabla \mathbf{U}||$ and its normalized version (Eq. (3.24)) used in the solution-based mesh adaptation SOL1 and SOL2, respectively. At large distance, where the impact of the viscosity becomes small, the flow is close to a potential flow, so that $||\nabla \mathbf{U}||$ only depends on the distance from the cylinder center r (see Appendix A). As expected, the measure of the gradients, either in its absolute (Fig. 4.11a) or in its relative (Fig. 4.11b) form, confirms this, as the contour plot exhibits a clearly circular shape far from the wake. Therefore, no geometry-based criterion is needed to calculate L^*, the solution-based criteria providing by themselves a smooth increase of the target mesh size from the cylinder surface to larger distances from it. The advantage of both solution-based criteria over geometry-based criteria is that, in the present case, they provide detailed mesh adaptation in and around the wake and its different shearing layers.

Closer to the cylinder and in the wake, $||\nabla U||$ and its rescaled value strongly differ, as illustrated in Fig. (4.12). Consequently, the two mesh adaptation strategies, SOL1 and SOL2, lead to some differences in the mesh refinement. This can be visualized in Figs. 4.13 and 4.14 where the corresponding mesh wireframes colored by vorticity are depicted at four different times, from $t = 2.5$ to $t = 5.5$, so as to highlight the local mesh element size. For better visualization of the actual solution, contours of the vorticity without the mesh wireframe are also shown in Figs. 4.15 and 4.16 and should be compared to similar images in Fig. 26 of Koumoutsakos et al. [93], reproduced in Fig. 4.17. From these figures, it is clear that the SOL2 mesh refinement enables capturing features that are not captured by SOL1 because they are too weak. For instance, some thin vorticity structures seem to be cut in the lower left image of Fig. 4.13 and the lower and top left images of Fig. 4.14, which does not occur with the SOL2 mesh refinement (right column of these figures).

However, these differences are mostly an artefact of the wireframe representation, as demonstrated by Figs. 4.15 and 4.16. The vorticity fields associated with SOL1 and SOL2 are almost identical at early times and discrepancies become only apparent at later times. For instance, at $t = 4.5$, the vortex pair being ejected into the wake looks better defined with SOL2. At $t = 5.5$ larger differences are observed in the region around this ejected vortex pair, where the most downstream vortex filament is mostly absent with SOL1. Other small discrepancies are also visible in the vortex filaments that link



(a) $||\nabla \mathbf{U}||$ obtained with SOL1. **(b)** $\frac{||\nabla \mathbf{U}||}{(||\mathbf{U}-\mathbf{U}_\infty||+U_\epsilon)}$ obtained with SOL2.

Figure 4.11: Contour plot in a logarithmic scale of the Frobenius norm of the velocity gradient for the SOL1 mesh adaptation **(a)** and its normalized version for the SOL2 mesh adaptation **(b)**. The circular shape obtained far from the wake (approximated by the black circle) clearly shows that these two quantities depend on r (except for the small region between the inlet and cylinder in case of SOL2).

the ejected vortices to the cylinder. Comparison with the reference case (Fig. 4.17) indicates that SOL2 performs slightly better. While all the key structures are captured by the PFEM, differences with the results of Koumoutsakos et al. [93] are nonetheless visible. First, the ejected vortex pair at $t = 5.5$ in the reference solution is fully disconnected from the cylinder, while these two vortices seem to still be partly connected through a vortex filament to the cylinder in the PFEM solutions. Furthermore, the vortex filament wrapping around the ejected vortices remain well distinct in the reference solution but seems to have mostly merged with the core of the ejected vortices in the present results. It is also interesting to note that the solutions remains well symmetric, although the mesh is not.

The drag coefficient obtained with SOL1 and SOL2 is compared to the reference case of Koumoutsakos et al. in Fig. 4.18. The two mesh refinement strategies give very similar results, except for a small time shift between both. Therefore, despite the differences observed in the mesh resolution, the mesh refinement performs very similarly near the cylinder surface, resulting approximately in the same drag coefficient. Finally, the drag coefficients obtained with the PFEM match qualitatively the reference case but quantitative differences are observed, in particular at later times. Specifically, after $t = 3$, the curve of Koumoustakos et al. shows a time lag with respect to that of PFEM (SOL1 and SOL2), and reaches a higher maximum peak around $t = 3.8$ and a lower minimum peak around $t \simeq 4.5$. The lower extrema in the PFEM results are most likely due to numerical dissipation associated with the finite mesh size and the resulting diffusion of vortical structures, as previously discussed. Further refining the mesh would improve the results but at a higher computational cost. Overall, the result are nonetheless encouraging, especially in view of the large Reynolds number. Furthermore, this case would be much more expensive to simulate with a uniform mesh and the added value of mesh adaptation is evident. Finally, the SOL2 mesh adaptation strategy is for this case better adapted than SOL1.

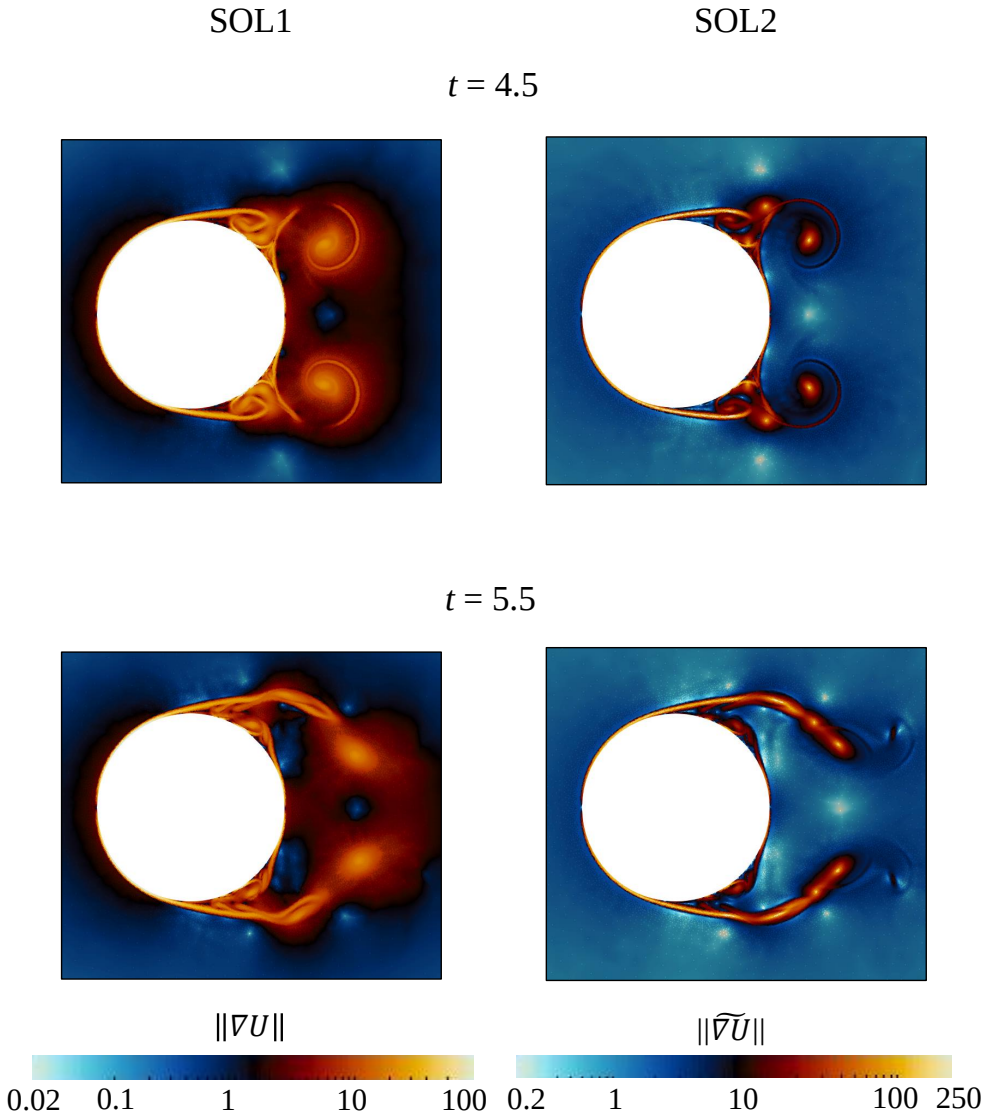


Figure 4.12: Contours of the Froebenius norm of the velocity gradients and normalized velocity gradients at times t=4.5 and t=5.5 for the SOL1 (left) and the SOL2 (right) mesh adaptations.

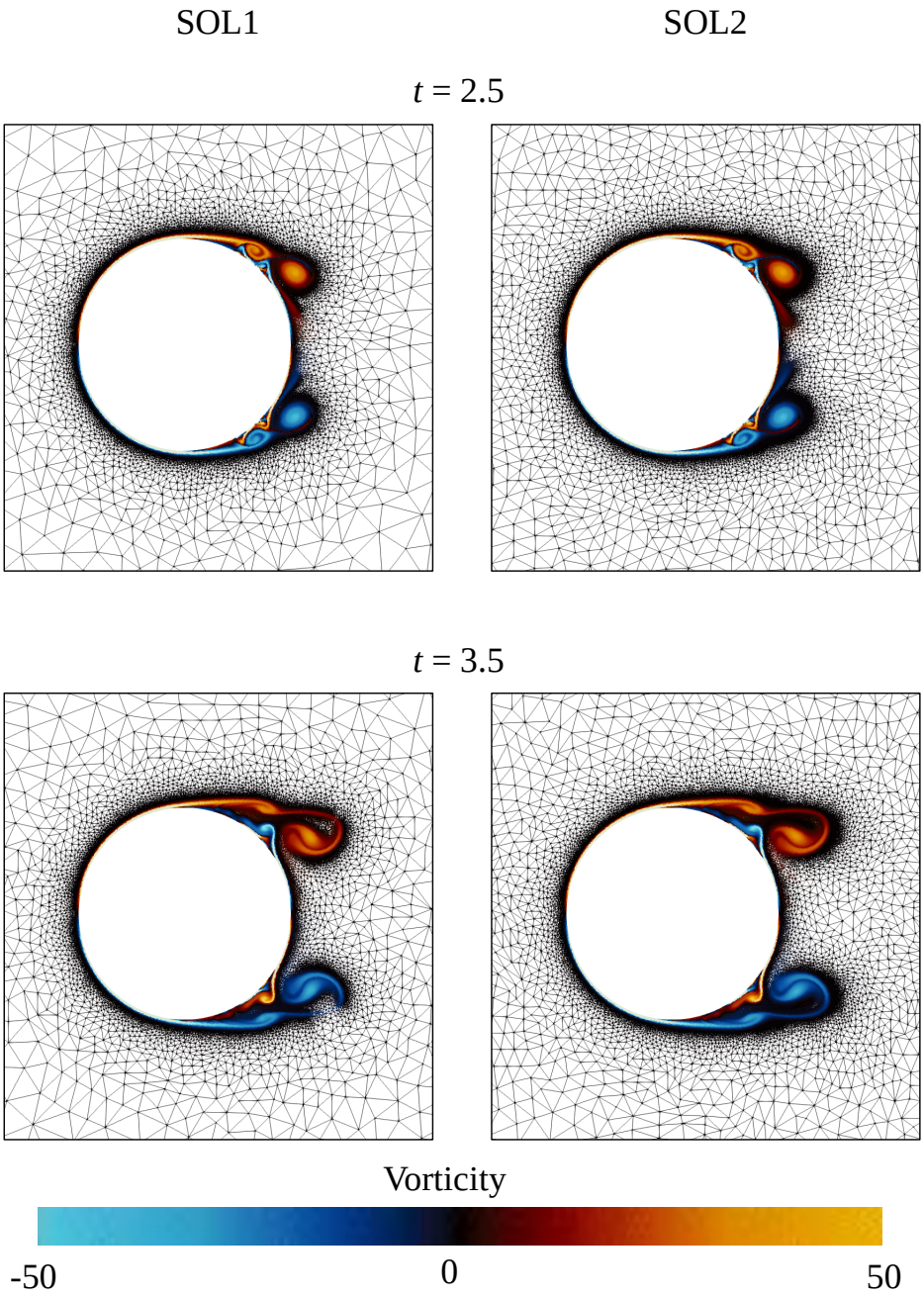


Figure 4.13: Mesh wireframe colored by the vorticity at times t=2.5 (top) and t=3.5 (bottom) for the SOL1 (left) and SOL2 mesh adaptation (right).

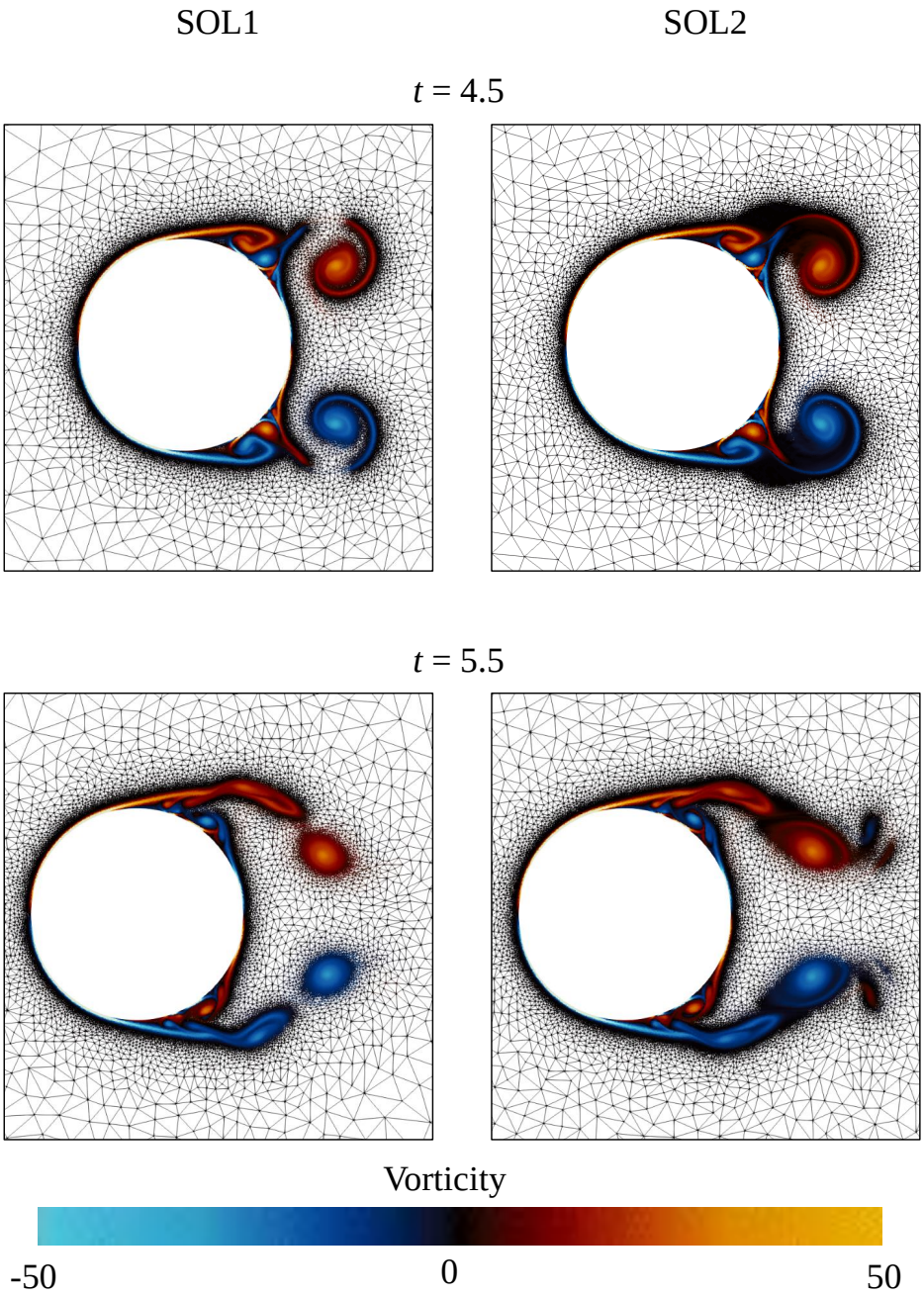


Figure 4.14: Mesh wireframe colored by the vorticity at times t=4.5 (top) and t=5.5 (bottom) for the SOL1 (left) and SOL2 mesh adaptation (right).

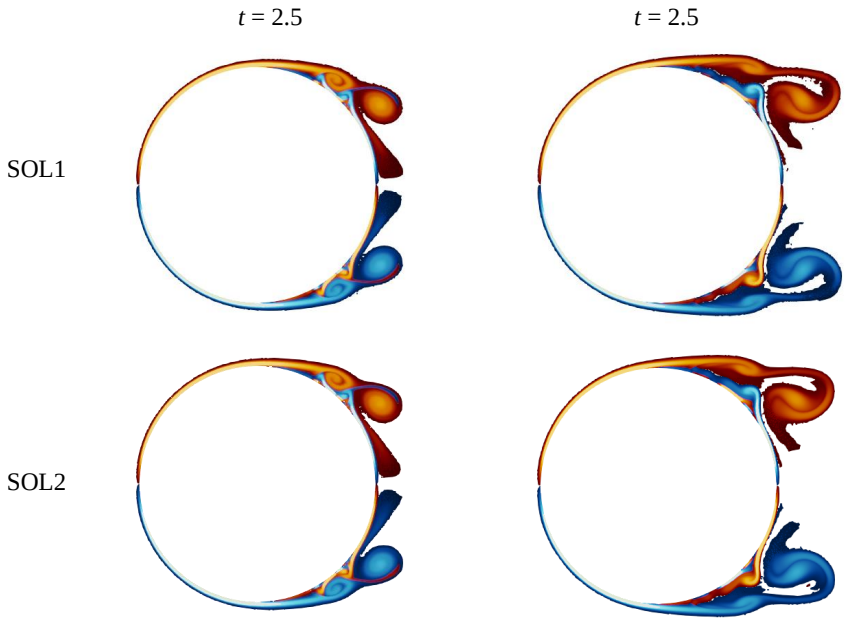


Figure 4.15: Vorticity contour obtained for an impulsively started cylinder at $Re_D = 9500$, at times $t = 2.5$ (left) and $t = 3.5$ (right) with the PFEM, using the SOL1 (top) and SOL2 (bottom) mesh adaptation.

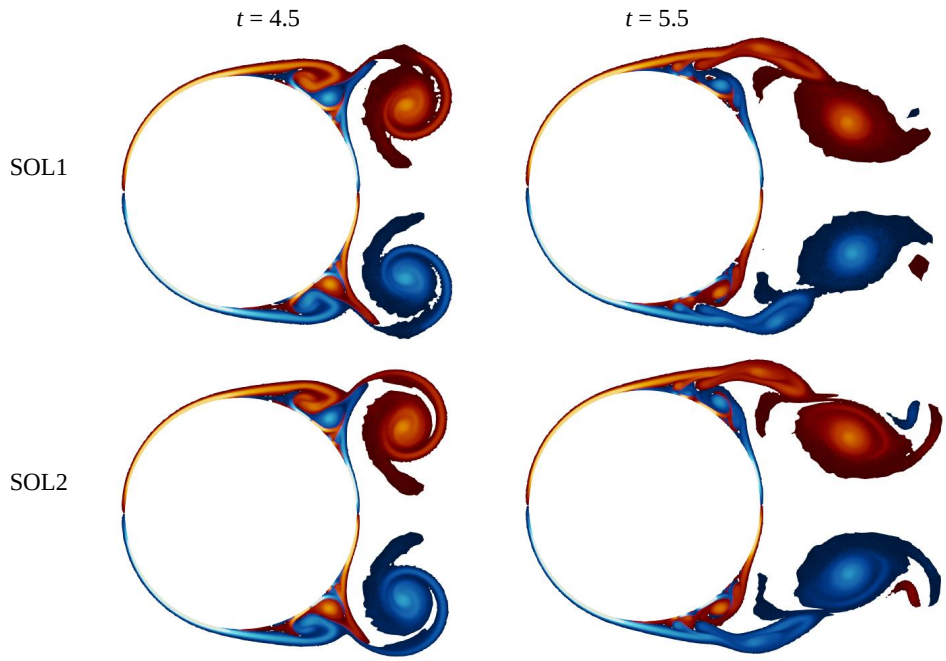


Figure 4.16: Vorticity contour obtained for an impulsively started cylinder at $Re_D = 9500$, at times $t = 4.5$ (left) and $t = 5.5$ (right) with the PFEM, using the SOL1 (top) and SOL2 (bottom) mesh adaptation.

Figure 4.17: Vorticity contour obtained at different times $t \in \{2.5, 3.5, 4.5, 5.5\}$ by Koumoutsakos et al. [93] for an impulsively started cylinder at $Re_D = 9500$. These results are taken as reference for the comparison with the present results (see Figs.4.15 and 4.16).

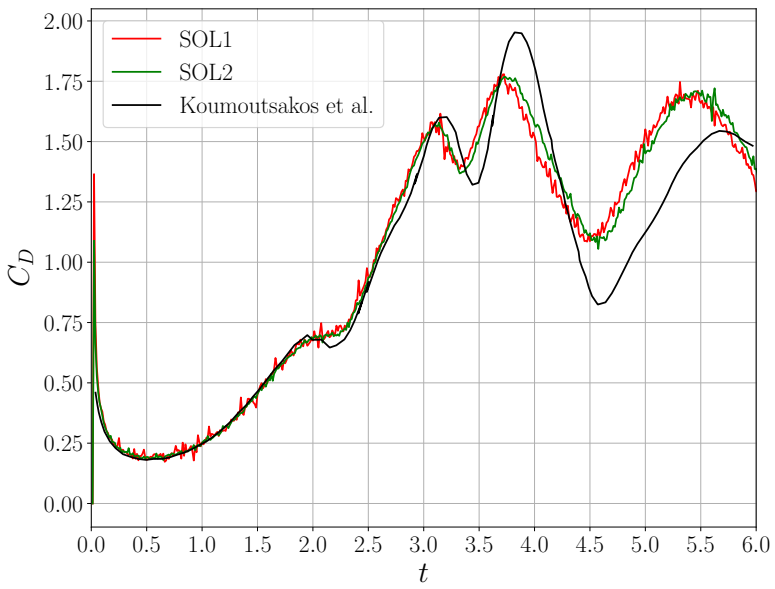
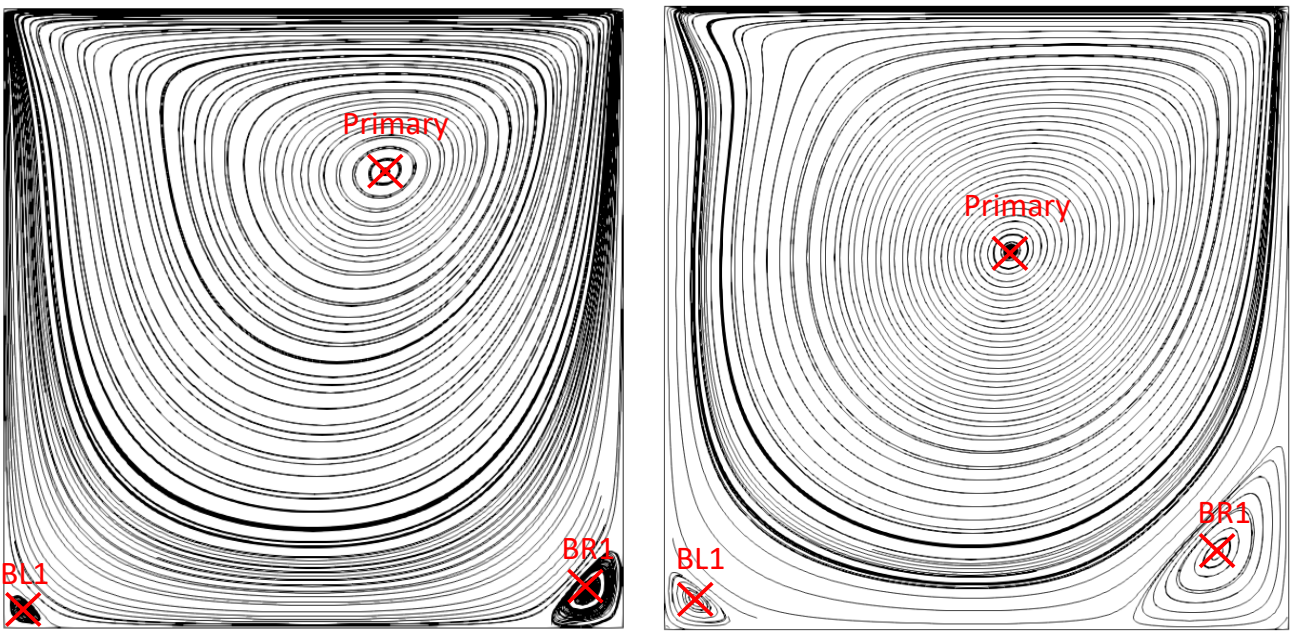


Figure 4.18: Drag coefficient a function of the time for an impulsively started cylinder at $Re_D = 9500$, obtained with the two different mesh refinement criteria SOL1 and SOL2, and reference results of Koumoutsakos et al. [93]

4.2 The lid-driven cavity

A square cavity is now considered, whose bottom and side walls are fixed and whose top wall (lid) moves from left to right at constant velocity. After a transient phase, the velocity field reaches a steady state characterized by a primary rotational structure and smaller vortices in the lower corners of the cavity, as illustrated in Fig. 4.19. All reported quantities are made non-dimensional using the fluid density ρ, the cavity side length L and the lid velocity U. Reynolds numbers of 100 and 400 are considered, with $Re = \rho UL/\mu$, where μ is the dynamic viscosity. First, a mesh convergence analysis is performed using a uniform mesh and results are compared with the literature for the case at $Re = 400$, because it is more limiting than the case at $Re = 100$ in terms of required spatial resolution (see Section 3.3). Then, still for $Re = 400$, several strategies to define the target mesh size are analyzed. Afterwards, results obtained with uniform and non-uniform meshes are compared. These results are mainly assessed in terms of the position of the primary and secondary vortices and the vorticity at these locations. Finally, the results with the most resolved uniform and non-uniform meshes are shown for the case $Re = 100$, and compared with the literature. The differences between the PFEM solutions at the two different Reynolds numbers are also briefly discussed.



(a) **(b)**

Figure 4.19: Streamlines at steady state obtained for the lid-driven cavity flow. The location of the three main vortices is indicated by the red crosses. (**a**) Re= 100 on a non-uniform mesh with a minimal mesh element size $L^*_{\text{min}} = 0.00375$ and a maximum mesh element size $L^*_{\text{max}} = 0.015$. (**b**) Re= 400 on a 80×80 uniform mesh, corresponding to an average element size $L^* = 0.0075$.

4.2.1 Main flow features and mesh convergence analysis

First, the convergence of the results at $Re = 400$ on a uniform mesh[1] is analyzed using a series of refined meshes with $N \times N$ subdivisions along the vertical and horizontal walls, ranging from 20×20 to 160×160. The corresponding total number of nodes is more or less $4N^2/3$, as triangular elements are used. For comparison, this would correspond to $L^*_{\text{min}} = L^*_{\text{max}} \approx 0.6/N$. The time step size is defined based on the minimum target mesh size as $\Delta t = 5L^*_{\text{min}}/12$ for this mesh convergence study and all subsequent simulations. The results obtained with these meshes are also compared to several references from the literature [94–97].

Figure 4.19b shows the streamlines of the solution obtained with the 80×80 mesh. As summarized in Table 4.4, it is found that the mesh with a resolution of 80×80 is sufficient to get results close to those of the literature. The most resolved mesh (160×160) does not lead to results that are significantly

[1]The mesh is not perfectly uniform since it deforms at each time step, but it has approximately a uniform element size.

		PFEM simulations				Ghia et al.	Schreiber et al.	Vanka	Hou et al.
Res. $\rightarrow$		20×20	40×40	80×80	160×160	257×257	180×180	321×321	256×256
Num. meth.		PFEM				CSI-MG	"homemade"	BLIMM	LBM
Form. var.		u, v, p				$\psi - \omega$	ψ	u, v, p	BGK model
Pr.	x_c	0.5522	0.5527	0.5562	0.5543	0.5547	0.5571	0.5563	0.5608
	y_c	0.6039	0.6080	0.6074	0.6059	0.6055	0.6071	0.6	0.6078
	ω	2.446	2.3155	2.292	2.289	2.2947	2.281	–	–
BR1	x_c	0.8910	0.8857	0.8856	0.8854	0.8906	0.8857	0.8875	0.8902
	y_c	0.1173	0.1213	0.1223	0.1224	0.1250	0.1143	0.1188	0.1255
	ω	0.3749	0.4007	0.4104	0.4161	0.433 52	0.394	–	–
BL1	x_c	–	0.0493	0.05	0.050 83	0.0508	0.05	0.05	0.0549
	y_c	–	0.0457	0.0467	0.047 07	0.0469	0.0429	0.05	0.0510
	ω	–	0.0675	0.044 57	0.060 37	0.056 97	0.0471	–	–

Table 4.4: Mesh convergence study on a uniform mesh for the lid-driven cavity test case at $Re = 400$: position (x_c, y_c) and associated vorticity ω of the primary and secondary (BR1 and BL1, see Fig. 4.19 for definition) vortices for different mesh sizes and comparison with the results of Ghia et al. (finite difference method, vorticity-streamfunction formulation) [94], Schreiber et al. (fourth-order finite difference method, vorticity-streamfunction formulation) [96], Vanka (finite difference method on staggered grid) [97] and Hou et al. (Lattice Boltzmann method) [95].

different (the difference with respect to the coarser mesh is of the same order of magnitude as the variability across the reference data), but increases noticeably the computational time. Because the intermediate 80×80 mesh provides sufficiently accurate results while keeping the computational cost limited, all following analyzes are based on meshes of a similar size.

4.2.2 Different mesh refinement strategies and comparison with uniform meshes

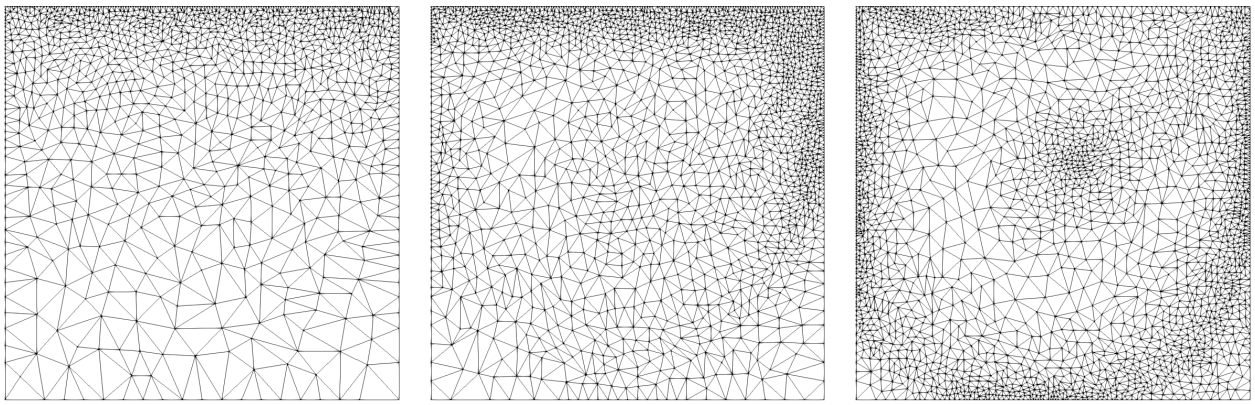
As discussed in Section 3.2.1, several criteria can be used to define the target mesh size. For this test case, a geometric and two physics-based criteria are investigated. The geometric criterion (denoted GEO) defines a target mesh size that increases linearly from L^*_{min} at the top wall to L^*_{max} at the bottom wall. The finer mesh at the lid is motivated by the larger velocity gradients and the thinner boundary layer in this region, as indicated by the closely packed streamlines in Fig. 4.19. Moreover, a finer mesh at the cavity upper corners reduces the effect of the velocity discontinuity induced by the moving lid.

It can be expected that this geometric criterion is not sufficient to accurately capture the vortical structures. An additional refinement based on the velocity gradients using Eq. (3.23) (denoted SOL1) is thus also considered. Nonetheless, because the corner vortices are much weaker than the primary vortical structure, their associated velocity gradients are much smaller, which leads to a rather large target mesh size there. A better targeted refinement of the corner vortices can be achieved with a solution-based criterion (denoted SOL2) that uses rescaled velocity gradients according to Eq. (3.24). The mesh size is thus prescribed based on the shape of the vortices rather than on their intensity. The different target mesh size parameters for the three criteria are summarized in Table 4.5. The mesh resulting from these three criteria are compared for $L^*_{\text{min}} = 0.6/80 = 0.0075$ and $L^*_{\text{max}} = 8L^*_{\text{min}}$.

	GEO	SOL1	SOL2
$\|\|\nabla \mathbf{u}\|\|_{\text{min}}$	–	2/9	2
$\|\|\nabla \mathbf{u}\|\|_{\text{max}}$	–	6	50
β	–	1/3	1
$(\mathbf{U}_\infty, U_\epsilon)$	–	–	$((0,0)\,,\, 0.01)$

Table 4.5: Target mesh size parameters of Eqs. (3.23) and (3.24) for the mesh refinement of the lid-driven cavity test case at $Re = 400$.

Typical meshes obtained at steady state using the above mesh refinement criteria are illustrated in Fig. 4.20. The GEO criterion produces a mesh with progressively increasing size in the direction



(a) GEO: $N_{\text{st}} = 1184$ **(b)** SOL1: $N_{\text{st}} = 3566$ **(c)** SOL2: $N_{\text{st}} = 2469$

Figure 4.20: Meshes at steady state obtained using three different mesh refinement strategies for the lid driven cavity flow at $Re = 400$. For all three cases, $L^*_{\text{min}} = 0.6/80 = 0.0075$ and $L^*_{\text{max}} = 8L^*_{\text{min}}$. The approximate total number of nodes at steady state, N_{st}, is also reported. The other target mesh size parameters are summarized in Table 4.5.

normal to the lid (Fig. 4.20a), while the SOL1 criterion leads to a mesh with the same background mesh stretching but with additional refinement at the edge of the primary vortex, in particular just below the top right corner (Fig. 4.20b). However, the resulting mesh at the two bottom corners is very coarse, preventing an accurate representation of the corner vortices there. The SOL2 criterion, on the other hand, yields a fine mesh along the entire edge of the primary vortex and at its core (Fig. 4.20c). Additionally, a better refinement at the bottom corners is also achieved.

Uniform vs. non-uniform meshes

One of the main motivation for the present mesh refinement algorithm is to be able to refine locally the mesh in order to improve the solution accuracy in specific regions while keeping the computational cost limited. Alternatively, this also provides the ability to use a coarser mesh in regions where the solution is smooth, thus reducing the computational cost while keeping the same accuracy of the solution. However, it should be emphasized that even a very local and spatially limited mesh refinement has an impact on the time step. More specifically, a reduction of the element size imposes a corresponding time step reduction. It is important to note that, within a Lagrangian formalism, the bound on the time step size is not directly related to stability but is a problem of convergence of the nonlinear algorithm. A too large mesh deformation within one time step makes the convergence very difficult, or even impossible. As a consequence, even when the decrease of L^*_{min} induces only a small increase in the total number of elements, a non-negligible increase in the overall computational cost due to a larger number of time steps is unavoidable.

Three non-uniform meshes based on the SOL2 refinement with the same L^*_{min} (corresponding to that of a 80×80 uniform mesh) and a decreasing L^*_{max} are compared to uniform meshes in Table 4.6. One can observe that increasing L^*_{max} deteriorates slightly the accuracy of the simulation, but also significantly reduces the number of nodes. For instance, Mesh 2 in Table 4.6 has half as many nodes as the 80×80 uniform mesh for virtually the same accuracy. Comparing Mesh 1 and the 40×40 uniform mesh that both have a similar number of nodes, one observe that, although the position of the primary vortex is surprisingly slightly less accurate for Mesh 1, the corresponding vorticity is much better predicted.

Comparison between $Re = 100$ and $Re = 400$

The flow physics is further analyzed by comparing the two Reynolds number cases. First, the results obtained at $Re = 100$ are summarized for the most resolved uniform and non-uniform (SOL2) meshes (among those used for simulations at that Reynolds number) in Table 4.7. Note that the resolution of these meshes is expectedly slightly lower than for $Re = 400$. Globally, the accuracy of the vortex

Mesh type →			Similar $L^*_{\min}$ Mesh 1	Similar $L^*_{\min}$ Mesh 2	Similar $L^*_{\min}$ Mesh 3	Uniform (40×40)	Uniform (80×80)	Uniform (160×160)
Parameters	$L^*_{\min}$		0.0075	0.0075	0.0075	0.015	0.0075	0.003 75
	$L^*_{\max}/L^*_{\min}$		8	4	2	1	1	1
	$\Delta t \times 10^3$		3.125	3.125	3.125	6.25	3.125	1.5625
	N_{init}		~ 500	~ 1200	~ 2900	~ 2200	~ 8500	~ 34000
	N_{steady}		~ 3300	~ 5200	~ 7900	~ 2700	~ 10500	~ 42000
Results	Primary	x_c	0.5594	0.5563	0.5551	0.5527	0.5562	0.5543
		y_c	0.6106	0.6083	0.6072	0.6080	0.6074	0.6059
		ω	2.281	2.2895	2.292	2.3155	2.292	2.289

Table 4.6: Comparison of different mesh adaptation strategies for the lid-driven cavity flow at $Re = 400$: SOL2 mesh refinement for three meshes with same $L^*_{\min}$ and decreasing $L^*_{\max}$ and three uniform meshes with decreasing $L^*_{\min}$. Mesh parameters, time step, initial and final number of nodes and position (x_c, y_c) and associated vorticity ω of the primary vortex.

Res. →		Uniform mesh 100×100	Non-Uniform mesh $L^*_{\min} = 0.00375$	Ghia et al. 257×257	Schreiber et al. 180×180	Vanka 321×321	Wright et al. 1024×1024
Num. meth.		PFEM		CSI-MG	"homemade"	BLIMM	FV
Form. var.		u, v, p		$\psi - \omega$	ψ	u, v, p	u, v, p
Pr.	x_c	0.6158	0.6156	0.6172	0.6167	0.6188	0.6157
	y_c	0.7376	0.7377	0.7344	0.7417	0.7375	0.7378
	ω	3.1852	3.1716	3.1665	3.182	–	3.1704
BR1	x_c	0.9428	0.9439	0.9453	0.9417	0.9375	0.9419
	y_c	0.0613	0.0604	0.0625	0.05	0.0563	0.062
	ω	0.0383	0.0379	0.0331	0.0255	–	0.0351
BL1	x_c	0.0343	0.0337	0.0313	0.0333	0.0375	0.0337
	y_c	0.0341	0.0319	0.0391	0.025	0.0313	0.0347
	ω	0.0162	0.0105	0.0156	0.008	–	0.0146

Table 4.7: Position and corresponding vorticity of the three main corner vortices for the lid-driven cavity flow at $Re = 100$. The PFEM simulations obtained with a uniform mesh are compared with those obtained with a non-uniform mesh with a progression $L^*_{\max}/L^*_{\min} = 4$, as well as with results from the literature: Ghia et al. (finite difference method, vorticity-streamfunction formulation) [94], Schreiber et al. (fourth-order finite difference method, vorticity-streamfunction formulation) [96], Vanka (finite difference method on staggered grid) [97] and Wright et al. (finite difference method with multigrids) [98].

position and associated vorticity is satisfactory, in the sense that the variability between the results obtained with the uniform and non-uniform meshes is smaller that that observed in the results from the literature. As already mentioned, results should be interpreted keeping in mind that they can significantly vary depending on whether small corner vortices are captured or not.

Table 4.8 compares the results on the best uniform and non-uniform (SOL2) meshes for the two Reynolds numbers. The changes in the solution when increasing the Reynolds number are illustrated by Fig. 4.19. It can be observed that, with an increase of the Reynolds number,

- the vorticity at the center of the primary vortex becomes weaker (from 3.18 to 2.29) and the vortex center moves to a lower position, closer to the cavity center;
- the two secondary vortices become larger, their center moves further away from the cavity corners, and the associated vorticity increases.

These overall changes result from the larger relative contribution of inertia when Re is increased. As expected, the secondary vortices become more prominent, thus modifying the primary vortex.

Finally, it should be emphasized that these results, and in particular the position of the different vortices, are very sensitive to many factors. This is clearly demonstrated by the variability of the results

		Re =100		Re =400	
Mesh type →		Uniform	Non-uniform	Uniform	Non-uniform
Resolution →		100×100	$L^*_{\min} = 0.00375$	160×160	$L^*_{\min} = 0.0025$
Pr.	x_c	0.6158	0.6156	0.5543	0.5552
	y_c	0.7376	0.7377	0.6059	0.6067
	ω	3.1852	3.1716	2.289	2.2945
BR1	x_c	0.9428	0.9439	0.8854	0.8856
	y_c	0.0613	0.0604	0.1224	0.1225
	ω	0.0383	0.0379	0.4161	0.4566
BL1	x_c	0.0343	0.0337	0.050 83	0.4950
	y_c	0.0341	0.0319	0.0471	0.0469
	ω	0.0162	0.0105	0.0604	0.0529

Table 4.8: Comparison of the position and associated vorticity of the three main vortices in the lid-driven cavity flow for the two Reynolds numbers considered on both uniform and non-uniform (SOL2) meshes.

in the literature (see Tables 4.4 and 4.7) despite very fine meshes. This sensitivity is most likely due to the presence of additional smaller and smaller vortices that emerge in the corners when increasing the grid resolution. The abrupt appearance of such a vortex impacts the overall flow, and correspondingly the other vortices. This phenomenon thus complicates the grid convergence analysis. Furthermore, because the lid-driven cavity flow is fully confined and involves boundary layers on all four walls, the gain achieved by using a non-uniform mesh is somehow limited; most Eulerian studies in the literature are actually based on uniform meshes. Nonetheless, this test case provides the opportunity to test different strategies of mesh refinement, as shown above.

4.3 Forced sloshing

The previous three test cases are classical examples for which an Eulerian approach would be better adapted. On the other hand, the PFEM can be of advantage when simulating flows with a free surface. To highlight the added value of adaptive mesh refinement and to illustrate the use of the proposed algorithm for free-surface flows, the sloshing in a tank undergoing a forced roll motion is now studied. The setup reproduces some experiments of Delorme et al. [99] and Souto-Iglesias et al. [100] for the case of water in the large tank with lateral impacts. A schematic of that setup and its dimensions is shown in Fig. 4.21 with the values of the geometrical and physical parameters summarized in Table 4.9.

The Reynolds number is defined with the initial water height H and the characteristic velocity $\sqrt{gH}$, where g is the gravity acceleration, such that the Froude number $\mathrm{Fr} = 1$. Using the shallow water dispersion relation, Souto−Iglesias et al. [100] estimate the resulting sloshing period as

$$T_s = 2\pi \left(\frac{\pi\, g}{L} \tanh\left(\frac{\pi\, H}{L} \right) \right)^{-1/2} = 1.9191\,\mathrm{s}\,. \tag{4.1}$$

The imposed rolling motion of the tank is an oscillatory rotational motion around the horizontal with an amplitude $\phi_{\max} = \pm 4°$ and a period T_f that is 85% of the natural sloshing period T_s. The corresponding tilt angle $\phi(t)$ with respect to the horizontal line has been taken from the experiment and is shown in Fig. 4.22. Finally, a pressure sensor is located on the left wall at a height corresponding to the initial water height H (red dot in Fig. 4.21).

[2]The units are m^{-1} instead of s^{-1} because the gradients are normalized by a scaling velocity.

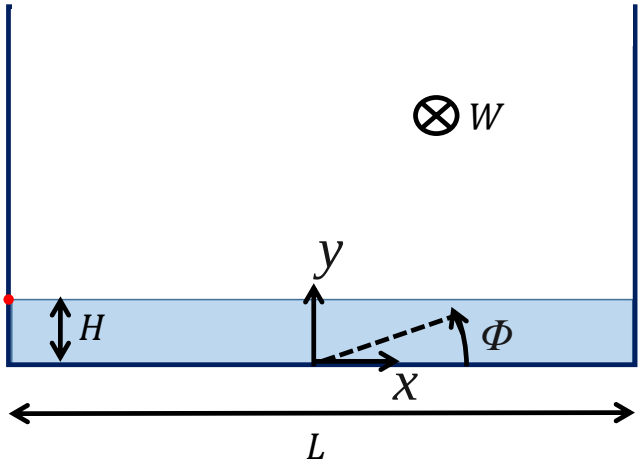


Figure 4.21: Schematic of the setup for the sloshing experiment of Delorme et al. [99] and Souto-Iglesias et al. [100]. Only two-dimensional simulations are performed, neglecting the presence of the front and back walls and the resulting friction. The red dot on the left wall indicates the position of the pressure sensor located at a height corresponding to the initial water height H. The respective numerical values are summarized in Table 4.9.

Initial water height	H	93 mm
Tank length	L	900 mm
Tank width	W	62 mm
Sloshing period	T_s	1.9191 s
Rolling amplitude	$\phi_{\max}$	$\pm 4°$
Density	ρ	998 kg/m^3
Dynamic viscosity	μ	$8.94 \cdot 10^{-4}$ Pa s
Surface Tension	σ	0.0728 kg/s^2
Reynolds number	Re	10^5
Froude number	Fr	1
Rolling period	T_f	1.6312 s

Table 4.9: Geometrical and physical parameters for the sloshing experiment of Delorme et al. [99] and Souto-Iglesias et al. [100] for the water case. The geometrical dimensions are defined in Fig. 4.21.

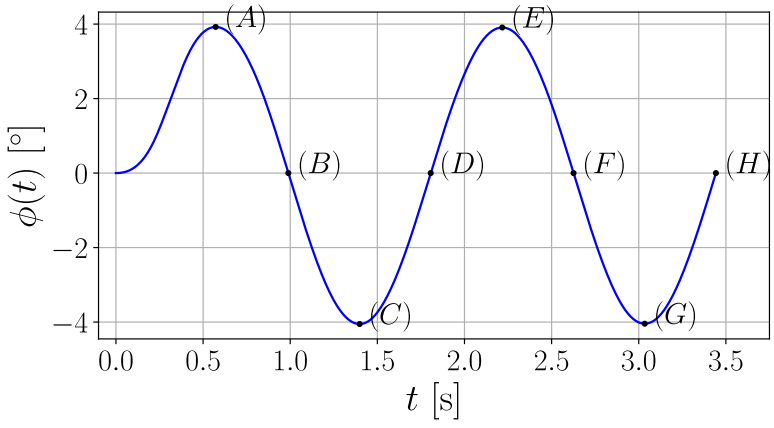


Figure 4.22: Tilt angle $\phi(t)$ of the tank as a function of the time taken from the sloshing experiment of Delorme et al. [99] and Souto-Iglesias et al. [100]. The points labeled (A) to (H) correspond to maximum, zero and minimum tilt.

Mesh adaptation

For this case, the mesh adaptation strategy relies on the combination of one geometric and two solution-based criteria. First, a mesh stretching from the minimum target mesh size $L^*_{\min} = 1.4\,\text{mm}$ at

$L^*_{\min}$	1.4 mm
$L^*_{\max}$	13 mm
$\|\|\widetilde{\nabla \mathbf{u}}\|\|_{\min}{}^2$	21.5 m^{-1}
$\|\|\widetilde{\nabla \mathbf{u}}\|\|_{\max}$	107.5 m^{-1}
β	1
U_ϵ	0.01 m/s

Table 4.10: Target mesh size parameters used for the solution-based mesh refinement of the sloshing test case.

the walls to $L^*_{\max} = 9.28 L^*_{\min}$ at a distance H from the walls is imposed through a geometric progression. Then, regions of large velocity gradients are refined according to Eqs. (3.23) and (3.24), while the free surface is refined using the criterion of Eq. (3.25). The corresponding parameters are summarized in Table 4.10. The results are also compared to those obtained on a uniform mesh with an element size $L^*_{\min}$ using the classical PFEM algorithm. Despite the slight differences in Eqs. (2.43) and (3.34), the same α is used for both mesh types. Moreover, two different values, $\alpha = 1.2$ and $\alpha = 1.4$, are considered to assess the impact of α on mass conservation. As shown in the following, the best results are obtained with $\alpha = 1.2$ for the non-uniform mesh and $\alpha = 1.4$ for the uniform mesh. These values are used when illustrating the results. Finally, the same time step size $\Delta t = 0.0005$ s is considered for all simulations.

Results

Figure 4.24 compares the deformation of the free surface between the uniform ($\alpha = 1.4$) and non-uniform ($\alpha = 1.2$) meshes at the time instants (A) to (H) defined in Fig. 4.22, i.e., at zero, maximum and minimum tilt angles. During this initial phase, both simulations lead to almost the same solution with small differences only visible after some time, for instance at the free surface close to the left wall (instant (G)) or towards the middle of the tank (instant (H)). This very good agreement between the two simulations is achieved despite a large difference in their mesh size. For the classical PFEM, its uniform mesh has the highest resolution everywhere so that its total number of nodes varies slightly around 29000 during this initial phase. On the other hand, with the new algorithm, the non-uniform mesh has an initial number of nodes of about 4200 that progressively increases due to mesh adaptation to about 7500 at instant (H), as shown in Fig. 4.23. The new algorithm can thus significantly reduce the computational cost while keeping the same accuracy. The effect of the curvature-based free-surface mesh adaption is clearly visible with a progressive refinement around the high-curvature zones, which allows a better description of the wave crest, as illustrated in Fig. 4.25. On the other hand, a much coarser mesh can be used in regions where the gradients are smaller.

The deformation of the free surface obtained with the non-uniform mesh is then compared to experimental measurements in Fig. 4.26 at the same time instants (A) to (H). The white dots correspond to the mesh nodes and are superposed to photographs of the experiment. Very good agreement can be observed for the free-surface motion during the first two sloshing periods ((A) to (F) in Fig. 4.26). Following the subsequent impact of the wave on the left wall, the flow becomes more chaotic, with splashes and individual vortices, and discrepancies become more apparent. The wavefront seems to break slightly earlier in the numerical simulations (instant (G)). Additionally, the free-surface reaches a higher height when impinging onto the right wall (instant (H)). Nevertheless, although discrepancies become more significant over time, the new algorithm provides a very good approximation of the flow and free-surface motion in this initial phase.

The experiment also includes pressure measurements at the pressure sensor (red dot in Fig. 4.21), which provide a more quantitative basis for comparison [99, 100]. The time evolution of the pressure is shown in Fig. 4.27. Very good agreement is found between the PFEM simulation on a non-uniform mesh (red line) and the experiment (green line) for the first wave reflection at $t \approx 1$ s and when the wave impacts the left wall at $t \approx 2.3$ s. The agreement remains good for the two subsequent impacts, in particular regarding the impact time, but a somewhat larger pressure level is predicted by the simulation in the post-impact phase (at $t \approx 4.3$ s and $t \approx 6$ s). Furthermore, the fourth impact seems to be predicted slightly earlier. The slower motion of the wave in the experiment is possibly due to the friction forces

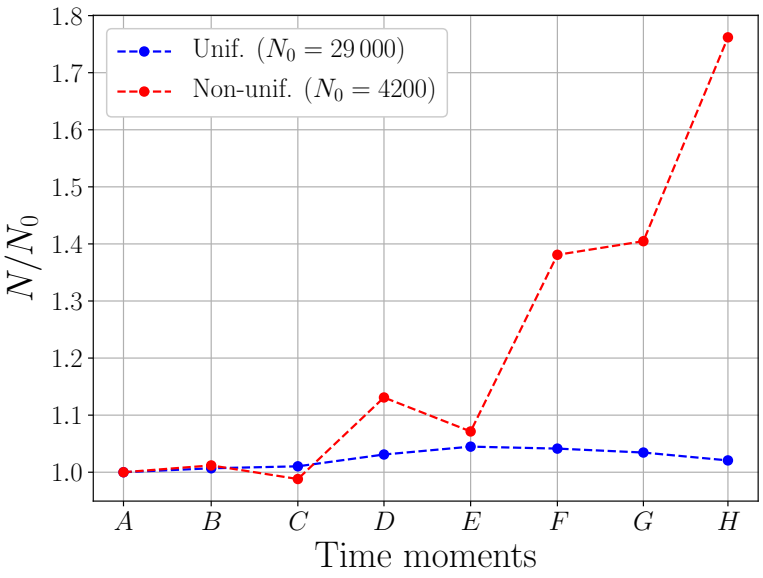


Figure 4.23: Evolution of the number of nodes at the time instants A to H (as shown in Fig. 4.24), for both uniform (blue) and non-uniform (red) meshes. The number of nodes in the non-uniform mesh progressively increases due to mesh adaptation. At instant H, the total number of nodes is $N \simeq 29500$ and $N \simeq 7500$ for the uniform and non-uniform meshes, respectively.

on the front and back walls of the tank, as the tank width is about 15 times smaller than its length. The presence of these front and back walls, and the corresponding friction, is not accounted for in the present two-dimensional simulations. Figure 4.27 also includes the results obtained with the classical PFEM on a uniform mesh with $\alpha = 1.4$ (blue line). In this case, although the first two impacts are well predicted, as expected from the above results, the long-time evolution of the pressure clearly shows discrepancies. In particular, a growing time lag is observed for the third, and mostly fourth impact. This time lag is even more significant with $\alpha = 1.2$ (not shown). Moreover, the post-impact pressure level continuously decreases at each impact.

To further analyze the differences between the two PFEM simulations and the experiment in the later phase, the free-surface deformation at each of the four impacts on the left wall is shown in Fig. 4.28. The main observation is that discrepancies between simulations and experiment increase at each impact, as the flow becomes more and more complex, three-dimensional and chaotic. Additionally, the results are much better with mesh adaptation than with the classical PFEM. For the non-uniform mesh, the figure illustrates well the mesh refinement at the free surface, close to the walls and in vortical regions. The overall free surface is well captured, despite some expected discrepancies in the impact region on the left wall. For the uniform mesh on the other hand, the aforementioned growing time lag is clearly visible and is particularly marked at the fourth impact. Even more striking in this case is the lower height of the free surface, which suggests non-negligible mass loss. Such mass destruction, and the resulting lowering of the average liquid height H, likely explains the time lag (larger sloshing period T_s, Eq. (4.1)) and possibly the decreasing pressure level over time that are observed with the classical PFEM on the uniform mesh.

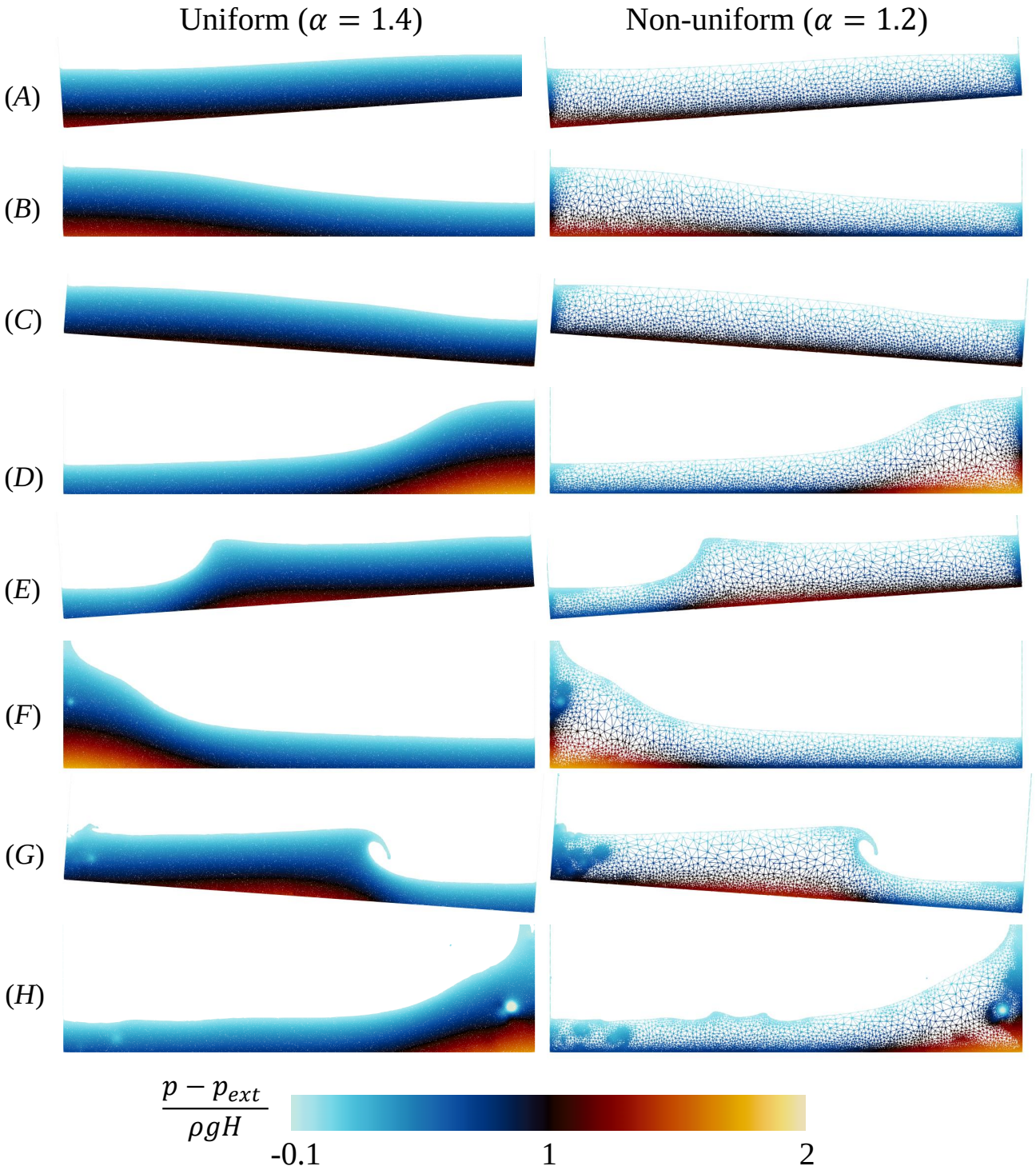
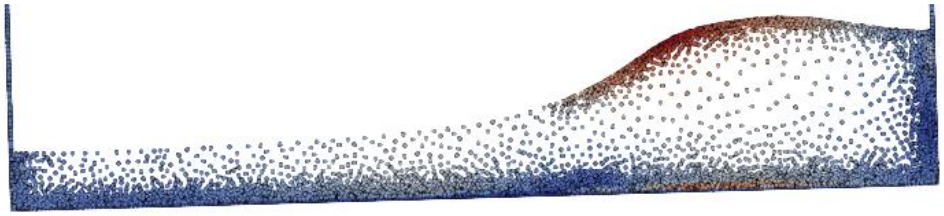
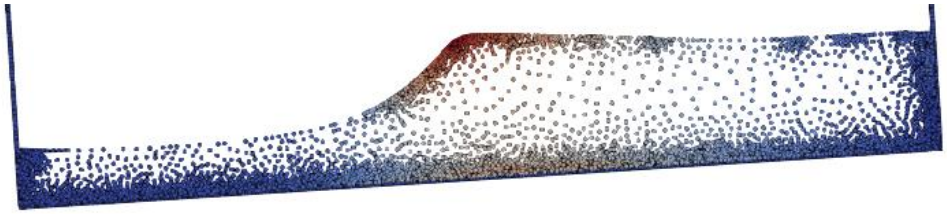


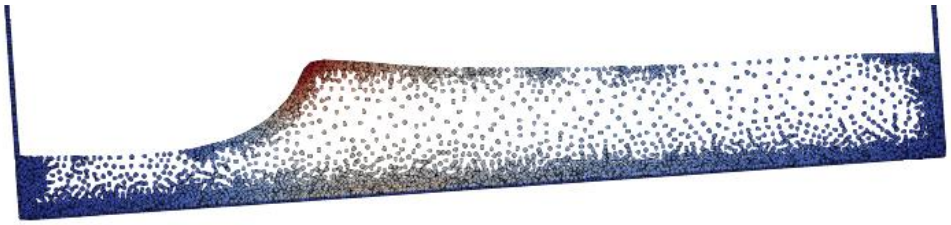
Figure 4.24: Free-surface deformation for the sloshing test case, at different times corresponding to the highlighted points in Fig. 4.22 for the uniform ($\alpha = 1.4$, left) and the non-uniform mesh ($\alpha = 1.2$, right). The color contour depicts the pressure field normalized by the initial hydrostatic pressure $\rho g H$ at the bottom of the pool.



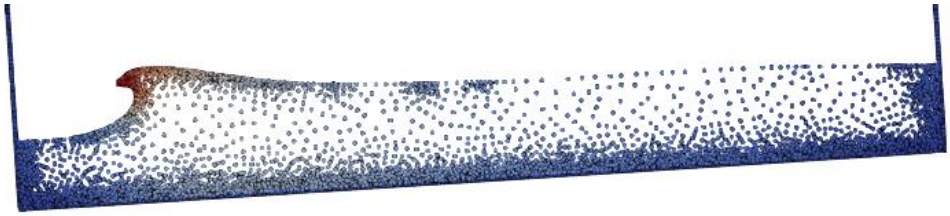
(a) $t = 1.9$ s, $||\mathbf{U}||_{\max} = 0.49$ m/s



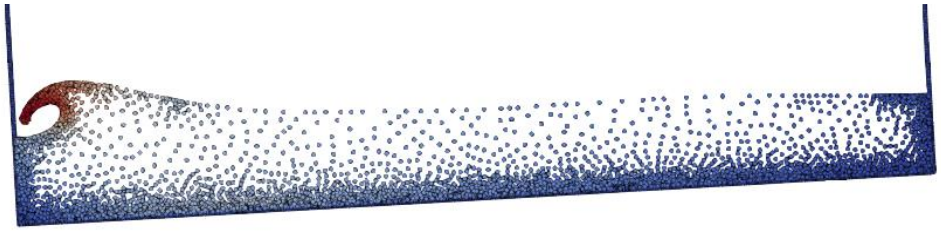
(b) $t = 2.1$ s, $||\mathbf{U}||_{\max} = 1.04$ m/s



(c) $t = 2.2$ s, $||\mathbf{U}||_{\max} = 1.42$ m/s



(d) $t = 2.3$ s, $||\mathbf{U}||_{\max} = 2.11$ m/s



(e) $t = 2.35$ s, $||\mathbf{U}||_{\max} = 2.23$ m/s

Figure 4.25: Nodal values of the velocity magnitude, at different times during the second sloshing period (between point (C) and (E) in Fig. 4.22). The corresponding maximum velocity is also reported.

Mass conservation

To further investigate the problem of mass conservation, Fig. 4.29 depicts the time variation of the total volume due to remeshing alone, ΔV_{rem}, and to both remeshing and time integration, $\Delta V_{\text{tot}} = \Delta V_{\text{rem}} + \Delta V_{\text{num}}$. The first observation is that the value of α has a non-negligible impact on mass conservation. In particular, a lower mass error is achieved with $\alpha = 1.4$ on a uniform mesh, while $\alpha = 1.2$ yields better results on the non-uniform mesh.

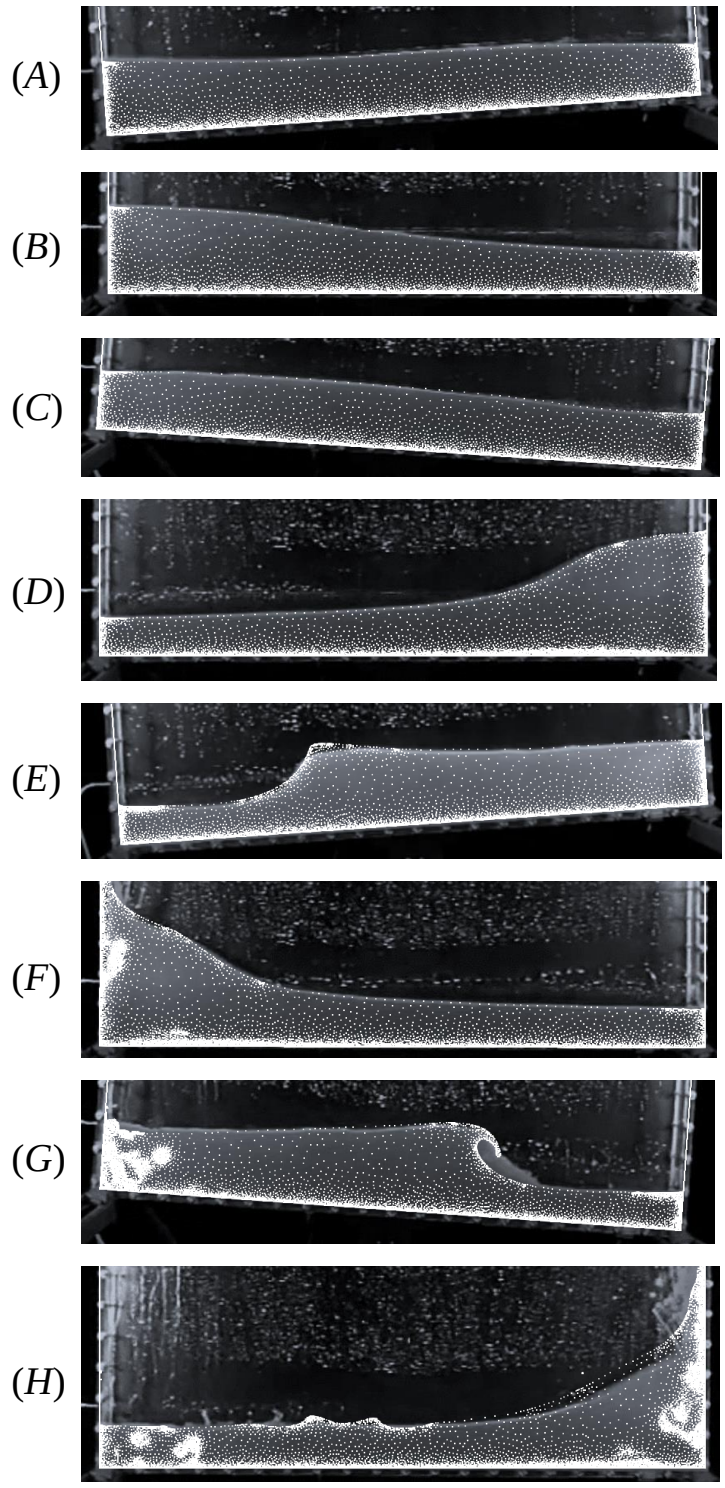


Figure 4.26: Free-surface deformation for the sloshing test case: PFEM computational nodes (white dots) superimposed to photographs of the experiment [99, 100] at different times corresponding to the highlighted points in Fig. 4.22.

The remeshing in the new algorithm leads to some mass creation (see Fig. 4.29(a)), but which remains limited compared to the larger mass destruction (about two times in magnitude when comparing the better results) in the classical algorithm. As already mentioned in Section 3.2.3, the new algorithm strongly reduces unwanted mass destruction at the free surface; this allows using a low value of α that limits mass creation. Note that most mass creation takes place at the vertical walls when wetted by splashes. As shown by the red dashed lines in Fig. 4.29, increasing α to 1.4 leads to a larger mass creation. On the other hand, a larger α reduces mass destruction with the classical PFEM on a uniform

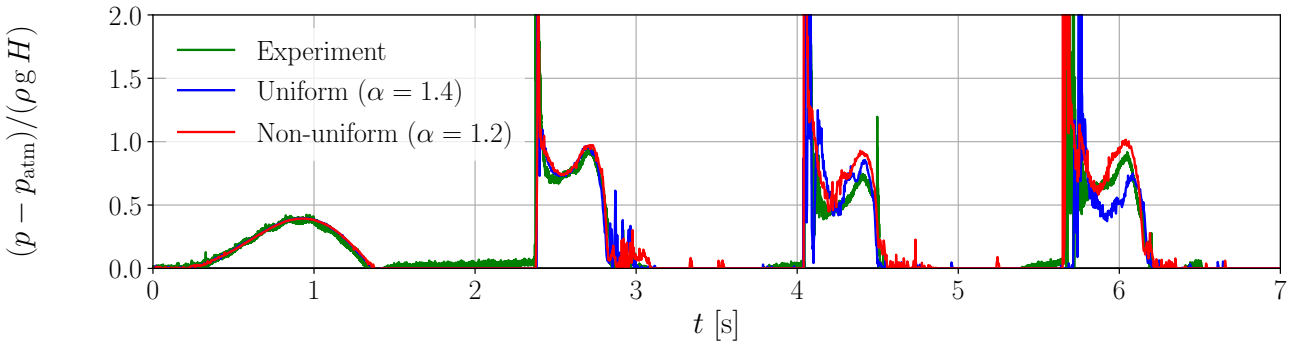


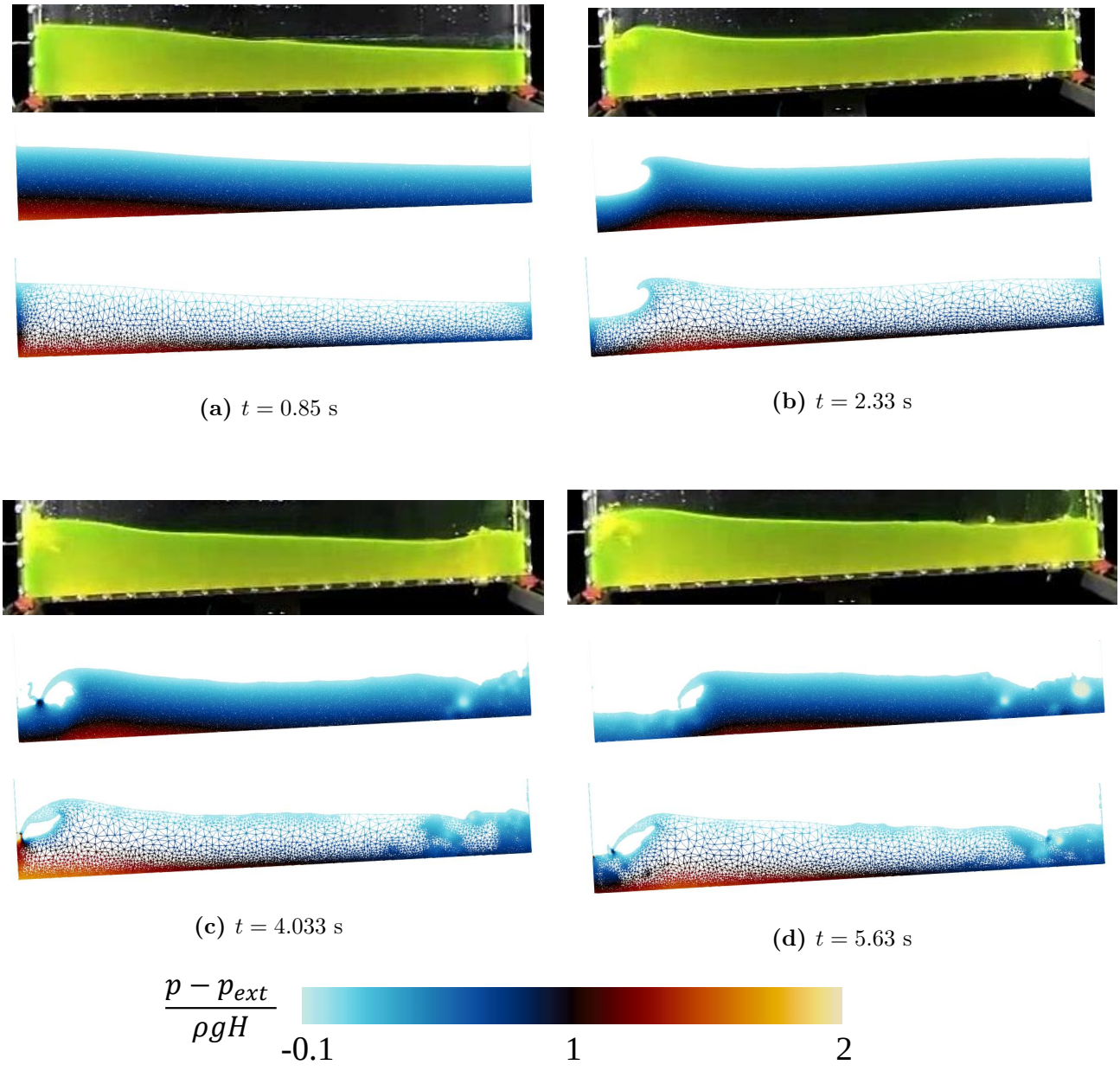
Figure 4.27: Non-dimensionalized pressure at the pressure sensor on the left wall of the tank as a function of the time. PFEM results obtained with a uniform mesh (blue) and a non-uniform mesh (red) are compared to the experimental measurements (green) [99, 100].

	Uniform	Non-uniform
N_{init}	~ 29000	~ 4200
$N_{\text{imp.2}}$	~ 30000	~ 5500
$N_{\text{imp.3}}$	~ 29000	~ 8000
$N_{\text{imp.4}}$	~ 26000	~ 8000

Table 4.11: Total number of nodes at the beginning of the simulation and after the second, third and fourth impacts on the left wall, for both the uniform ($\alpha = 1.4$) and the non-uniform ($\alpha = 1.2$) meshes. The decrease of the total number of nodes of the uniform mesh is due to mass destruction linked to nodes that are eliminated when too close to each other.

mesh. This mass destruction mostly results from nodes deleted on purpose to prevent a bad mesh quality when they get too close to each other. This is further supported by Table 4.11 that summarizes the total number of nodes at the beginning of the simulation and at the second, third and fourth impacts on the left wall. Because the uniform mesh has elements of size L^*_{min}, its initial number of nodes is approximately six times that of the non-uniform mesh. However, its mesh size decreases between the third and fourth impact. On the other hand, the total number of nodes of the non-uniform mesh expectedly increases over time due to mesh adaptation. Nevertheless, towards the end of the simulations, the number of nodes of the non-uniform mesh remains more than three times smaller than that of the uniform mesh. Finally, the numerical time integration induces in both cases some mass destruction (see Fig. 4.29(b)). For the new algorithm the two contributions, ΔV_{rem} and ΔV_{num}, cancel each other almost perfectly (probably by chance here) such that the total error in mass conservation is almost zero, while it amounts to more than 25% after 8 s with the classical PFEM using $\alpha = 1.4$.

Overall, the results obtained with the new algorithm are very good despite some discrepancies at later times, mostly due to the inherently chaotic nature of splashes, two-dimensional approximation and finite mesh resolution. Moreover, the use of mesh adaptation and boundary tracking reduces the error in mass conservation. In particular, the new algorithm allows choosing a lower value of α (more stringent criterion), which decreases mass destruction. Additionally, the value of α has been found to have an important impact on mass conservation. This value could potentially be further optimized for this specific case, but mass conservation would still remain a major challenge [44]. This issue can be mitigated with the new algorithm. Moreover, for the same maximum mesh resolution, the non-uniform mesh has a lower total number of nodes, such that the computational cost is significantly reduced. Finally, it should be mentioned that the significant error in mass conservation due to the time integration is most likely due to the accumulation of errors following the relatively long simulation time and repeated impacts at the side walls.



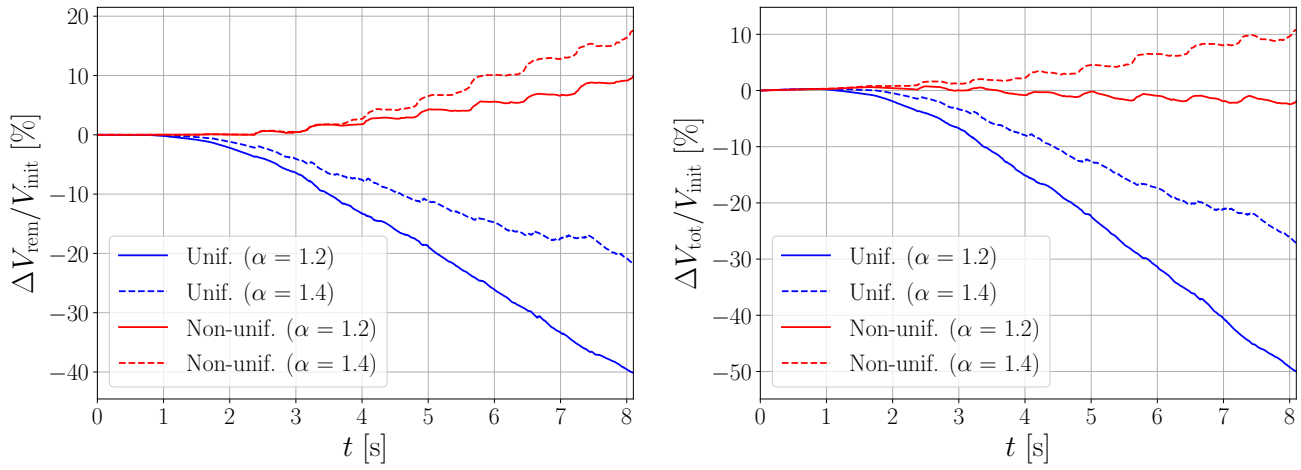
(a) $t = 0.85$ s

(b) $t = 2.33$ s

(c) $t = 4.033$ s

(d) $t = 5.63$ s

Figure 4.28: Free-surface deformation for the sloshing case at four different times corresponding to the first wave reflection and the three successive impacts on the left wall: photograph from the experiment [99] (top) and mesh wireframe from the PFEM simulations on the uniform mesh with $\alpha = 1.4$ (middle) and non-uniform mesh with $\alpha = 1.2$ (bottom), colored by the non-dimensional pressure.



(a) **(b)**

Figure 4.29: Variation of the total volume (mass) as a function of the time for the classical PFEM on a uniform mesh and for the new algorithm on a non-uniform mesh for both $\alpha = 1.2$ and 1.4 in Eqs. (2.43) and (3.34), respectively. (a) Error ΔV_{rem} due to the remeshing only, and (b) overall error including the error due to the time integration, $\Delta V_{\text{tot}} = \Delta V_{\text{rem}} + \Delta V_{\text{num}}$.

4.4 Drop falling into a liquid bath

The fifth test case considered is taken from Franci et al. [44] and corresponds to a two-dimensional drop falling into a bath of the same liquid. A schematic of the problem is shown in Fig. 4.30 and the key parameters are summarized in Table 4.12.

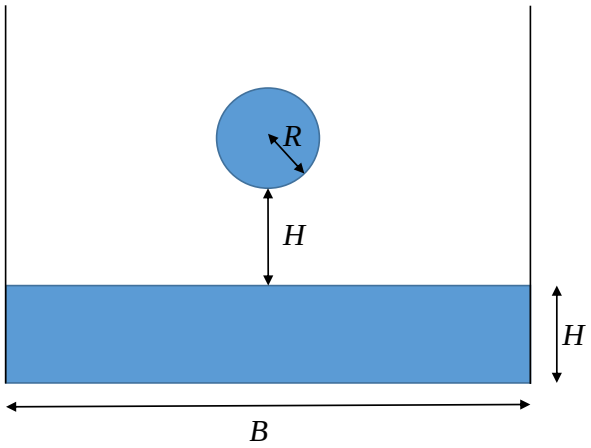


Figure 4.30: Schematics of the initial configuration of the test case where a two-dimensional drop falls into a tank full of the same viscous fluid, as proposed by Franci et al. [44]. The numerical values of the different geometric parameters are summarized in Table 4.12.

Initial liquid height	H	0.07 m
Length of the tank	B	0.3 m
Initial drop position	H	0.07 m
Initial drop radius	R	0.025 m
Liquid dynamic viscosity	μ	0.1 Pa·s
Liquid density	ρ	10^3 kg/m^3

Table 4.12: Geometric and physical parameters for the 2D drop falling into a liquid bath, as proposed by Franci et al. [44]. The definition of the geometric parameters is given in Fig. 4.30.

To reduce the computational time, the simulations begin with the drop closer to the free surface, as the initial phase only involves the drop's uniform acceleration. In practice, the simulation is started at $t_0 = 108.3$ ms, i.e., when the drop is at a distance $h_0 = 0.5R$ above the free surface, with the initial drop velocity $U_0 = \sqrt{2g(H - h_0)}$.

Mesh adaptation

Two different meshes are considered. The first one is uniform and the classical α-shape technique with $\alpha = 1.2$ in Eq. (2.43) is used, while the second simulation relies on a non-uniform mesh using the new refinement and boundary recognition algorithm with $\alpha = 1.2$ in Eq. (3.34). These results are then compared to the most resolved case of Franci et al. [44].

Again, the target size of the non-uniform mesh relies on a combination of geometric and solution-based criteria. First, the tank's side walls are discretized using the finest mesh resolution L^*_{min} in order to minimize the mass error during their wetting. Then, a geometric criterion imposes a linear increase of the target mesh size from L^*_{min} at the side walls to L^*_{max} at a distance $0.75\,D$ away from these walls. No special refinement is imposed on the bottom no-slip wall as the impact of the boundary layer in this region is considered small. Then, the curvature-based criterion, Eq. (3.25), is used to refine the free surface with $m = 270$, provided that the resulting target mesh size is not smaller than L^*_{min}. This is combined with a solution-based refinement relying on Eq. (3.23). On the other hand, the drop is initially discretized at its surface using a mesh size of L^*_{min}, which is slightly smaller (62%) than the value that would be imposed by the curvature-based criterion. Finally, this test case relies additionally on the contact detection algorithm to minimize mass conservation errors during the impact of the drop onto the bath's free surface and merging of the two fluid subdomains.

The mesh parameters for both the uniform and non-uniform meshes are summarized in Table 4.13 and the initial non-uniform mesh is shown in Fig. 4.31. The uniform mesh is similar to that of Franci et al. [44]. It is interesting to observe that the two meshes have approximately the same final number of nodes N_{post}. At last, the time step size is set to $\Delta t = 0.157$ ms for both simulations.

Results

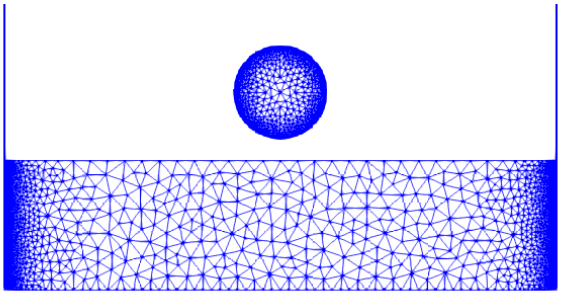


Figure 4.31: Initial non-uniform mesh for the falling drop. The mesh refinement in the vicinity of the side walls, as imposed by the geometric criterion, and at the drop surface is clearly visible.

	Uniform	Non-uniform
α	1.2	1.2
h [mm]	1.5	-
L^*_{min} [mm]	~ 1	0.18
L^*_{max} [mm]	~ 1	7.3
N_{init}	~ 14000	~ 8000
N_{post}	$\sim 14000 - 15000$	$\sim 14000 - 16500$
$\|\|\nabla \mathbf{u}\|\|_{\text{min}}$ [s^{-1}]	-	85
$\|\|\nabla \mathbf{u}\|\|_{\text{max}}$ [s^{-1}]	-	3.15
β	-	1/3

Table 4.13: Mesh parameters for the falling liquid drop test case. Both a uniform mesh similar to that of Franci et al. [44] and a non-uniform mesh using the solution-based criterion given by Eq.(3.23) are considered. The definition of α is given by Eq. (2.43) for the uniform mesh and by Eq. (3.34) for the non-uniform mesh. The number of nodes at the beginning of the simulation and after the drop impact is also reported.

A key feature of this test case is the merging of two fluid domains. As discussed above, the PFEM is well adapted to deal with such cases but it also suffers from mass conservation errors. The mesh refinement technique and the contact detection algorithm proposed here attempt to mitigate this issue. The application of the contact algorithm is illustrated in Fig. 4.32, which shows the mesh of the drop and bath free surface at different instants in time shortly before and during the merging process for both mesh types. The mesh of the falling drop is initially much finer than that of the flat liquid pool. Without contact detection algorithm, the merging of the drop with the bath occurs when the drop is within a distance of the bath mesh size of the free surface, with a thereby large associated mass error. The contact detection allows anticipating the merging and refining accordingly the liquid bath free-surface region around the expected impact point. The mass conservation error is thus lower, of the order of the smallest element size.

The motion of the bath free surface after the drop impact is shown in Fig. 4.33 and should be compared to Fig. 25 of Franci et al. [44]. The results with the uniform mesh are very similar to those of Franci et al., with small discrepancies, particularly close to the walls. These discrepancies can be explained by the chaotic nature of the problem. This high sensitivity to small perturbations is highlighted by the asymmetry of the solution at later times. The mesh resolution has thus also a clear impact, as demonstrated by the results of Franci et al. and the present comparison between uniform and non-uniform mesh. The more accurate representation of the free surface through local mesh refinement is further illustrated by the close view on a splash next to the right wall in Fig. 4.34. The non-uniform mesh yields a much more refined description of the liquid filament because mesh adaptation ensures that small elements are used there and the boundary recognition algorithm prevents elements to be removed unless they are smaller than the highest grid resolution or have their three nodes on the free surface. Nonetheless, the total number of nodes is kept limited by using a coarser mesh away from the free surface, which greatly reduces the overall computational time.

Mass errors

Unlike the previous test case, the error in mass conservation due to remeshing is here of the same order of magnitude for the classical PFEM on the uniform mesh and for the new algorithm, as illustrated in Fig. 4.35. For the uniform mesh this error oscillates between periods with predominantly mass destruction and periods of mass creation, such that the error seems to remain bounded over time. On the other hand, a monotone mass increase is observed with the new algorithm. This increase seems to take place during certain events between phases where the mass remains mostly constant. This is again due to the interface tracking algorithm which strongly reduces mass destruction. Despite the relative

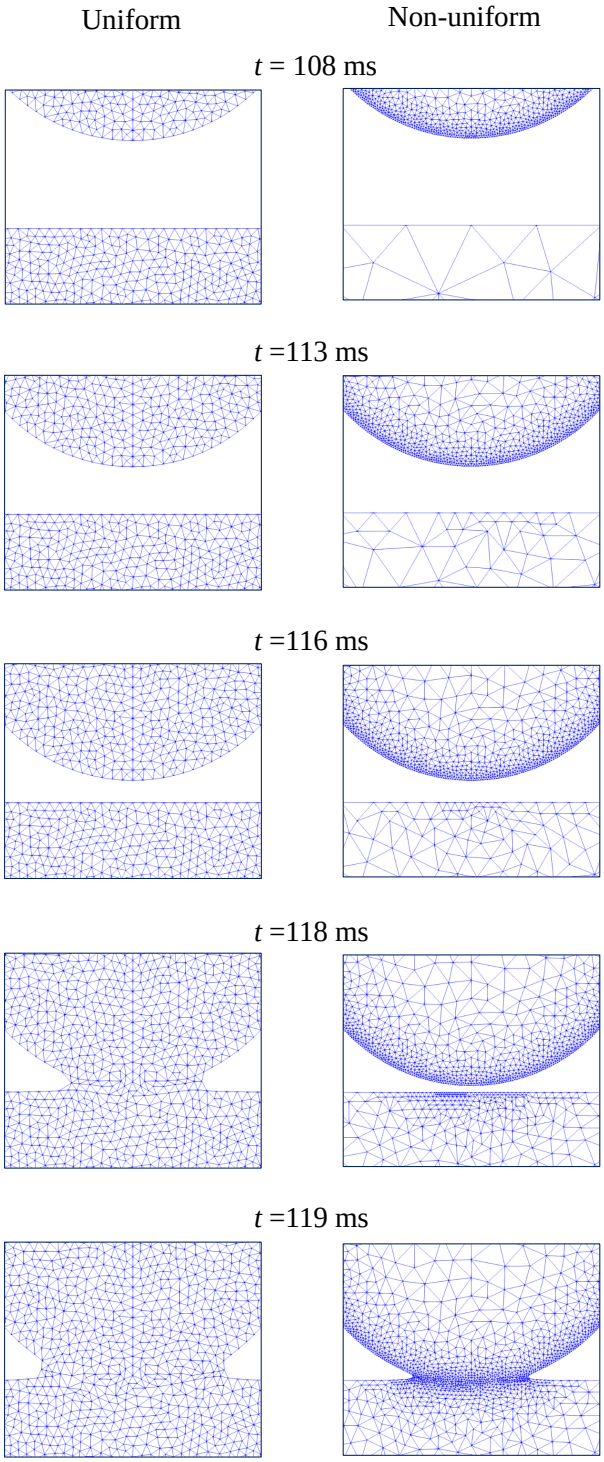


Figure 4.32: Comparison between a uniform mesh (left) and a non-uniform mesh with contact detection algorithm (right) at several time instants before the initial drop impact and during merging with the bath. The non-uniform mesh at the free surface of the liquid bath is initially much coarser than the respective mesh of the drop. By identifying the upcoming merging process, the algorithm can refine the liquid bath mesh in the expected impinging region. As shown by the last two time instants, the size of the added elements during the merging process is thus proportional to the smallest mesh size so as to minimize the mass conservation error.

small value of α, mass is created at the vertical walls during wetting events (see for example Fig. 4.34). A larger value of α would further exacerbate this issue. Owing to the contact detection algorithm, a much lower mass increase is also observed at the drop impact ($t \approx 0.1$ s). Despite the similar mass conservation error due to remeshing, the new algorithm allows a much better representation of the free surface, and, thereby, a more accurate time integration. This is shown by Fig. 4.35b, which represents the variation of the total volume (mass) due to both remeshing and time integration. Because the time

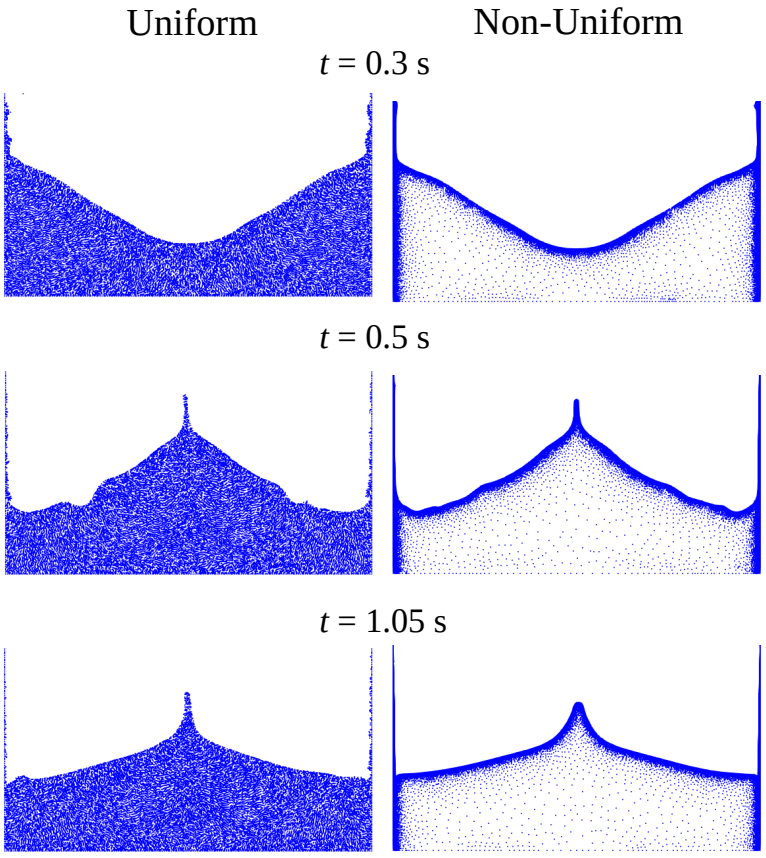


Figure 4.33: Free-surface deformation at three instants in time after the impact of the falling drop for the uniform (left) and non-uniform (right) mesh. The snapshots correspond to those in Fig. 25 of Franci et al. [44].

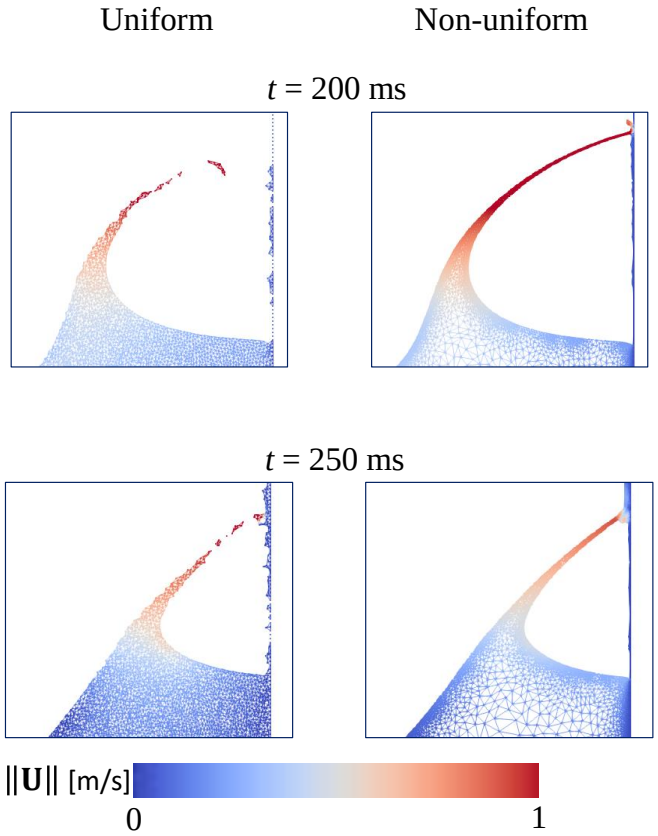


Figure 4.34: Free-surface deformation and splash at the right wall after the fall of a drop into a liquid bath. Two different time instants are shown for the uniform (left) and non-uniform (right) meshes. Contour of the velocity magnitude.

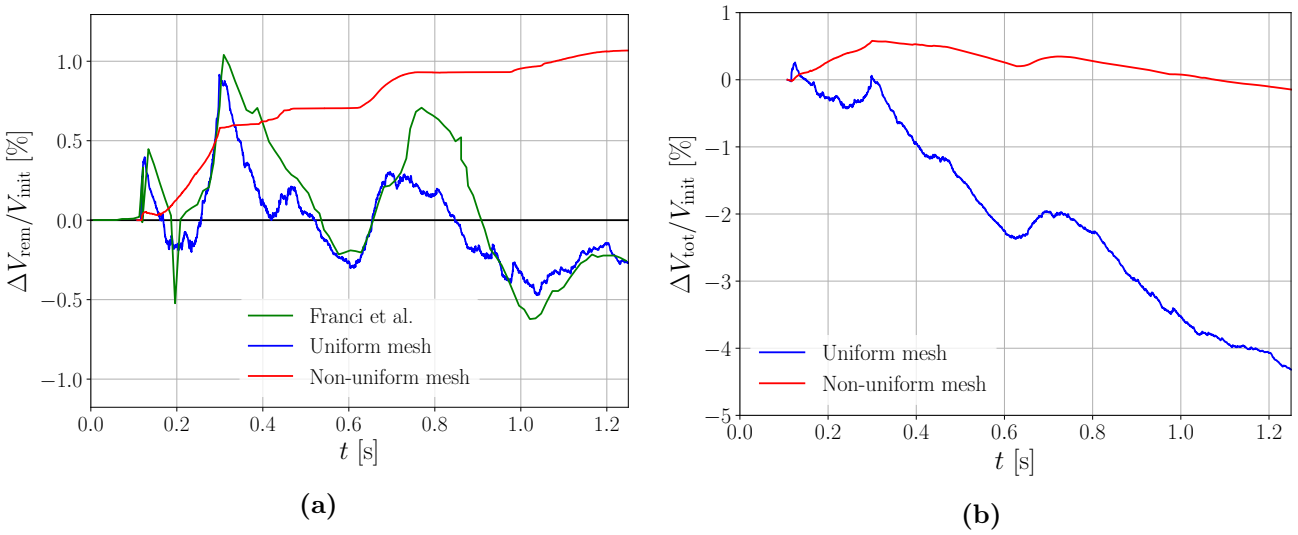


Figure 4.35: (**a**): Variation of the total volume (mass) due to remeshing as a function of the time for the falling drop test case: most resolved results of Franci et al. [44] (green), classical PFEM algorithm on the uniform mesh (blue) and new algorithm with contact detection on the non-uniform mesh (red). The parameters used for the simulations are summarized in Table 4.13.(**b**): Variation of the total volume (mass) due to both time integration and remeshing as a function of the time for the falling drop test case: classical PFEM algorithm on the uniform mesh (blue) and new algorithm with contact detection on the non-uniform mesh (red).

integration leads to mass destruction, the two contributions for the new algorithm again almost cancel each other, such that the net error remains very small. On the other hand, the classical PFEM shows a continuous mass decrease over time. This again illustrates the accuracy improvement brought by the new algorithm.

4.5 Cylinder pulled out of a bath at constant velocity

The last test case is directly related to the main application of interest. It corresponds to a solid cylinder of radius a that is initially immersed in a liquid bath at a depth d from the free surface and pulled out vertically at a constant velocity U (see schematics in Fig. 4.36). The main goal of this test case is to show how the present mesh adaptation algorithm provides a more accurate representation of a stretching free surface and thin film dynamics above and below the cylinder. Moreover, it is used to highlight the importance of using mesh adaptation to reduce the computational cost of the simulation, while enabling the use of a large domain to limit the impact of the boundary conditions on the flow. For this test case however, a rather small domain is used, in order to allow comparison with uniform meshes. It can already be pointed out that the mesh adaptation is essential to study this application with a reasonable CPU time.

The physical quantities, non-dimensionalized by the cylinder radius a, velocity U and density ρ, are summarized in Table 4.14. A free-slip boundary condition is imposed at the tank walls and a no-slip condition at the cylinder surface. The flow is characterized by an initial deformation and stretching of the free surface. Then, when the cylinder crosses the interface, a thin liquid film forms around it by entrainment and eventually drains along its surface to fall back into the bath.

Setup of the simulations

Three simulations are considered. The first one uses the classical PFEM with $\alpha = 1.2$ in Eq. (2.43) and a uniform mesh whose average element size is $L = 0.024$. The two other simulations are performed on two different non-uniform meshes with the new boundary tracking and mesh adaption algorithm and $\alpha = 1.2$ in Eq. (3.34). The corresponding target mesh size is based on a geometric criterion to refine the region around the cylinder. In particular the mesh size increases linearly from $L^*_{\text{min,GEO}}$ at the cylinder surface to $L^*_{\text{max,GEO}}$ at a distance d_{GEO} away from the cylinder. This is combined with a solution-based

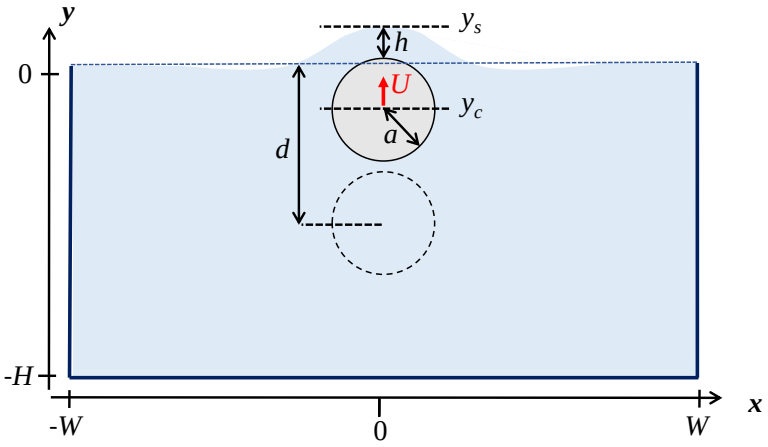


Figure 4.36: Schematics of the setup for the cylinder pulled out of a liquid bath at constant velocity.

Cylinder radius	a	1
Cylinder release depth	d	3
tank depth	H	6
tank half width	W	6
Cylinder velocity	U	1
Reynolds number	Re_D	200
Froude number	Fr	0.5

Table 4.14: Geometric and physical parameters for the case of the cylinder pulled out of a liquid bath at constant velocity. The geometric parameters are defined in Fig. 4.36. All quantities are non-dimensionalized by the cylinder radius a and its velocity U, except for the Reynolds number which uses the diameter $D = 2a$ as reference length scale.

		Coarse mesh	Fine mesh
GEO	$L^*_{\text{min,GEO}}$	0.02	0.005
	$L^*_{\text{max,GEO}}$	0.28	0.28
	d_{GEO}	5	5
Eq. (3.23)	$L^*_{\text{min,SOL1}}$	0.028	0.007
	$L^*_{\text{max,SOL1}}$	0.28	0.28
	$\|\|\nabla\mathbf{u}\|\|_{\text{min}}$	0.0058	0.0058
	$\|\|\nabla\mathbf{u}\|\|_{\text{max}}$	2	2
	β	1/3	1/3

Table 4.15: Parameters of the geometric and solution-based mesh refinement criteria to define the target size of the two non-uniform meshes for the case of the cylinder pulled out of a liquid bath at constant velocity.

criterion, Eq. (3.23), to ensure sufficient refinement in the wake of the cylinder. The parameters for these two target mesh size criteria are summarized in Table 4.15.

The uniform and the coarser non-uniform meshes have approximately the same maximum resolution, but the latter has initially about 15 times fewer nodes (see Fig. 4.37a). Because of mesh adaptation, the number of nodes of the non-uniform mesh expectedly increases during the simulation from about 4000 to 8500, but remains nonetheless seven times smaller than that of the uniform mesh. An illustration of the nodal density increase in the cylinder wake and at the free surface is shown in Fig. 4.37b. On the other hand, the finer non-uniform mesh has a four times higher resolution and its total number of nodes averaged over the simulation is approximately 60000, of the same order as the uniform mesh. Finally, the time step size is $\Delta t = 0.02$ for the first two simulations, and four times smaller, i.e., $\Delta t = 0.005$, for the simulation on the finer non-uniform mesh.

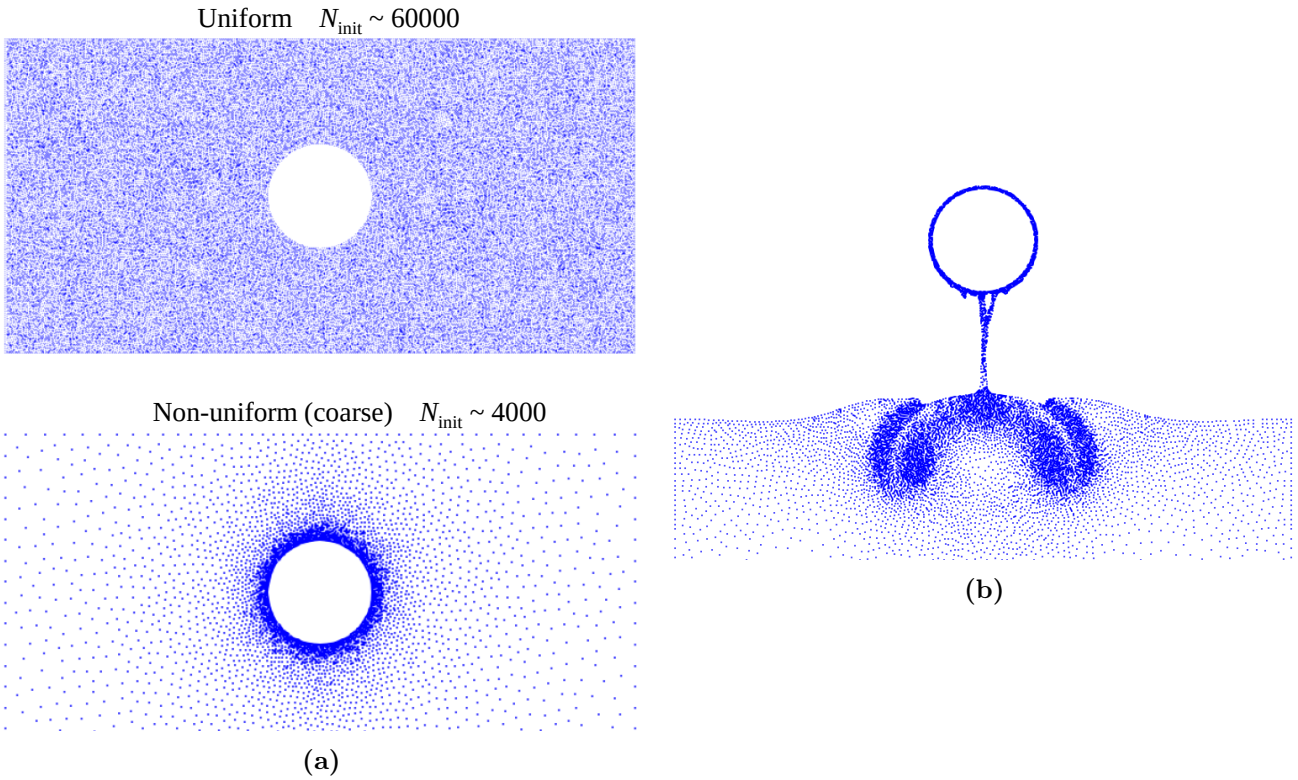


Figure 4.37:
(a) Initial distribution of mesh nodes for the cylinder pulled out of a liquid bath at constant velocity. The wireframe is not shown for better visibility. The uniform mesh (top) has initially about 15 times more nodes than the coarser non-uniform mesh (bottom) for the same maximum resolution.
(b) Distribution of mesh nodes for the coarser non-uniform mesh at time $t = 6$. The total number of nodes has increased up to 8500 because of the mesh refinement at the free surface and in the wake of the cylinder, but remains about seven times smaller than that of the uniform mesh.

Results

The free-surface deformation shortly before the cylinder crosses the interface ($t = 2.5$) and the drainage of the film when the cylinder is above the interface ($t = 6$) are compared for the three simulations in Fig. 4.38. During the initial phase, the free surface is stretched, such that some elements at the free surface becomes very obtuse and are consequently eliminated by the classical α-shape algorithm, contributing thus to mass destruction (see also Fig. 2.5). The new algorithm allows on the other hand a much more accurate representation of this free surface and thus greatly reduces mass destruction. This is even clearer at later times, where the surface of the film around the cylinder appears very rough and the filament draining down below it breaks into several "drops" for the classical PFEM. With mesh adaption and boundary tracking, the film thickness remains more homogeneous and the filament does not break yet. It is also interesting to observe that the solution remains much more symmetric with the finer non-uniform mesh.

Mass conservation is further analyzed in Fig. 4.39, which shows both the changes in total volume due to remeshing only, ΔV_{rem}, and to both remeshing and time integration, ΔV_{tot}. One can observe that, for the uniform mesh (blue dashed line), mass already starts to continuously decrease due to remeshing at $t = 2$, while with the new algorithm mass conservation is maintained until shortly before $t = 4$ (red dahsed line). However, a continuous mass increase is then observed with the new algorithm, which reaches the same level as the mass destruction for the classical PFEM at the end of the simulation. This creation of mass mostly takes place when two filaments dripping from the cylinder approach each other. When their distance is of the order of the neighboring element size, new elements are created between them, leading to their merging (see also Fig. 3.14b). Consequently, the use of a finer mesh (green dashed line) greatly reduces this mass increase. Unlike the previous cases, the time integration induces mostly a mass increase such that the overall error on mass conservation is decreased for the classical PFEM,

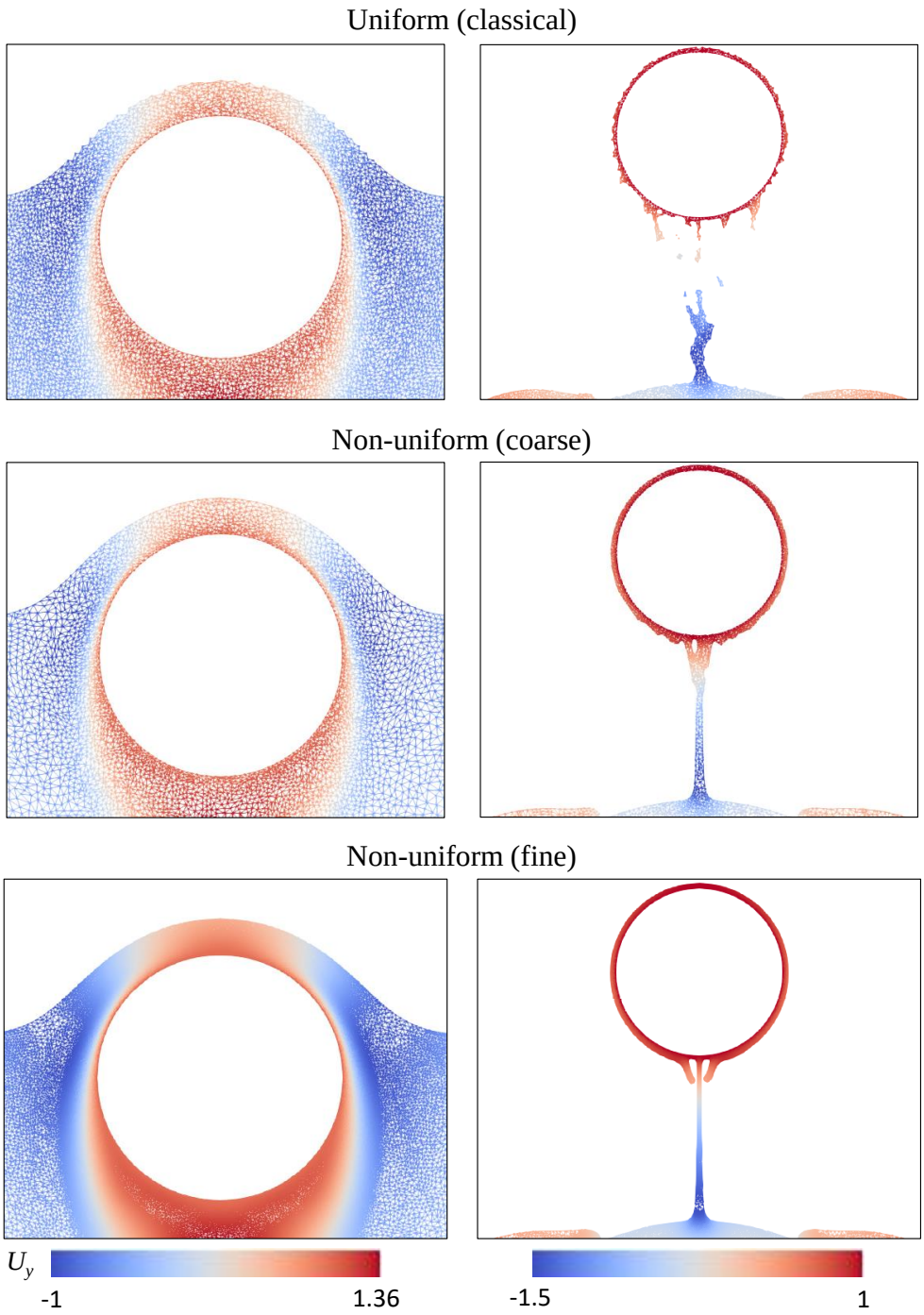


Figure 4.38: Wireframe mesh colored by the vertical component of the velocity at $(t = 2.5, y_c = -0.5)$ (left) and $(t = 6, y_c = 3)$ (right) for the cylinder pulled out of a liquid bath at constant velocity: classical PFEM on a uniform mesh (top), and new algorithm on the coarser (middle) and finer (bottom) non-uniform mesh.

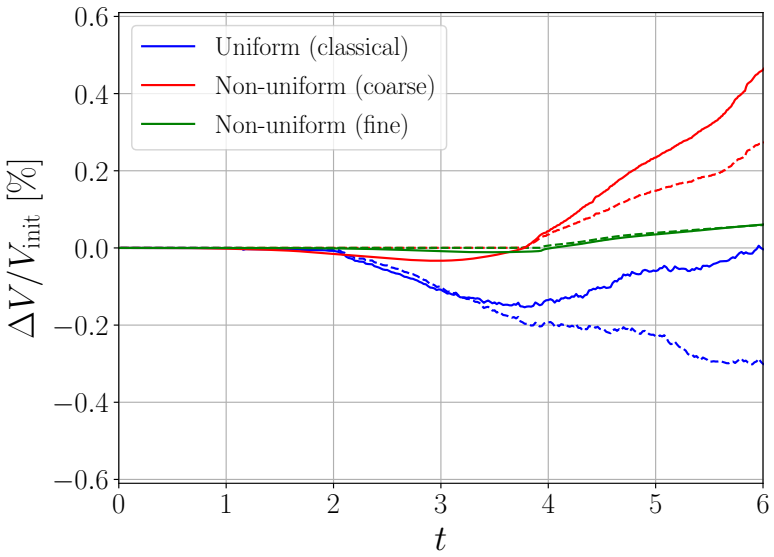


Figure 4.39: Variations of the total volume (mass) due to remeshing only, ΔV_{rem} (dashed line), and to remeshing and time integration, ΔV_{tot} (continuous line), as a function of time t for the cylinder pulled out of a liquid bath at constant velocity.

	Uniform (classical)	Non-uniform (coarse)	Non-uniform (fine)
N_{mean}	60×10^3	8.5×10^3	60×10^3
$T_{\text{tot}}/T_{\text{tot unif.}}$	1	0.16	2.8
$T_{\text{rem}}/T_{\text{rem unif.}}$	1	0.11	5.4
$\frac{T_{\text{rem}}}{N}$	0.03 s	0.047 s	0.16 s
$T_{\text{rem}}/T_{\text{tot}}$	7.53 %	14.1 %	14.5 %

Table 4.16: Total number of nodes averaged over the simulation, N_{mean}, total simulation time T_{tot} and time required for the remeshing (including mesh adaption and boundary tracking) T_{rem} for the cylinder pulled out of a bath at constant velocity. Both T_{tot} and T_{rem} are reported with respect to the classical PFEM on a uniform mesh. The ratio of remeshing time to total simulation time is also indicated. Note that the simulation on the fine non-uniform mesh uses a four times smaller time step size than the two other simulations.

while it increases for the new algorithm on the coarser mesh. On the finer mesh, the error due to time integration is almost negligible because of both the smaller elements and smaller time step size. It should be emphasized that the simulations approach their limit in terms of mesh resolution as the film around the cylinder and the filaments slowly reach a thickness of the order of the mesh size. The apparent better mass conservation with the classical PFEM at the end of the simulation is likely fortuitous as the overall thinning and draining of the liquid film do not seem to be physically accurate. Despite a non-negligible mass creation, the solution with the new algorithm on the coarser mesh resembles much more the solution on the fine mesh. As demonstrated by the results on the finer mesh, a reduction of the mass conservation error can be achieved by imposing a smaller target mesh size, but at the expense of a higher computational cost. It should be emphasized that this finer description of the free surface enables the study of the dynamics occurring in the film, which would have been impossible with the classical algorithm. Those physical aspects are discussed in Chapters 5 and 6.

Finally, the advantage of mesh adaptation is to reduce the computational time while keeping a similar accuracy or to increase accuracy while keeping a similar computational time. This is illustrated in Table 4.16, which summarizes the averaged number of nodes, N_{mean}, total simulation time T_{tot} and time used for remeshing (including the different steps of the new algorithm) T_{rem} for the three simulations. Despite an increase in the cost of the remeshing step relative to the overall computational time, the new algorithm allows a much lower number of nodes for a similar accuracy (size of the smallest elements), which translates into a reduction of the computational time by more than 85%. On the other hand, if the size of the smallest elements is reduced four times, a much more accurate solution is obtained.

In this case, the overall simulation time is about 2.8 larger than for the uniform mesh, although the total number of nodes is approximately the same for both cases. The reason for this increase is the four times smaller Δt. Although a factor of four would be expected, the lower computational cost observed is due to fewer iterations required to converge the nonlinear system of equations. Note that these results are representative of the other test cases, which clearly illustrates the advantage of the new algorithm.

Chapter 5

The 2D flow around cylinders and spheres pulled out of a bath at constant velocity

This chapter aims to study the physics of the flow around a body impulsively pulled out of a liquid bath at constant velocity. It mostly focuses on the case of a 2D rising cylinder, but the case of a rising sphere using a 2D axisymmetric formulation is also considered. The chapter is structured as follows. First, a literature review on the topic is presented. Then, a mesh convergence analysis is performed for the cylinder, which is followed by different analyses of the physics for a typical range of parameters. Some comparisons with experiments are then presented to complement the numerical analyses. Finally, the physics of flows at very low Reynolds number ($Re < 1$) is investigated numerically. The chapter ends with a brief discussion about the sphere to highlight the differences. The nomenclature for both cases is defined in Fig. 5.1. In addition, the non-dimensional numbers involved in this problem are summarized in Table 5.1.

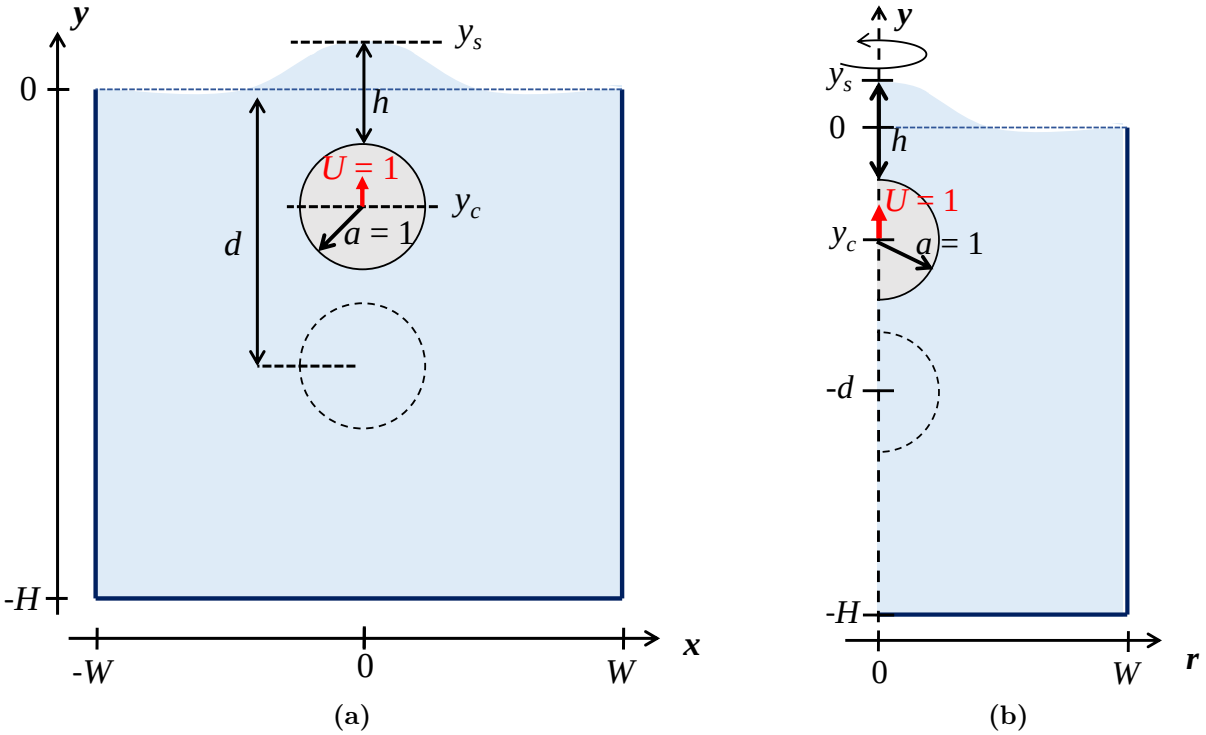


Figure 5.1: Nomenclatures for the cylinder **(a)** and the sphere **(b)** pulled out of a liquid bath at constant velocity: tank depth H and width $2W$, cylinder/sphere radius a, cylinder/sphere initial depth $d = -y_c(t = 0)$, vertical position $y_c(t)$ of the cylinder/sphere center, cylinder/sphere vertical velocity U, height $y_s(t)$ of the free-surface, thickness $h(t) = y_s(t) - (y_c(t) + a)$ of the liquid column above the cylinder/sphere apex. The origin of the vertical axis is at the initial location of the free surface. All quantities are non-dimensionalized by the cylinder radius a and velocity U.

Froude number	Fr	$U^2/g\,a$
Reynolds number	Re_D	$2U\,a/\nu$
Weber number	We	$\rho U^2\,a/\sigma$

Table 5.1: Three main non-dimensional numbers for the cylinder/sphere pulled out of a liquid bath at constant velocity. These definitions involve the cylinder/sphere radius a as the characteristic length scale, except for the Reynolds number which is based on the cylinder/sphere diameter $D = 2a$ instead, and velocity U. The other physical quantities g, ν, σ and ρ are respectively the acceleration of gravity, the kinematic viscosity, the surface tension and the density of the fluid.

.

5.1 Literature review

The interaction of a free surface with a submerged cylinder is a well-known subject in fluid mechanics. Havelock [101–103] studied the flow around a submerged cylinder in a uniform stream. More specifically, he studied analytically the vertical force on the cylinder using the method of images of the potential flow theory. Afterwards, Dean studied the diffraction of a linear wave due to a restrained cylinder [104], whose results have been later confirmed by Ursell [105]. At the same time, Havelock investigated a problem closer to the present problem of interest, i.e., the free-surface response to a cylinder started from rest [106, 107]. Havelock papers considered the cylinder at constant velocity (as in the present case) or constant acceleration, using linear theory, i.e., approximating the free-surface quantities at the initial free-surface position for all times (see Chapter 1). These results have later been extended to the nonlinear regime by Tuck [108]. On the other hand, among other authors, Tyvand and Miloh [109, 110], instead of analyzing the long-time evolution predicted by linear theory (with possible nonlinear corrections), studied the short time dynamics, successively triggering nonlinear effects, using a small-time expansion of the full nonlinear initial/boundary value problem. They found that, during the interface crossing, the thickness h of the film above the cylinder was almost uniform along the cylinder surface, i.e., independent of the angle from the cylinder apex. This result confirmed earlier inviscid simulations of the problem performed by Telste [111] and has been confirmed by Greenhow and Moyo [112]. A similar behavior is also found in the present work.

Until then, all theoretical works and simulations of this kind of problems relied on a potential flow formulation, which had been successfully used for a long time to represent free-surface flows. Unfortunately, the potential flow theory assumes an inviscid fluid such that viscous effects cannot be investigated in that framework. For instance, the "waterfall breaking" below the cylinder, as observed by Greenhow and Lin [113], cannot be predicted by inviscid simulations as it arises from vortices induced by viscosity.

However, more recent works have been done on similar problems involving viscous fluid flows and vortex dynamics. For instance, Horowitz & Williamson [114] studied the dynamics of freely rising or falling cylinders inside a fluid; their trajectories were found to be rectilinear or not, depending on the density ratio of the cylinder and the fluid. A second article by the same authors gives more attention to the influence of the vortex-induced vibrations for the similar problem [115]. Truscott et al. studied the unsteady forces on spheres during free-surface water entry [116] and the water exit dynamics of buoyant spheres [117]. Nair and Bhattacharyya [118] studied the cylinder water entry and exit problems using a volume of fluid method (VOF) [119]. Similar problems have been investigated by Ni et al. with a Boundary Element method (BEM) [120] and by Haohao et al. with a Lattice Boltzmann method [121]. Interestingly, the latter have shown that the free-surface elevation strongly depends on the Froude number for values lower than $Fr \approx 4$, while the dependency is less significant for larger Froude numbers. Wu et al. [122] also investigated this dependency for both fixed and free spheres rising towards the water surface.

The objective of this work falls into this line and aims to investigate the problem of a rising cylinder (or sphere) toward the free surface at constant velocity. Note that a work of collaborators can be found on the same subject [123]. Unlike most previous studies (e.g., Havelock [106, 107], Tyvand and Miloh [109, 110], Telste [111], Greenhow [112], etc.), viscous effects, such as boundary layers and wake dynamics, should be here accounted for. Moreover, the PFEM provides the opportunity to investigate the long time dynamics, including phases after the interface crossing, such as fluid separation below

the cylinder (sphere), but also the dynamics of the thin film above the cylinder. For instance, unlike traditional ALE methods, the PFEM, through its continuous remeshing, can represent the free-surface folding on itself. Within this framework the goal is to provide new results with respect to the literature, as well as to confirm some known results [111]. In particular, it is shown that the exit dynamics is governed by the balance between inertia and gravity, as also confirmed by the experiments of Liju et al. [124] at $Re > 500$. More specifically, the dependency of the free surface elevation with respect to the Froude number is highlighted.

5.2 The rising cylinder

The focus is here on the case of a cylinder rising towards the free surface at constant velocity. The main flow features have already been mentioned in Chapter 1. In particular, the following ones are of main interest:

- The free-surface surge and film thinning, measured by the free-surface displacement $y_s(t)$ and the fluid height $h(y_c)$ above the cylinder apex (see Fig. 5.1): the influence of different parameters on h is discussed, such as the release depth, the Froude number, and the wake dynamics.
- The skin friction coefficient c_f at the cylinder surface and the drag coefficient C_D, and the influence of the Froude and Reynolds numbers on those quantities. These quantities are also compared to the case without free surface to identify the role and contribution of this free surface.
- The flow in the wake of the cylinder. A comparison with the case without free surface is also done to identify the impact of the latter on the wake dynamics.

In a first step, numerical aspects related to the PFEM simulations are discussed. In particular, following the guidelines given in Section 3.3, mesh and time step convergence analyses are presented. Then, the main features of the flow are analyzed and discussed. This is followed by the comparison between simulations and in-house experiments [125], where focus is mainly on the film thickness $h(y_c)$ and the drag force.

5.2.1 Convergence analysis of the space and time discretizations

To assess the accuracy of the numerical results, the different sources of numerical errors should be quantified. For this purpose, the general discussion of Section 3.3 is exploited, and both mesh and time step convergence analyses are performed. The case of a rising cylinder is considered, whose parameters, using the nomenclature of Fig. 5.1a, are summarized in Table 5.2. The tank dimensions are chosen sufficiently large to minimize the effect of its walls.

5.2.1.1 Choice of the mesh resolution

As already discussed in Section 3.3, one important limiting parameter in terms of spatial resolution for CFD simulations is the thickness of boundary layers, which is directly related to the value of the

a	1
d	7
H	28
W	21
Re_D	10^3 & 10^4
Fr	0.5
We	500

Table 5.2: Parameters used in the mesh and time step convergence analyses for the case of a cylinder pulled out of a bath at constant velocity. All lengths (whose nomenclature is given in Fig. 5.1a) are non-dimensionalized by the cylinder radius a. The non-dimensional numbers are defined according to Table 5.1.

Reynolds number. As a reminder, in Section 3.3, the following relation for the minimal mesh size has been estimated:

$$L^*_{\mathrm{min,c}}(Re_D) = \frac{1}{4\sqrt{Re_D/2}}\,. \tag{5.1}$$

This gives $L^*_{\mathrm{min,c}} = 1/90$ and $1/280$ for the two Reynolds numbers considered here, where $L^*_{\mathrm{min,c}}$ is non-dimensionalized by the cylinder radius a. This provides a first estimate of the minimal mesh resolution to obtain an accurate solution.

In practice, the initial mesh is defined by imposing a grid spacing Δs on the boundaries. For the present case, $\Delta s = \Delta s_{\mathrm{min}}$ on the cylinder boundary and Δs_{max} on the tank walls. Note that this resolution remains unchanged during the simulation. Additionally, Δs_{min} and Δs_{max} are related to L^*_{min} and L^*_{max} through $L^* = \sqrt{2}\Delta s/2$ such that the square mesh size L^2 of a uniform triangular mesh would correspond to half the area Δs^2 of a square with side length Δs.

For the mesh convergence analysis, Δs_{max} has been set to 2 and kept fixed, while the convergence on Δs_{min} has been studied. Following values have been considered:

$$\Delta s_{\mathrm{min}} \in \left\{ \frac{1}{20}, \frac{1}{40}, \frac{1}{80}, \frac{1}{160}, \frac{1}{320} \right\}\,, \tag{5.2}$$

which corresponds to the following values for L^*_{min}:

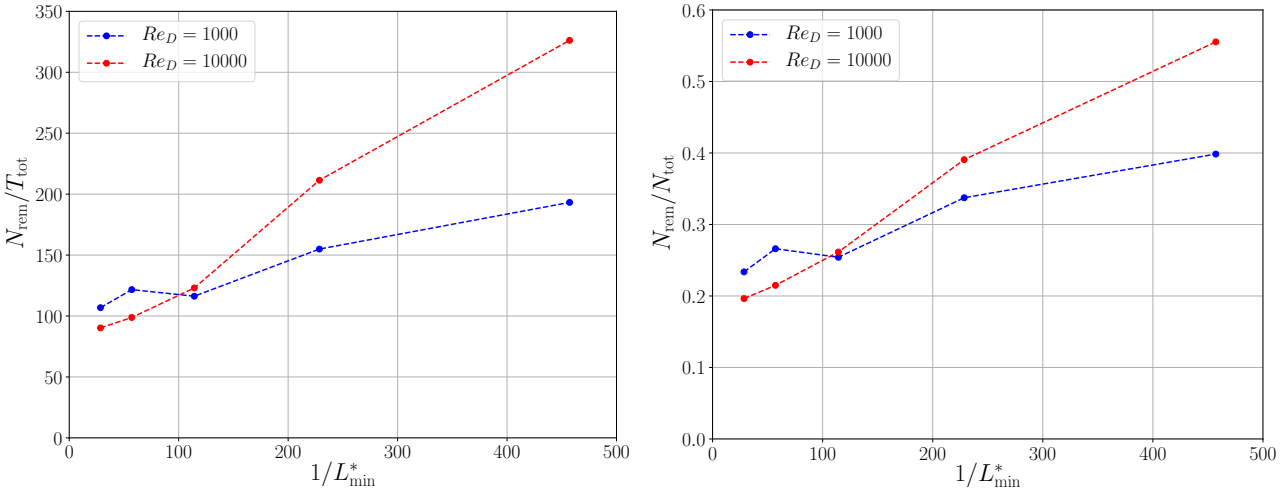
$$L^*_{\mathrm{min}} \in \left\{ \frac{0.7}{20}, \frac{0.7}{40}, \frac{0.7}{80}, \frac{0.7}{160}, \frac{0.7}{320} \right\}\,. \tag{5.3}$$

From that set, the two or three most refined meshes are expected to be sufficiently accurate (i.e., having a minimal mesh size smaller than $L^*_{\mathrm{min,c}}(Re_D)$) for the case $Re_D = 1000$, while only the most refined one should have a sufficient accuracy for the case $Re_D = 10000$. This is later confirmed to be more or less in agreement with the mesh convergence analysis. Note that, in order to minimize the impact of the time discretization, the same time step size, $\Delta t = 0.7a/(320U) \simeq 1/457$, is used for all simulations. It corresponds to the time step size leading to the maximum acceptable mesh distortion near boundaries for the most resolved mesh if the boundary layer were not resolved.

Because for low mesh resolution the mesh distortion over one time step is much lower than that obtained with finer meshes, it could be inefficient to remesh the domain at each time step. Instead, the remeshing takes place only once the distortion factor λ, Eq. (3.33), is above a given threshold. In practice, it is checked whether the average mesh distortion over a subset of elements whose distortion is higher than a first threshold λ_1, is higher than a larger threshold λ_2. The remeshing criterion mathematically writes as

$$\underset{\lambda_i > \lambda_1}{\mathrm{mean}}(\lambda_i) > \lambda_2 \quad \text{with} \quad \lambda_2 > \lambda_1\,. \tag{5.4}$$

It is a compromise between taking the average distortion over the entire mesh (a criterion where negligible distortion occurring in many places could mask the strong mesh distortion occurring in boundary layers) and taking the overall maximum distortion factor (a criterion that may be too sensitive to very localized mesh distortion, possibly even at a single isolated element). For the present analysis, $\lambda_1 = 3$ and $\lambda_2 = 3.5$ have been chosen. The total number of times remeshing is performed is plotted in Fig. 5.2 as a function of $1/L^*_{\mathrm{min}}$. In the left plot (Fig. 5.2a), the total number of remeshing processes is normalized by the total non-dimensional physical time of the simulation, while in the right plot (Fig. 5.2b), it is normalized by the total number of time steps. Remeshing takes place on average at every second iteration for the finest mesh, but only every five iterations for the coarsest mesh. One can also observe that, for the Reynolds number $Re_D = 10000$, the number of remeshing processes shows a stronger dependency on $1/L^*_{\mathrm{min}}$ than for $Re_D = 1000$.



(a) **(b)**

Figure 5.2: Total number of remeshing processes divided by the total (physical) simulation time (**a**) or by the total number of time steps (**b**).

Geometrical and solution-based mesh refinement setup

Together with the definition of the global maximal and minimal values of the target mesh size, L^*_{max} and L^*_{min}, the simulations use two different criteria to impose the local target mesh size. The first one is a geometric criterion, prescribing a linear progression of L^* from $L^*_{\text{min,GEO}} = L^*_{\text{min}}$ at the cylinder surface, to $L^*_{\text{max,GEO}} = L^*_{\text{max}}$ at a distance of 21 from the cylinder center. The other one is a solution-based criterion, given by Eq. (3.23), and based on the norm of the velocity gradient tensor, as described in Section 3.2.1. The parameters used for Eq. (3.23) are given in Table 5.3. An illustration of the effect of this solution-based mesh refinement is illustrated in Fig. 5.3. It can be seen that the mesh size in the wake is slightly smaller than outside of it, but remains limited in order to avoid a too large computational time. Note that, as the mesh size in the wake is similar for all meshes, the mesh convergence of the different flow features in the wake is not studied.

Mesh convergence parameters

For the mesh convergence analysis, the accuracy of different flow features is assessed, whose correct representation requires a sufficient resolution very close to the cylinder surface. The convergence in the wake is not studied, and it is assumed that the correct representation of the separation process is sufficient to obtain a good estimate of the impact of the wake on the free-surface deformation. Although wake instabilities are not accurately represented, the starting depth is supposed to be small enough ($d = 7$) such that no strong asymmetries appear (as a reminder, a much finer mesh in the cylinder wake was used for the cylinder at $Re_D = 9500$ in Section 4.1.4, and it was shown that the flow remained symmetric at least until $t = 5.5$).

Practically speaking, the following mesh convergence criteria are used:

$L^*_{\text{min,SOL}}$	0.7/20
$L^*_{\text{max,SOL}}$	0.7/2
β	1/3
$\|\|\nabla \mathbf{u}\|\|_{\text{min}}$	0.08
$\|\|\nabla \mathbf{u}\|\|_{\text{max}}$	2

Table 5.3: Parameters of the solution-based criterion for mesh adaptation used in the mesh and time step convergence analyses of the rising cylinder. All quantities are non-dimensionalized using the cylinder velocity U and its radius a. Note that $L^*_{\text{min,SOL}}$ and $L^*_{\text{max,SOL}}$ respectively play the role of L^*_{min} and L^*_{max} in Eq. (3.23), but differ from the global extreme target mesh sizes.

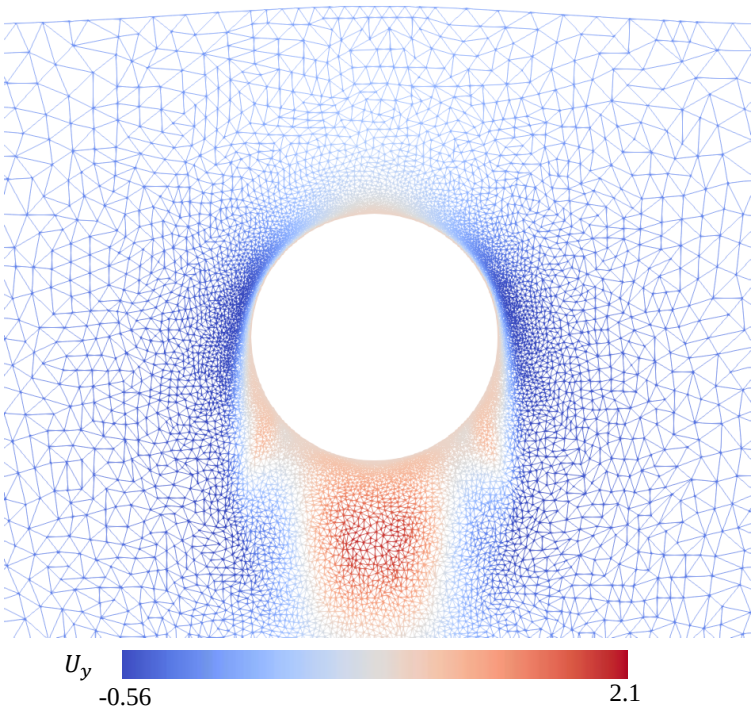


Figure 5.3: Wireframe mesh colored by the vertical component of the velocity, u_y, for the Reynolds number $Re_D = 1000$. The minimal mesh size is set to $L^*_{\min} = 0.7/80$ and the parameters used for the solution-based mesh refinement are given in Table 5.3. The image corresponds to the time $t = 4.5\,(y_c = -2.5)$.

- **The skin friction coefficient c_f at the cylinder surface, at different specific times.** It is a typical characteristic quantity used in the context of mesh convergence analysis in CFD applications. The skin friction coefficient is defined as

$$c_f = \frac{\tau_{w,1}}{\frac{1}{2}\rho U^2} = \frac{\mu \left.\frac{\partial u_t}{\partial n}\right|_w}{\frac{1}{2}\rho U^2}, \tag{5.5}$$

  where $\tau_{w,1}$ is the wall shear stress, and $\left.\frac{\partial u_t}{\partial n}\right|_w$ is the variation, along the wall-normal direction, of the tangential component u_t of the velocity field at the wall.

- **The time evolution of the drag coefficient $C_D(t)$.** It is defined as the total force opposed to the cylinder motion, divided by $\frac{1}{2}\rho a U^2$. The drag coefficient depends on both the friction at the surface and the pressure difference between the top and the bottom surfaces of the rising cylinder. In general, the contribution of the pressure to the total drag is well captured if the separation of the boundary layer is, which should be the case provided that the skin friction coefficient is well captured everywhere.

- **The time evolution of the film thickness $h(t)$ and its specific value $h^* = h(t_{y_c=-1})$ when the cylinder apex reaches the free-surface initial position.** Because it is the key quantity of interest in this work, it is important to represent it accurately.

Results

The skin friction along the cylinder surface is shown in Fig. 5.4 and 5.5, for different cylinder positions y_c (from -5 to 2) and the two Reynolds numbers. For both Reynolds numbers, increasing the spatial resolution leads to some monotonic, but not complete, convergence and to a reduction of oscillations. Also, unresolved simulations strongly underestimate the surface friction. This can be understood by looking at the boundary layer region, as illustrated in Fig. 5.6. In that figure, the cylinder side ($\theta = \pi/2$) is shown at position $y_c = -3$ for the case $Re_D = 1000$, where the coarse mesh

with $L^*_{\min} = 0.7/40$ is superposed to the finer mesh with $L^*_{\min} = 0.7/160$. It is clear that with the coarse mesh (thick black lines), the boundary layer, illustrated by the contour plot of the y-component of the velocity field obtained with the finer mesh (grey lines), cannot be properly resolved. Therefore, the linear finite elements adjacent to the cylinder surface underestimate the velocity gradient at the wall, $\frac{\partial u_t}{\partial n}\big|_w$, which results in an underestimation of the skin friction coefficient. Moreover, the observed oscillations of c_f are also directly related to the poor resolution. More specifically, from one element adjacent to the wall to the next element, the velocity slopes $\frac{\partial u_t}{\partial n}\big|_w$ are directly correlated with the size and shape of these elements.[1] They can therefore differ from one element to the next, along the cylinder surface. This becomes less pronounced as the boundary layer becomes more resolved. In this case, the slope is directly related to the physics and less dependent on the elements size, as confirmed by the lower oscillations observed for the meshes with the higher resolution.

The second observation is that, globally, the results for all meshes are closer to the converged solution for the case at $Re_D = 1000$ because, as previously mentioned, the boundary layer is thicker in this case. In particular, it seems that the friction coefficient is completely converged for the finer mesh ($L^*_{\min} = 0.7/320$), and almost converged for the mesh with $L^*_{\min} = 0.7/160$. For the case $Re_D = 10000$, it seems that none of the meshes has a sufficient resolution to obtain a fully converged solution. However, the difference between the two finer meshes (orange and red curves) is not too large.

In light of these results, $L^*_{\min} = 0.7/160$ is deemed to be sufficient to obtain a correct representation of the boundary layer close the the cylinder surface for all cases where $Re_D \leq 1000$. For the cases with $1000 < Re_D < 10000$, a similar resolution is used as the results are not too far from convergence and the computational cost remains manageable. However, it should be kept in mind that in that case, the results still entail a small numerical error. Finally, for the cases with $Re_D > 10000$ a finer mesh should definitively be used in order to get acceptable results, i.e., results sufficiently accurate to provide a meaningful description of the physics.

The second quantity of interest considered in this mesh convergence analysis is the drag coefficient C_D. Note that the correct computation of the skin friction coefficient is necessary to correctly predict C_D, because c_f has both a direct and indirect effect on it, respectively through the viscous shear stress, on the one hand, and through the wake separation and associated pressure drag on the other hand. The drag coefficient is shown for the two Reynolds numbers in Fig. 5.7. The two best resolutions give almost the same solution (except for a small discrepancy at $Re_D = 10000$ around $y_c = -2$). At $Re_D = 1000$ convergence seems to be even achieved for a resolution $L^*_{\min} = 0.7/80$. Similarly to the skin friction coefficient, a reduction of the oscillations is also observed when the mesh is refined.

A comparison of the present results at $Re = 10000$ with the drag coefficient of the impulsively started cylinder in Section 4.1.4 (see Fig. 4.18) shows that the oscillations of the drag coefficient at $t \approx 3$ to 5.5 in Fig. 4.18 are much less prominent in Fig. 5.7 ($y_c \approx -4$ to -1.5). These much weaker oscillations could possibly be explained by a poor mesh resolution in the cylinder wake. In the present mesh convergence analysis, the wake is not progressively refined, while the test case of Section 4.1.4 has been computed on a much finer mesh, allowing small vortices to develop and modulate the drag. Another likely explanation for this difference is the presence here of the free-surface, as discussed later in Section 5.2.2.2.

To assess the impact of the mesh resolution on the free-surface deformation, the film thickness $h(t)$ for different mesh sizes is shown in Fig. 5.8. No difference can be observed in the initial phase. Discrepancies begin to appear and progressively grow above $y_c = -1$ when the cylinder crosses the interface and the film starts thinning. This is however partly due to the semi-log representation. Beyond the need to choose a sufficient resolution to resolve the boundary layer, the required accuracy also depends on the minimal thickness $h(t)$ that one would like to resolve. In other words, the film keeps thinning until its thickness approaches the maximum grid resolution $L^*_{\min}$. Beyond this point, the mesh becomes much too coarse to correctly represent the physics[2], as illustrated by the coarsest meshes in Fig. 5.8. The minimum mesh size thus imposes a maximum simulation time. Using the second maximal resolution, $L^*_{\min} = 0.7/160$, the overall simulation can be run until $y_c = 6$ ($y_c = 2$) for $Re = 1000$ ($Re = 10000$)

[1] The value of the velocity slope is constant on the element as linear shape functions are used.

[2] In some cases, beyond the resolution limit, viscous-dominated flows are recovered since viscosity strongly dominates the flow, and the thin film theory should apply perfectly [126], using the lubrication approximation of Reynolds [127]. Therefore, the behavior below this resolution can be obtained from the literature.

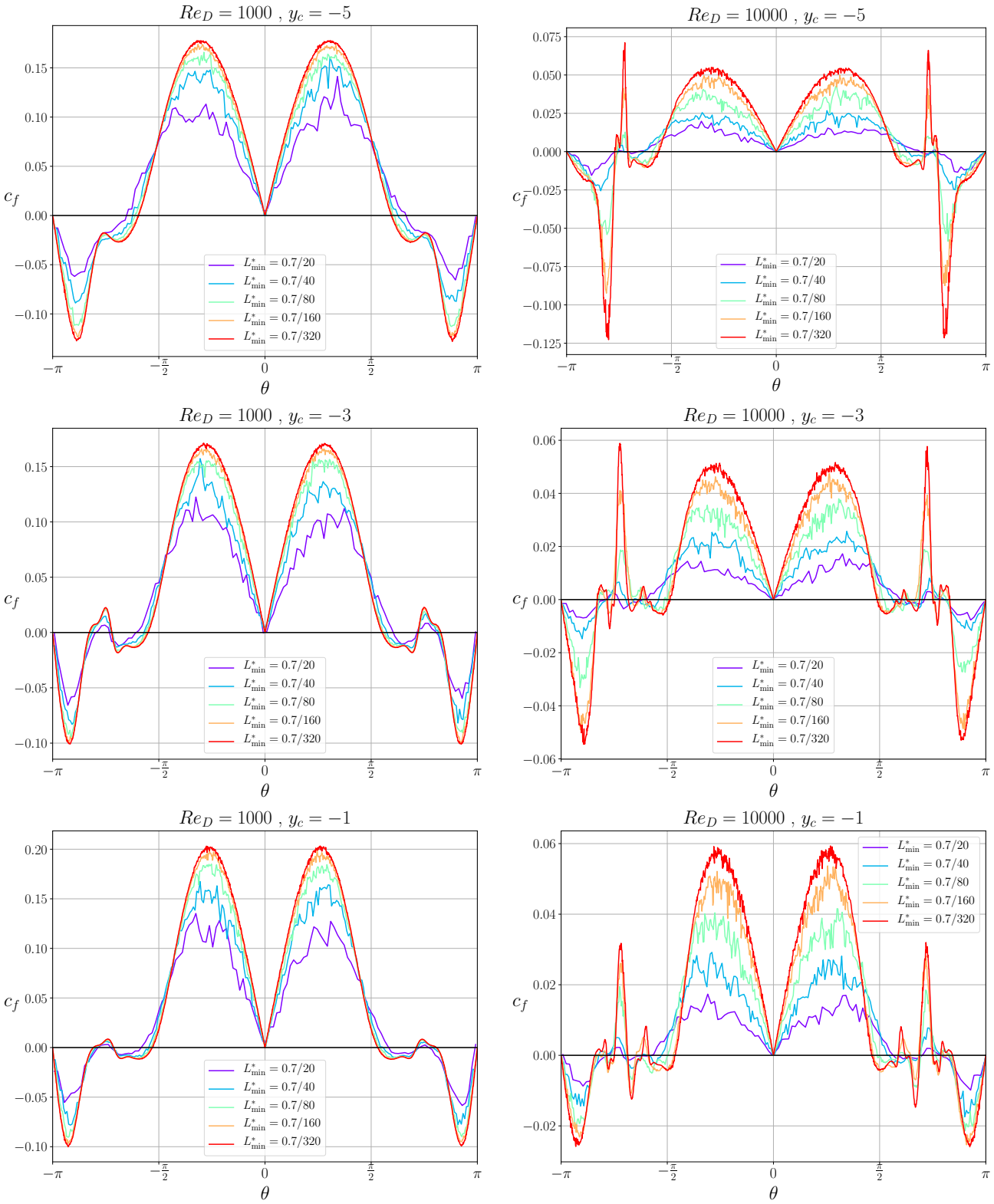


Figure 5.4: Skin friction coefficient on the cylinder surface at several time instants (i.e., $y_c = -5, -3, -1$) for meshes with different minimal mesh sizes $L^*_{\min}$. The top of the cylinder corresponds to the value $\theta = 0$. The left column is for $Re_D = 1000$ while the right column is for $Re_D = 10000$.

while still having several layers of mesh elements to represent the film. This is deemed sufficient within the scope of the present work. If the later dynamics is of interest, then the grid resolution, at least in the film, must be increased. Finally, Fig. 5.9 shows the variation of the specific value $h^* = h(t_{y_c=-1})$ of the film thickness when the cylinder apex reaches the free-surface initial position for different mesh resolutions. One can observe that the difference between the two most refined meshes is less than 0.5%, suggesting again that the second finest mesh is sufficient to have a good representation of the free

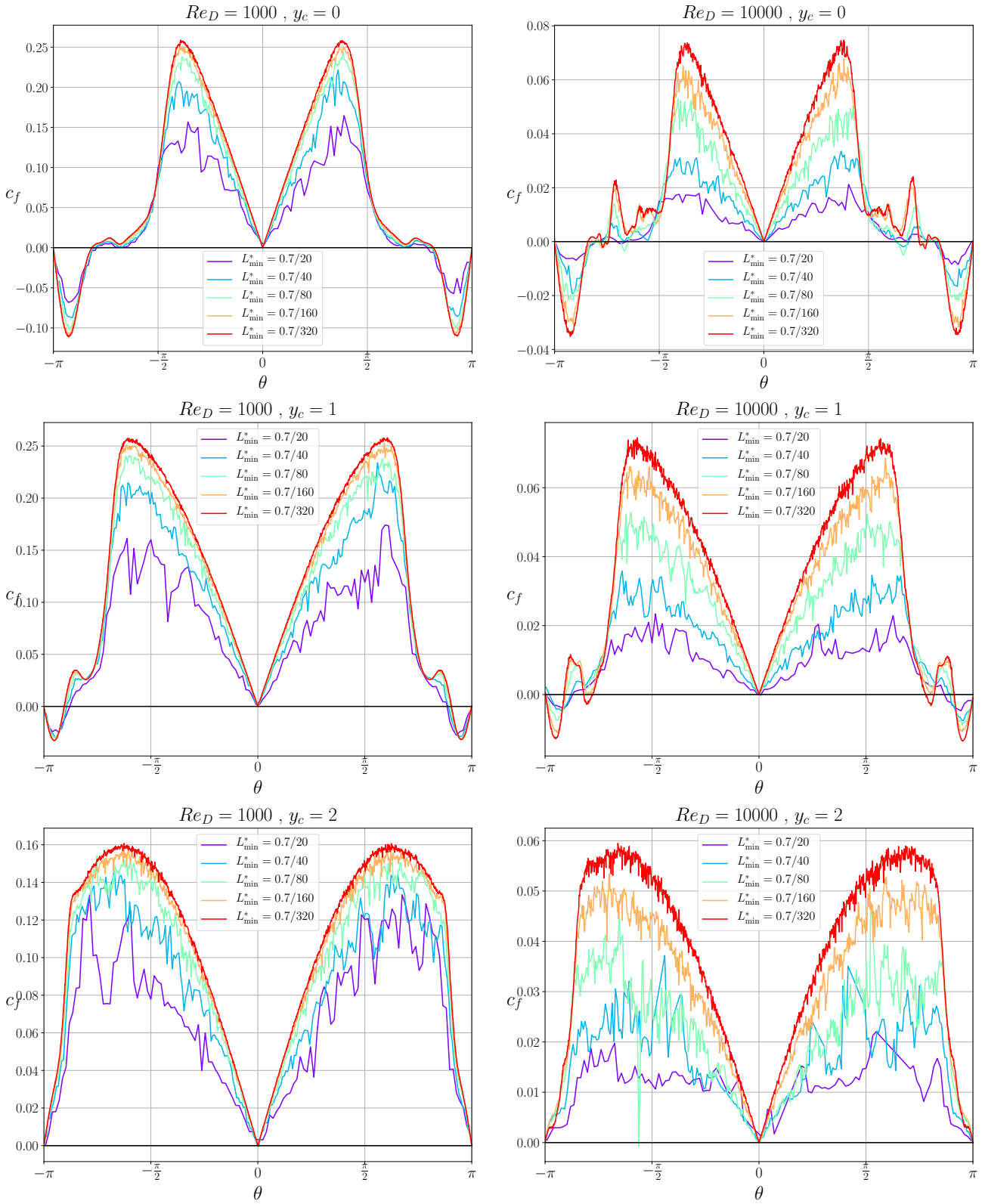


Figure 5.5: Skin friction coefficient on the cylinder surface at several time instants (i.e., $y_c = 0, 1, 2$) for meshes with different minimal mesh sizes $L^*_{\min}$. The top of the cylinder corresponds to the value $\theta = 0$. The left column is for $Re_D = 1000$ while the right column is for $Re_D = 10000$.

surface.

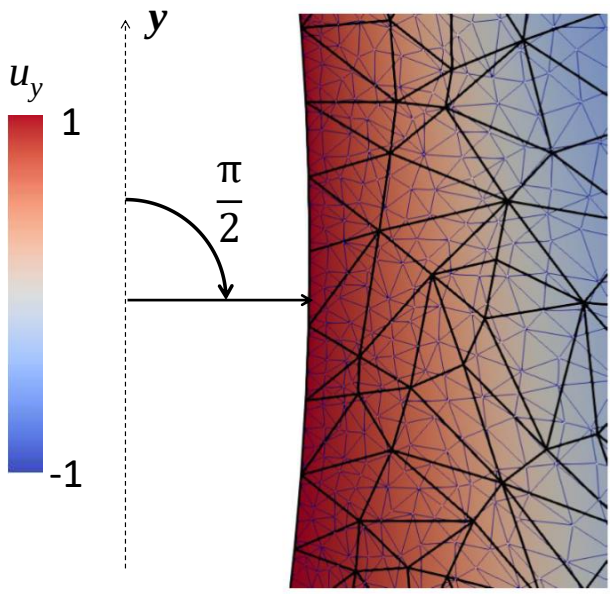
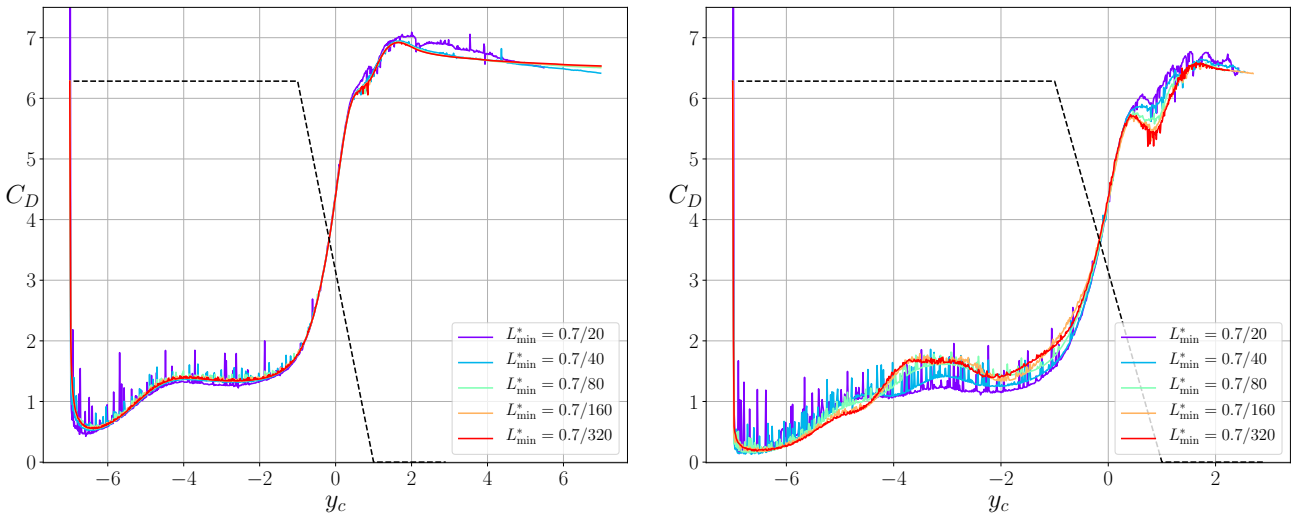
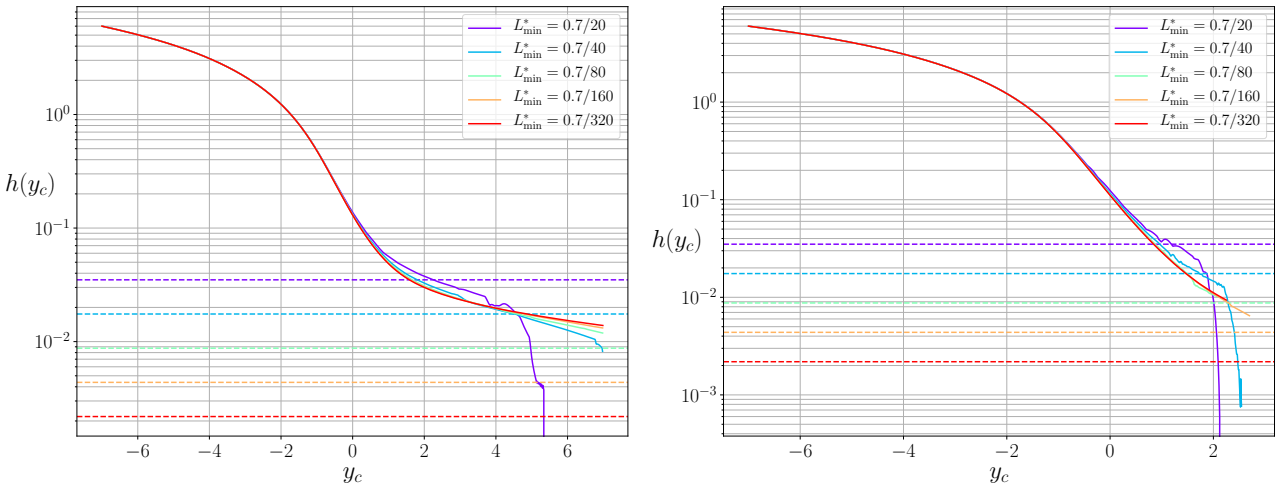


Figure 5.6: Boundary layer on the side of the rising cylinder (around an angle of $\frac{\pi}{2}$ from its apex). The coarse mesh (black lines) has a minimal mesh size $L^*_{\text{min}} = \frac{0.7}{40}$ at the wall, while that of the fine mesh (blue lines) is $L^*_{\text{min}} = \frac{0.7}{160}$. The color plot represents the vertical velocity component v for the most refined mesh.



(a) $Re_D = 1000$ **(b)** $Re_D = 10000$

Figure 5.7: Drag coefficient as a function of the cylinder position at Reynolds numbers 1000 (left) and 10000 (right) for different mesh resolutions. The dashed black line represents the evolution of the buoyancy force, assuming that the free surface does not deform.



(a) $Re_D = 1000$ **(b)** $Re_D = 10000$

Figure 5.8: Film thickness h above the rising cylinder as a function of the cylinder position for meshes with different minimal mesh sizes at Reynolds number 1000 (left) and 10000 (right). The corresponding resolution limits $L^*_{\min}$ are shown by the horizontal dashed lines.

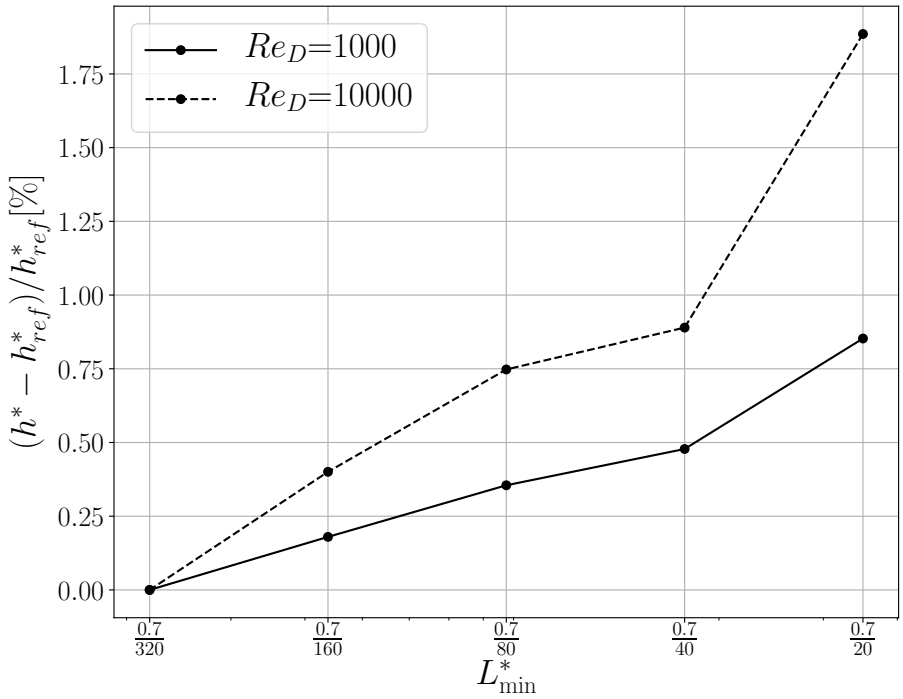


Figure 5.9: Percentage of variation of $h^* = h(t|_{y_c=-1})$ using different mesh resolutions with respect to the results obtained with the highest mesh resolution ($L^*_{\min} = 0.7/320$), for the two different Reynolds numbers.

5.2.1.2 Choice of the time step

A time step convergence analysis has been performed to determine the appropriate time step size. To do so, the minimal mesh size $L^*_{\min} = 0.7/160$ has been chosen, and only the case at $Re_D = 1000$ has been considered. It has been shown above that such level of spatial resolution was sufficient to get accurate results at that Reynolds number. Therefore, the focus is solely on the time step analysis. Because of the no-slip condition, it is known that higher mesh distortion occurs close to the cylinder surface, and that the time step should therefore scale as $\frac{L^*_{\min}}{U}$ (see Fig. 3.19 in Section 3.3). With $U = 1$ this yields $\Delta t \simeq 0.7/160$. However, since the boundary layer is resolved, a larger time step can possibly be used, again according to Fig. 3.19. Values smaller and larger than 0.7/160 are therefore considered, such that the time step size is in the following set:

$$\Delta t \in \left\{ \frac{0.7}{80}, \frac{0.7}{160}, \frac{0.7}{320}, \frac{0.7}{640} \right\} . \tag{5.6}$$

The time step analysis is performed by considering the film thickness $h(y_c)$ and the drag coefficient C_D. In Fig. 5.10, no difference is visible between the h curves obtained with the different time step sizes. Therefore, it seems that the largest time step is adequate. However, the use of larger time steps can cause problems when solving the nonlinear equations. Specifically, if the time step is too large, the nonlinear iterative algorithm cannot converge to the imposed tolerance. In this case, the algorithm automatically reduces the time step, as illustrated in Fig. 5.11 (Appendix D explains how the convergence of the nonlinear algorithm is assessed, and how the surface tension is computed). This issue is mainly due to the difficulty of achieving convergence of the surface tension force because it is highly dependent on the mesh geometry (as also discussed in Appendix D). In other words, similarly to the no-slip boundary condition at walls, the surface tension at deforming interfaces represents a limiting factor, and should thus also be taken into account in the choice of the time step size.

The results for the drag coefficient (Fig. 5.12) are again very similar for the different chosen time steps. However, a closer look (Figs. 5.12b and 5.12c) reveals that the drag coefficient is subject to some oscillations that are larger for smaller values of the time step size. This has already been pointed out by Cerquaglia et al. [65], who have shown that the spurious pressure oscillations frequently observed in the PFEM are inversely proportional to the time step size. Quite remarkably, the conclusion of this analysis seems to be that, to reduce pressure oscillations, finer meshes and larger time steps should be used. If this recommendation is extended to the context of mesh refinement, for similar pressure oscillation levels, the use of smaller mesh elements should enable the use of smaller time steps, but lowering too much the time step size for a similar mesh resolution would lead to larger pressure oscillations.

As a conclusion, the intuitive idea of using $\Delta t \simeq \frac{L^*_{\min}}{U}$ seems to provide a good estimate. Larger time steps would lead to larger mesh distortions and convergence issues, while too small time steps would lead to increased pressure oscillations. The idea is therefore to choose a value slightly smaller than $\frac{L^*_{\min}}{U}$, such that no convergence issues are encountered, while no large pressure oscillations are introduced.

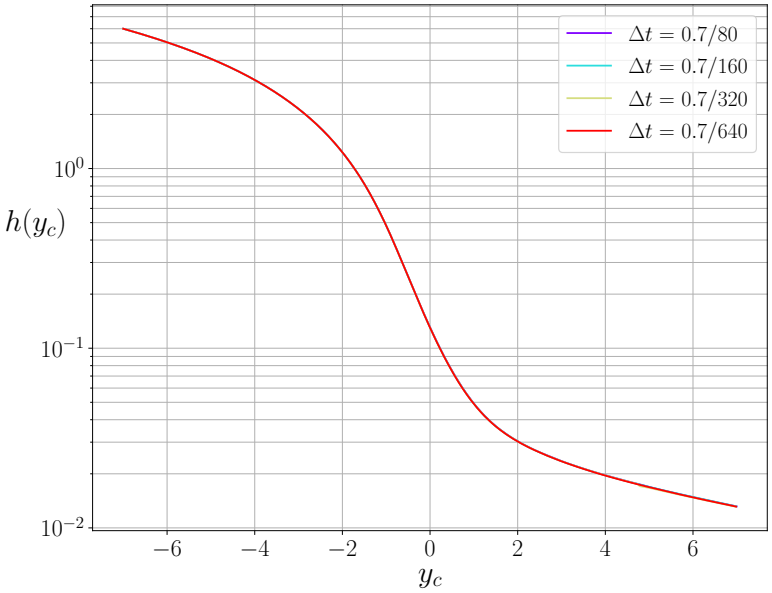


Figure 5.10: Film thickness h above the rising cylinder as a function of the cylinder position for the four different times steps at $Re_D = 1000$ and with $L^*_{\text{min}} = 0.7/160$. The curves are so close to each other that they are practically indistinguishable.

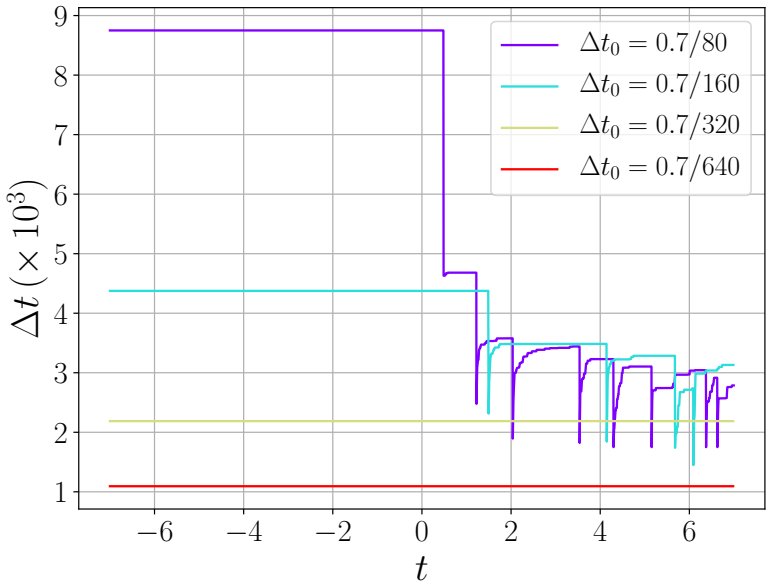
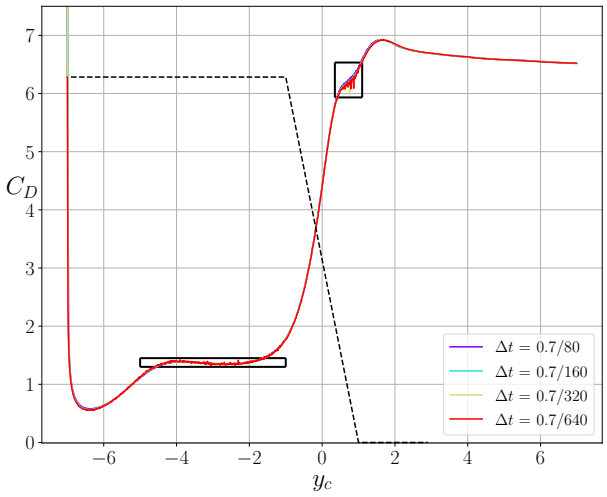
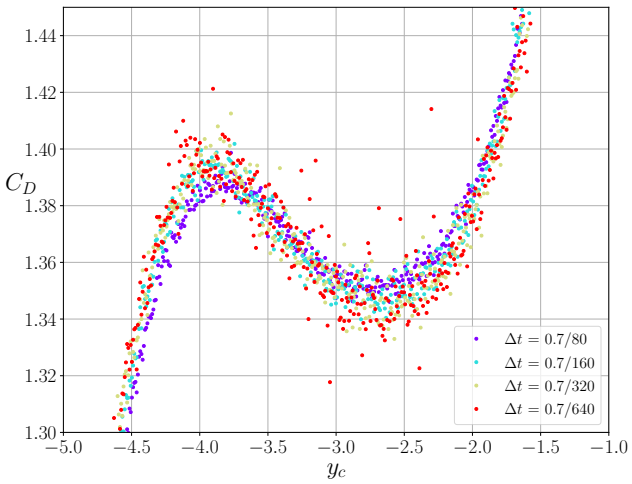


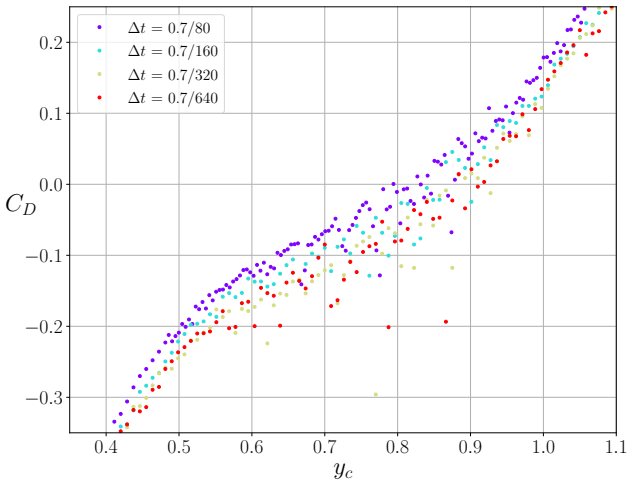
Figure 5.11: Reduction of the time step size over the simulation time because of convergence issues of the nonlinear iterative algorithm, for similar simulations at $Re_D = 1000$ with different initial time step sizes Δt_0 and $L^*_{\text{min}} = 0.7/160$.



(a)



(b)



(c)

Figure 5.12: Drag coefficient as a function of the cylinder position for different time step sizes at $Re_D = 1000$ with $L^*_{\min} = \frac{0.7}{160}$. **(a)** entire simulation and **(b,c)** closer view (corresponding to the black boxes in **(a)**) on the oscillations observed around $y_c = -3$ and $y_c = 0.75$.

5.2.2 Analysis of the different flow features

Using the parameters determined from the previous convergence analyses, the physical features of the flow around a rising cylinder are now investigated. The first feature of interest is the evolution of the surface deformation and film thickness h over the entire simulation, from the cylinder impulsive start to the development and drainage of the thin film around the cylinder. As discussed in the literature review, the flow physics is mostly controlled by the balance between inertia and gravity, quantified by the Froude number $Fr = \frac{U^2}{ga}$, where U is the cylinder velocity, g the gravity acceleration, and a the cylinder radius. Another important parameter is the starting cylinder depth d as it determines the development stage of the wake when the cylinder reaches the free surface. The free-surface deformation depends on all these aspects, as discussed in Section 5.2.2.1.

The other point of interest is the flow dynamics during the early stage, when the cylinder is still immersed. It includes the skin friction and the vorticity on the cylinder surface, the overall drag on the body and the wake dynamics (mainly evaluated through the shape of the streamlines). Here again, it is shown that the Froude number plays a significant role during the early phase dynamics, even if the Reynolds number, $Re_D = \frac{2Ua}{\nu}$, is the main parameter governing the wake dynamics (as in bounded flows). All these aspects are discussed in Section 5.2.2.2.

5.2.2.1 Free-surface surge and film thinning

First, the focus is on the amplitude of the free-surface deformation, quantified by the film thickness h above the rising cylinder, as defined in Fig. 5.1a. The variation of h with time t, and/or cylinder position $y_c = -d + Ut$, and its specific value $h^* = h(y_c = -1)$ when the top of the cylinder reaches the initial free surface, are both investigated. In this context, the flow is divided into two phases. The first phase corresponds to the surface surge, when the cylinder is below the initial free surface ($y_c < -1$). The second phase starts at the interface crossing ($y_c > -1$), after which the film $h(t)$ becomes thinner ($h(t) < 1$). For sufficiently long simulations, the film thinning is such that it reaches the viscous-gravity length scale l_ν,

$$l_\nu = \left(\frac{4Fr}{Re_D^2}\right)^{\frac{1}{3}}, \tag{5.7}$$

at which gravity and viscous forces are in balance [128] (see Appendix E for the derivation of this length scale). The transient regime from $h \simeq 1$ to $h \simeq l_\nu$ is discussed in Chapter 6, for which a dedicated model is presented. Note that all variables are non-dimensionalized using the cylinder radius a and velocity U (except the Reynolds number Re_D). As aforementioned, h and h^* depend mainly on the Froude number, larger Froude numbers leading, quite intuitively, to larger h and h^* (see Fig. 5.14). However, for a sufficiently small starting depth d, h and h^* also depend on d. Vincent et al. [123] have shown that, using a free-slip boundary condition at the cylinder surface (no wake), there is a threshold value $d_{\max}(Fr)$ above which h^* does not depend anymore on d. Their figure summarizing the findings of their parameter space exploration is reproduced in Fig. 5.13. It shows h^* as a function of the Froude number Fr and release depth d. The theshold $d_{\max}(Fr)$ is indicated by the curved continuous shadow-like line. For larger values of d, h^* only depends on the Froude number.

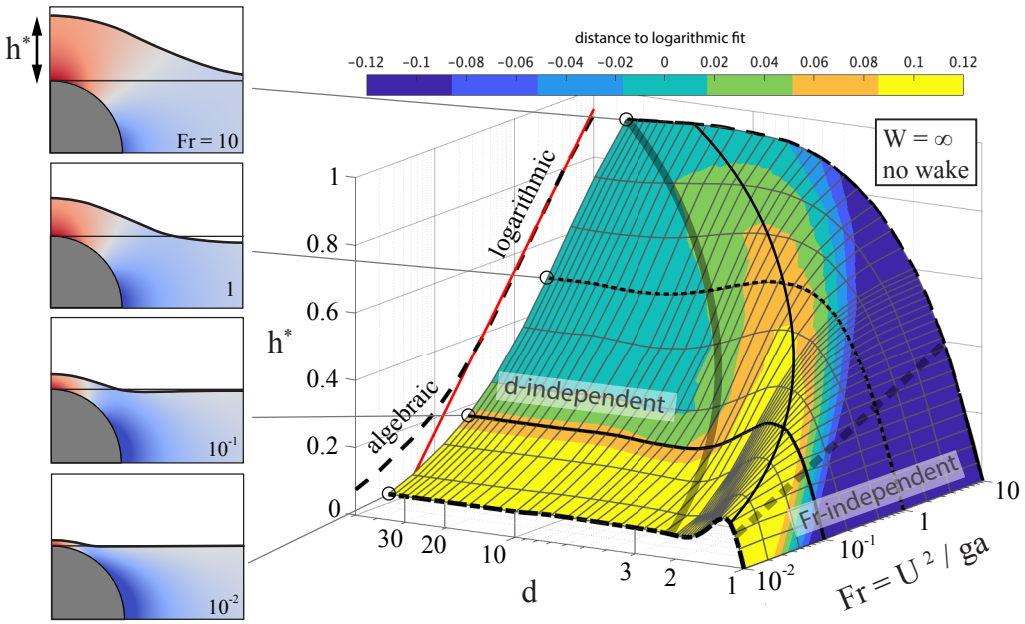


Figure 5.13: Characteristic bulge height h^* at $y_c = -1$ in slip conditions (no wake) for an unbounded domain (i.e., where the side walls are at infinity) as a function of the starting depth d and Froude number. h^* follows an algebraic trend for $Fr \leq 0.2$, and a logarithmic trend above; surface color indicates the distance to the best logarithmic fit (red line). A ridge betrays transient dynamics: the solid line marks the expected crest location found by matching travel time to $y_c = -1$ and expected time of occurence of the first surging peak. The thick diffuse lines mark the approximate boundaries of zones where h^* becomes independent of either d (solid line) or Fr (dashed line). Taken from Vincent et al. [123].

Configurations and simulation setup

As summarized in Table 5.4, three sets of simulations are considered, each corresponding to a different fluid. The first one, F1, is just a generic very viscous fluid. Only one velocity is considered so that the Reynolds number is fixed, while the Froude number is varied from 0.01 to 1 by changing the gravity acceleration. The two other sets of simulations use water and oil, respectively, for later comparison with in-house experiments. In this case, the Froude number is varied by changing the cylinder velocity so that the Reynolds number varies accordingly.

In the case of water, the mesh resolution has been kept moderate to limit the computational cost, although the corresponding Reynolds number is large. According to the previous mesh convergence analysis, the boundary layer in this case is not fully resolved. However, the viscous effects are small in the early stage of the surface elevation so that the resulting error should remain limited. Furthermore, the geometrical dimensions of the cylinder and tank are different for each set of simulations so as to assess the effect of confinement. According to the inviscid analysis of Vincent et al. [123], the starting depth d is larger than the threshold $d_{\max}$ for most cases, except for the water case at $Fr > 1$. Nonetheless, one should keep in mind that the presence of the wake in the present viscous simulations can introduce a stronger dependence on d. Finally, it should be noted that surface tension has been neglected for all cases, as its effect is negligible, at least until the free surface becomes very distorted with regions of strong curvature.

Impact of the Froude number on the surface deformation

The surge of the free surface, $h(y_c)$, is reported for the first set of simulations, F1, in Fig. 5.14a. As expected, h increases with Fr for a given cylinder position y_c. Note that the semi-log representation emphasizes the differences between the different curves at large y_c and reduce them at low y_c. In some region around $y_c \simeq 0$ for the five largest Froude numbers and around $y_c \simeq -1$ for the four smallest ones, the curves appear to be straight indicating an exponential decrease of the film thickness. In the next chapter, it is shown that this exponential decrease is mainly driven by a balance between gravity and inertial terms, consistently with the observations of Telste [111] and Liju et al. [124]. For larger values of y_c, when the cylinder emerges out of the bath, the film thinning rate decreases because the contribution of gravity to the pressure gradient inside the film slowly decreases, and the relative contribution of the convective terms increases. In addition, the viscous effects also become relatively more important as the film gets thinner, contributing also to the slowing down of the film thinning. Note that for smaller Froude numbers, the slope decrease takes place for smaller value of h, which simply indicates that the film is thinner at the time when the cylinder emerges out of the bath.

The same dependence on the Froude number is observed for h^*, as shown in Fig. 5.14b. Using simulations with a free-slip boundary condition (no wake), Vincent et al. [123] have shown that h^* is well approximated by

$$h^*(Fr) = \begin{cases} 0.6Fr^{0.45} & \text{for} \quad \mathrm{Fr} < 0.2 \\ 0.17\ln(Fr) + 0.55 & \text{for} \quad \mathrm{Fr} > 0.2\,. \end{cases} \tag{5.8}$$

However, it seems unlikely that the threshold value $Fr = 0.2$ has any physical justification, the behavior of the curves being very similar below and above $Fr = 0.2$. A better approximation of the dependency of h^* on Fr, which covers a larger range of Fr, can be obtained using the so-called Lambert $\mathcal{W}$ function, defined by its reciprocal as

$$\mathcal{W}^{-1}(x) \triangleq x\,\exp(x)\,. \tag{5.9}$$

With a fit of the form

$$h^*(Fr) = \mathcal{W}(a\,Fr^b)\,, \tag{5.10}$$

a and b being two calibration constants, we have

$$h^* \exp h^* = a\,Fr^b\,. \tag{5.11}$$

When h^* is small, which is the case when Fr is small, the exponential is close to 1 and

$$h^* \approx a\,Fr^b\,, \tag{5.12}$$

which is the form used by Vincent et al. [123] to approximate h^* for $Fr < 0.2$. On the other hand, taking the logarithm of Eq. (5.11) and putting h^* in evidence yields

$$h^* \left(1 + \frac{\ln(h^*)}{h^*}\right) = \ln(a) + b \ln Fr\,. \tag{5.13}$$

In the limit of large h^* ($h^* \gg 1$), the expression in parenthesis on the left-hand side approaches 1 such that one has approximately

$$h^* \approx \ln(a) + b \ln Fr\,, \tag{5.14}$$

which is the form proposed by Vincent et al. [123] for the fit with $Fr > 0.2$. Therefore the Lambert $\mathcal{W}$ function allows a smooth transition from an algebraic to a logarithmic behavior, as somehow suggested by its other name in the literature: the logarithm product.

For the simulation set F1 in Table 5.4, such a fit (obtained using a non-linear least square technique) is illustrated in Fig. 5.14b, where $a = 1.2$ and $b = 0.6$. We observe that Eq. (5.12) provides a very good approximation of $h^*(Fr)$. The present calibration constants are interestingly quite different from those in Eq. (5.8), explaining the deviation observed in Fig. 5.14b from the present results. The main reason is that the present PFEM simulations include viscous effects, such that h^* is increased by the presence of the wake (which leads to a larger zone of high pressure above the cylinder), while the results of Vincent et al. [123] have been obtained using a free-slip boundary condition at the cylinder surface (no wake). Unfortunately, it also means that the fit not only depends on the Reynolds number, but also on the starting depth since it may impact the wake length. Note that all simulations of the F1 set are at the same Reynolds number, therefore resulting in the same wake dynamics as illustrated in Fig. 5.15. As a consequence, the curves obtained in Fig. 5.14 remains meaningful but using another Reynolds number or another starting depth may require a different calibration.

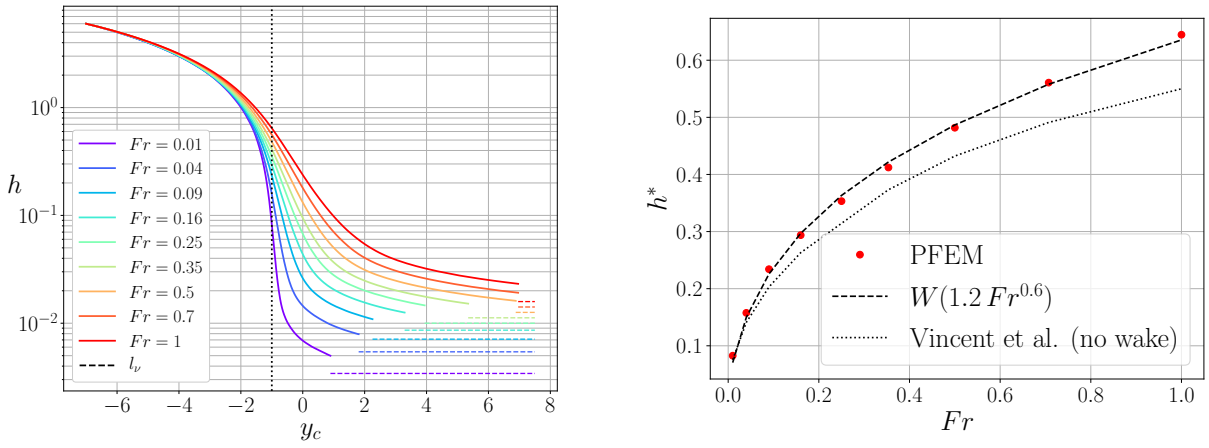
Nevertheless, ignoring the influence of the wake, the fact that a unique fit with two parameters is enough to describe $h^*(Fr)$, is a remarkable result. Furthermore, it is shown in the next chapter that the Lambert $\mathcal{W}$ function is a great tool that can help in the modeling of the film dynamics.

			F1	Water	Oil
Cylinder diameter	D	m	2	0.05	0.05
Cylinder velocity	U	m/s	1.	[0.2 - 1]	[0.099 - 1.162]
Acceleration of gravity	g	m/s^2	$1/Fr$	9.81	9.81
Kinematic viscosity	ν	mm^2/s	2000	1	50
Tank half width	W	-	21	22	6
Tank height	H	-	28	29	28.8
Initial depth of the cylinder	d	-	7	7	19.8
Reynolds number	Re_D	-	1000	[10000 - 50 000]	[99-1162]
Froude number	Fr	-	[0.01-1]	[0.16-4]	[0.04-5.5]
Minimal mesh size	$L^*_{\min}$	-	0.003	0.004	0.003
Maximal mesh size	$L^*_{\max}$	-	1.2	1.2	0.8
Time step size	Δt	-	0.0033	0.0025	0.0033

Table 5.4: Simulation parameters for the cylinder rising towards a free surface at constant velocity. Three sets of simulations are considered, each with a different fluid. All non-dimensional quantities have been non-dimensionalized using the cylinder radius a and velocity U, except for the Reynolds number which uses the cylinder diameter D instead of a, to be consistent with the literature.

Influence of the bath half width W and starting depth d, on the intial surge speed c^*

Another important aspect is the impact of lateral confinement and starting depth of the cylinder. Based again on simulations with a free-slip boundary condition, Vincent et al. [123] have found correlations which link the initial (non-dimensional) surge speed c^*, i.e., the vertical velocity at $t = 0$ of the free-surface point vertically above the cylinder, with the starting depth d, as illustrated in Fig. 5.16. In particular, they have identified two regimes:



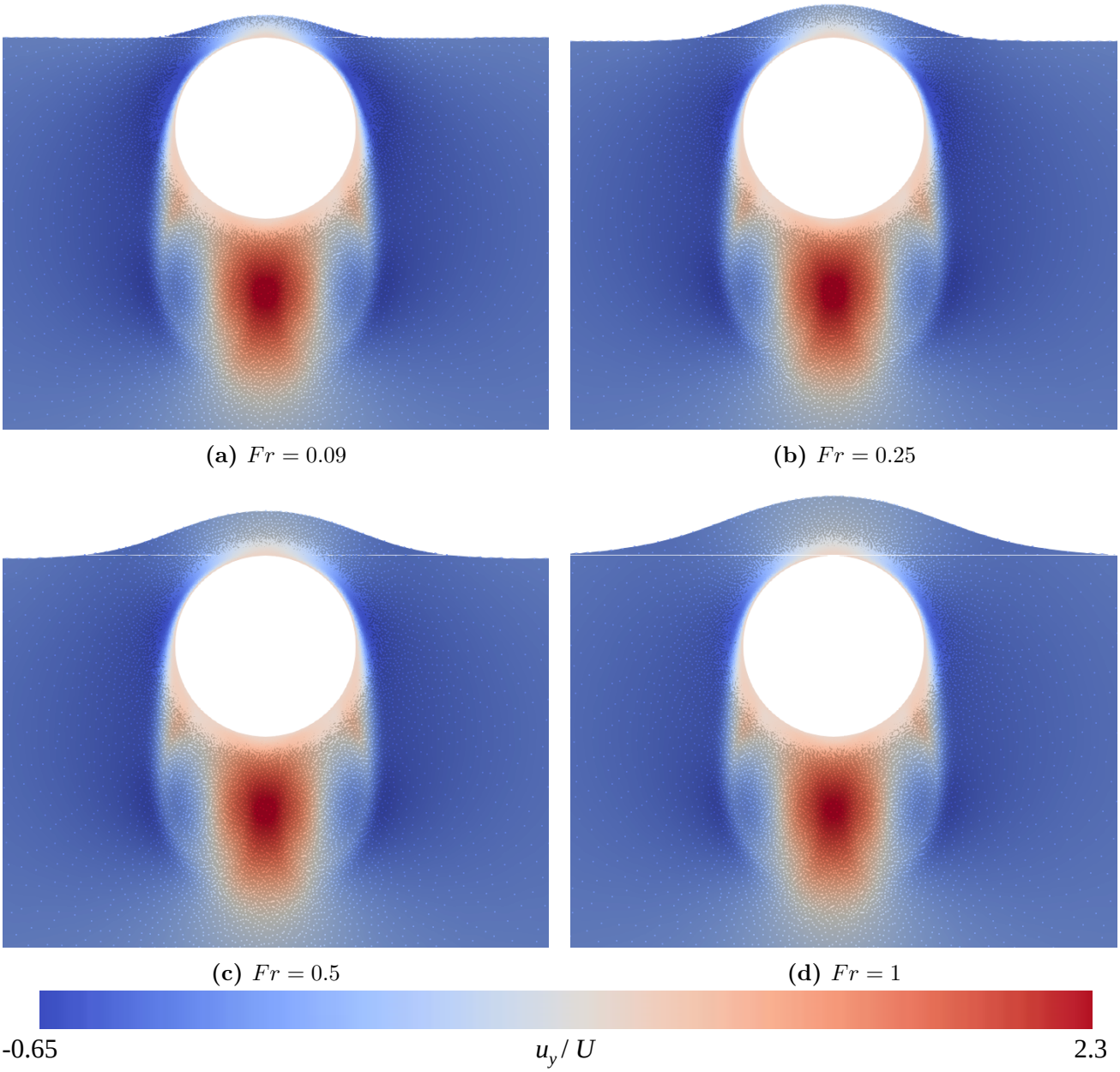
(a) **(b)**

Figure 5.14:
(a) Evolution of h as a function of the cylinder position y_c for different Froude numbers, $Re_D = 1000$ and for a starting depth $d = 7$ (case F1 in Table 5.4). The horizontal dashed lines indicate the corresponding viscous-gravity length scale l_ν defined in Eq. (5.7). The vertical dotted line highlights the cylinder position $y_c = -1$ where h^* is defined.
(b) Corresponding variation of $h^* = h(y_c = -1)$ with respect to the Froude number.The red symbols correspond to the simulation results, the dashed line to the regression using the Lambert $\mathcal{W}$ function according to Eq. (5.10), and the dotted line the prediction by vincent et al. [123] given by Eq. (5.8) in the absence of a wake.

- In the unbounded regime, for $\frac{d}{W} < 1/2$, the initial surge speed c^* is well approximated by $\frac{2}{d^2}$.
- In the laterally confined regime, for $\frac{d}{W} > 1/2$, the initial surge speed c^* is well approximated by $\frac{20}{W^2}\mathrm{e}^{-3.14\frac{d}{W}}$.

These correlations are evaluated for selected simulations of Table 5.4. The first two sets of simulations ("F1" and "Water") fall in the case of the unconfined regime, as $\frac{d}{W} \simeq \frac{1}{3} < \frac{1}{2}$, and an initial surge speed approximately equal to $\frac{2}{d^2}$ should be recovered. On the other hand, the results for the oil case fall into the second category. The results are listed in Table 5.5, where the deviation from the predictions using the confined and unconfined correlations is reported. For each series of simulations, the error is within 10% of the predicted values, which is satisfying considering the no-slip boundary condition imposed here compared to the free-slip condition used by Vincent et al. to obtain the correlations.[3]

[3]So far, the PFEM code does not include free-slip/stress-free boundary conditions on curved surfaces. The main difficulty arises from the Lagrangian nature of the method, where additional implementation is required, for instance to correct the node position after they slide on a curved boundary, or to add kinematic constraints. Such developments may be considered for future work, and ideas are presented in the perspectives in Chapter 7.



(a) $Fr = 0.09$ **(b)** $Fr = 0.25$

(c) $Fr = 0.5$ **(d)** $Fr = 1$

Figure 5.15: Surface elevation at the moment when the top of the cylinder reaches the initial free surface ($y_c = -1$), for a Reynolds number $Re_D = 1000$ and different Froude numbers(case F1 in Table 5.4). The contour plot illustrates the vertical component of the velocity field.

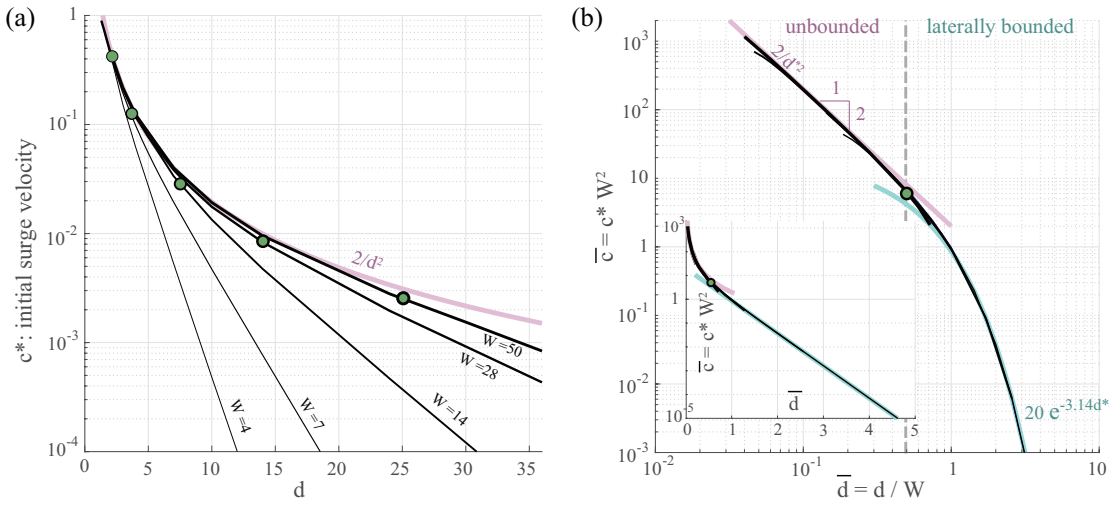


Figure 5.16: Initial surge velocity c^* : (**a**) raw and (**b**) renormalised data in log-log scale (main) and linear-log scale (inset). At values of $\overline{d} = d/W$ larger than 1/2, the surge is substantially altered by lateral confinement, and the dependence in $\overline{d}$ changes from an algebraic relationship (small $\overline{d}$) to an exponential decay (large $\overline{d}$). The initial surge speed is independent of Fr. Taken from Vincent et al. [123].

	F1	Water	Oil
Initial depth of the cylinder d	7	7	19.8
Tank half width W	21	22	6
d/W	1/3	0.3182	3.3
Confined	No	No	Yes
Initial surge speed (theoretical) c^*_{th}	0.0408	0.0408	1.75×10^{-5}
Average initial surge speed (simulations) c^*_{sim}	0.0367	0.0379	1.91×10^{-5}
Standard deviation of c^*_{sim}	2.6×10^{-5}	3.2×10^{-4}	1.4×10^{-7}
% of error on c^*	10%	7.1%	9.1%

Table 5.5: Initial surge speed c^* of the impulsively started gravity wave, for different simulations. The values have been obtained by averaging the results at the different Froude numbers, with a standard deviation more than 100 times smaller, which clearly indicates that this initial velocity is independent of the Froude number. In each case, the deviation of the initial surge speed remains within 10% of the correlations proposed by Vincent et al. [123].

Impact of the wake

The free-surface deformation depends also indirectly on the starting depth d through the cylinder wake. As shown by Vincent et al. [123], a longer wake induces a larger free-surface deformation. In particular, by comparing simulations using a no-slip condition at the cylinder surface to a simulation with a stress-free condition (i.e., without wake), they observed an additional increment δ_{wake} of the surface deformation h^* that increases with the starting depth d (see Fig. 5.17 left). To quantify the wake size, they introduced the so-called jet area j as the area of the flow region downstream of the cylinder for which the vertical flow velocity component $v > U$, and $j^* = j(y_c = -1)$ the value of j when the cylinder reaches the free surface (see Fig. 5.17 middle and Fig. 5.18 left). As shown in Fig. 5.17 right, $\delta_{\text{wake}}/\sqrt{j^*}$ depends only on the Froude number and not on the starting depth, as long as $Fr \leq d/3$, i.e., as long as the wake as sufficient time to develop without interference from the free surface. Furthermore, for $Fr > d/3$, the ratio $\delta_{\text{wake}}/\sqrt{j^*}$ seems to reach a constant value that is independent of Fr. Vincent et al. [123] also showed that the jet size j increases during the cylinder rise following an algebraic law whose exponent slightly varies with the Reynolds number but is independent of the Froude number (see Fig. 5.18).

To compare the present results with those of Vincent et al. [123] in terms of the impact of the wake on the surface elevation, their relations given by Eq. (5.8) has been subtracted from the present values of h^* to get the increment δ_{wake}. The results are presented in Fig. 5.19 for the case $d = 7$.

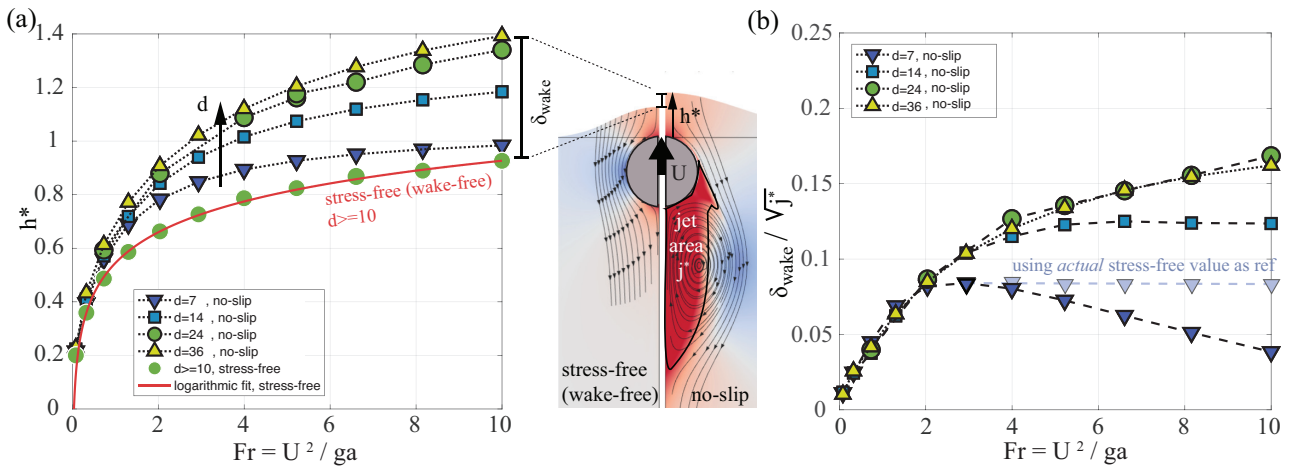


Figure 5.17: Wake-induced surge excess. (**a**) Characteristic bulge height h^* for various Froude numbers and starting depths, with and without wake. (**b**) Wake-induced surge normalised by $\sqrt{j^*}$, where j^* is the characteristic jet area. Deeper starting depths induces larger wakes at the time of crossing, which in turn promotes additional swelling δ_{wake} of the water layer above the cylinder. $W = 28$, $H = 48$, $Re_D = 10\,000$. Taken from Vincent et al. [123].

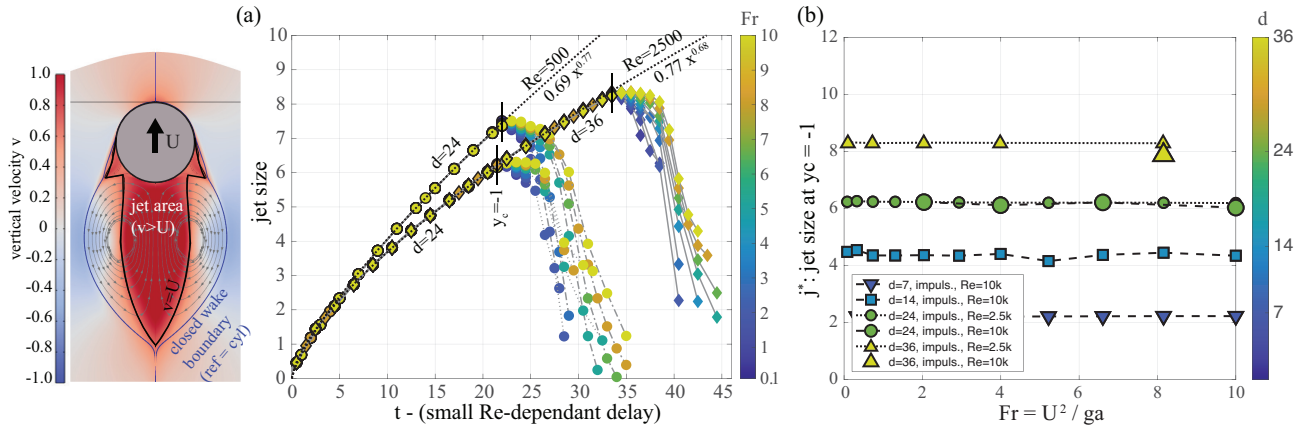


Figure 5.18: Wake size and jet area (no-slip). (**a**) Time evolution of jet area and for various Fr (color bar) and d, and (**b**) characteristic value j^* at $y_c = -1$ for various d, Re_D and Fr. The jet area, proportional to the wake size, undergoes an algebraic, Fr-independent growth until the cylinder crosses the interface, followed by a quick Fr-dependant decay. The growth and the characteristic wake size both exhibit a slight dependance in Re_D. Taken from Vincent et al. [123]. Note that the wake only starts to grow after a small Re_D-dependent time delay, which was estimated, based on (**a**), to 1 and 1.5 for $Re_D = 500$ and $Re_D = 2500$, respectively.

For the renormalization, the wake area j^* at the time when $y_c = -1$ is required. To compute this, the growing time of the wake during the rising phase is required, the latter being equal to $t^- = d - 1 - \delta t(Re_D)$, $\delta t(Re_D)$ being the time delay required for the wake to start growing. In the present case, it has been estimated based on Fig. 5.18(**a**), where a linear interpolation on the Reynolds number enables to find $\delta t = 1.125$ for $Re_D = 1000$. The delayed time t^- is therefore equal to $d-1-\delta_t \simeq= 4.875$. Injecting this into the relations of Fig.(**a**) (both relations give similar results for this value of t^-), a value of $j^* \simeq 2.3$ is found, which seems to be in agreement with the results shown in Fig. 5.18(**b**) for a release depth $d = 7$, despite the use of different Reynolds numbers.

In Figure 5.19, the agreement between the present results and those of Vincent et al. [123] is quite satisfying, which demonstrates that the present approach correctly captures the impact of the wake on the free-surface elevation. Quite interestingly, the increment δ_{wake} seems to follow a different trend at very low Froude number ($Fr < 0.1$). In particular, δ_{wake} obtained from the present simulations is larger than expected. This was not pointed out by Vincent et al., possibly because of their coarser sampling of the Froude number range below $Fr = 1$.

These results are very useful for modeling the surge dynamics. In particular, they suggest that, for a sufficiently deep starting position, the surface elevation h^* can be modeled as an "inviscid" contribution

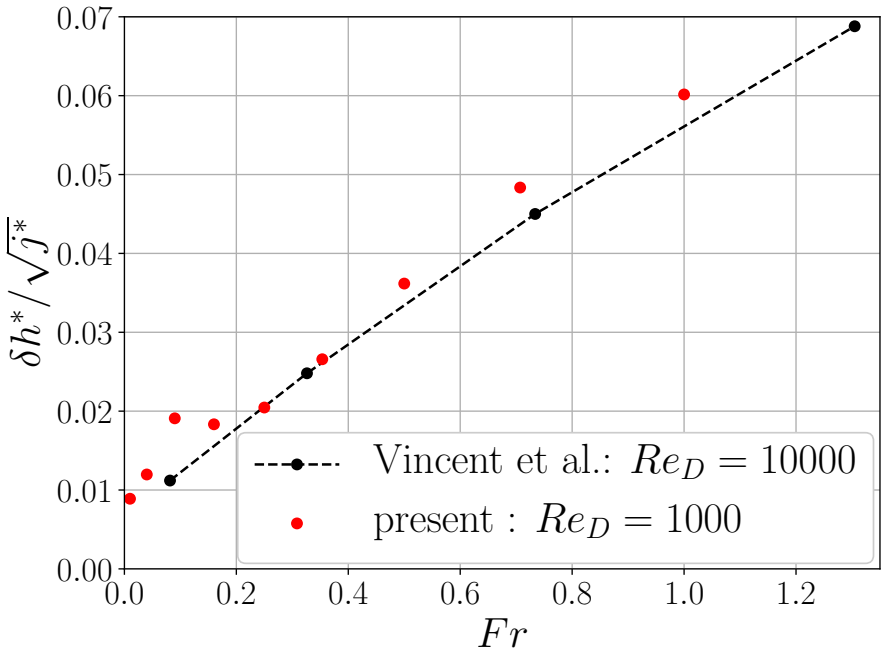


Figure 5.19: Variation of the free-surface elevation increment δ_{wake} with respect to the Froude number. Present results and results of Vincent et al. [123] are shown for an initial depth $d = 7$. For the present case, a renormalizing wake area $j^* \simeq 2.3$ is used.

that depends only on the Froude number plus a correction δ_{wake} that takes into account the effect of the wake. This wake correction non-dimensionalized by the wake size (jet area j^*) when the cylinder reaches the free surface also depends only on the Froude number, while the wake size itself depends on the initial starting depth d and, weakly, on the Reynolds number, but not on the Froude number.

5.2.2.2 Drag, skin friction, vorticity and wake dynamics

In this section, aspects associated with the boundary layer and wake of the cylinder are further investigated, including overall drag, surface friction and vorticity, and streamlines in the wake. Two different sets of simulations are considered, whose parameters are given in Table 5.6. The first set is at constant Froude number and the second set at constant Reynolds number.

In addition, a simulation of a cylinder rising towards an impermeable but inviscid horizontal wall is performed to investigate the limiting case of a very low Froude number, for which the free surface does almost not deform. The same parameters as in the simulation set 2 are used in this case.

Numerical setup

For both series of simulations, a constant velocity and a no-slip boundary condition are imposed at the cylinder surface, while a free-slip boundary condition is imposed at the tank walls. For the first set of simulations, two geometrical refinement criteria are used. The first one interpolates L^* from $\frac{L^*_{\text{max}}}{10}$ at a distance 22/3 from the vertical center line to L^*_{max} at the walls. The other geometrical criterion interpolates from L^*_{min} at the cylinder surface to L^*_{max} at a distance of 22 from the cylinder center. In addition, the velocity-gradient based criterion given by Eq. (3.23) is used with the parameters of Table 5.6. Note that the threshold values of the target mesh size for the solution-based criterion are denoted $L^*_{\text{min,SOL}}$ and $L^*_{\text{max,SOL}}$ to differentiate them from L^*_{min} and L^*_{max} used for the geometry-based criteria.

For the second set of simulations, only one geometrical criterion is used, interpolating L^* from L^*_{min} at the cylinder surface to L^*_{max} at a distance of 22 from the cylinder center. The solution-based criterion for mesh refinement is also used but with different parameters, as summarized in Table 5.6.

		Set n° 1	Set n° 2
Tank half width	W	22	21
Tank height	H	28	28
Initial depth of the cylinder	d	6	7
Froude number	Fr	$\{0.5, 10\}$[4]	$\{0.25, 0.5, 1, 2\}$
Reynolds number	Re_D	$\{550, 3000, 9500\}$	1000
Min. mesh size	L^*_{min}	0.003 or[5] 0.0015	0.002
Max. mesh size	L^*_{max}	1.2	1.2
Time step size	Δt	0.0025 or 0.00125	0.0033
Power parameter	β	1/3	1/3
Max. velocity gradient norm	$\|\|\nabla\mathbf{u}\|\|_{\text{max}}\|\|$	2	2
Min. velocity gradient norm	$\|\|\nabla\mathbf{u}\|\|_{\text{min}}\|\|$	0.03125	0.016
Min. mesh size for SOL	$L^*_{\text{min,SOL}}$	0.014	0.014
Max. mesh size for SOL	$L^*_{\text{max,SOL}}$	0.14	0.186

Table 5.6: Geometrical, physical, mesh and mesh refinement parameters of two different sets of simulations. All quantities are non-dimentionalized by the cylinder velocity U and its radius a, except the Reynolds number which uses the cylinder diameter as a characteristic length. The solution-based mesh refinement parameters are used in Eq. (3.23).

Vorticity and skin friction at the cylinder surface

For the simulations at a Froude number of 0.5 (simulation set 1 in Table 5.6), the wall vorticity[6] at the cylinder surface is plotted for $Re_D = 550$, 3000 and 9500 in Fig. 5.20, at different times after an impulsive start. The black curves are the results for an impulsively started cylinder without free surface obtained by Koumoutsakos et al. [93] using a vortex method with around 300000 vortex elements. Despite the lower number of nodes in our simulations, i.e., from 35000 to 55000 for $Re_D = 550$ and from 60000 to 80000 for $Re_D \geq 3000$, the present results are relatively close to those of Koumoutsakos et al. in the early stage, owing to the small mesh size near the cylinder surface and in its wake. For later times, the cylinder approaches the free surface and some deviations from Koumoutsakos results can be observed. These discrepancies can be attributed in part to the lower mesh resolution, but mostly to the presence of the free surface in the present simulations. More specifically, the increasing weight of the liquid above the cylinder, due to the surface elevation, tends to increase the velocity tangential to the cylinder surface, which increases the skin friction and the vorticity at the wall. Note that the increasing velocity above the cylinder is clearly observable in analyses introduced in Chapter 6, and is mainly found to be a function of the Froude number. We can also observe that the flow complexity increases with time: the vorticity changes sign at several locations along the cylinder surface, and more vortex shedding events take place. This had already been discussed in Section 4.1.4. As expected, this trend is even more prominent and occurs more rapidly when the Reynolds number increases.

To further investigate the impact of gravity on the boundary layer, the skin friction coefficient c_f for the different Froude numbers at a constant $Re_D = 1000$ (simulation set 2 in Table 5.6) is plotted for different cylinder positions y_c in Fig. 5.21, together with the results obtained for the simulation of the cylinder rising toward a wall (only for $y_c < -1$).

While the results are very similar at a depth $y_c = -4.5$, the curves progressively differ from each other as the cylinder approaches the free surface, the skin friction being larger for smaller Froude numbers. In other words, while the free surface deforms less at smaller Froude numbers, this smaller deformation has a stronger impact on the flow around the cylinder, like if the cylinder was approaching a wall. In particular, in the limit of very small Froude numbers, the free surface should not move at all, provided that the friction effects remains negligible on the overall film dynamics, which should be the case if h is not too close to 0. This is confirmed by the simulation of the cylinder rising towards the fixed wall with

[4]This series also includes simulations in a bounded domain (i.e., a domain without deforming interfaces), where the Froude number is irrelevant.

[5]The resolution L^*_{min}=0.003 is used for $Re_D = 550$ only.

[6]See Appendix F for the relation between surface vorticity and friction coefficient.

free slip boundary condition. At $y_c = -1.5$, the maximum values of the skin friction increases as the Froude number decreases, and is found at a smaller angle θ, closer to the top of the cylinder. To further support this argument, the contour of the horizontal component of the velocity is shown in Fig. 5.22 for the wall case and the case at $Fr = 0.25$. These similarities between the low Froude number simulations and the simulation of the cylinder approaching a wall have already been observed by Telste [111].

It is important to note that, while there are clear differences across Froude numbers for the friction on the upper part of the cylinder ($|\theta| < \pi/2$), c_f is almost identical for all Froude numbers along the bottom surface of the cylinder ($|\theta| > \pi/2$), at least as long as the cylinder remains immersed ($y_c \leq -1.5$). This suggests that the free surface has almost no impact on the wake dynamics. The same vortex shedding process, with separations and reattachments of the boundary layer, occurs during the cylinder rise toward the free surface. After the interface crossing, the impact of the Froude number becomes visible. Separation only occurs close to the cylinder bottom, where the thin film draining along the cylinder surface separates, as shown in Fig. 5.23. Interestingly, at small Froude number, large oscillations of the skin friction coefficient are observed along the bottom part of the cylinder (see Fig. 5.21, $Fr = 0.25$, $y_c = 2$, $|\theta| > \pi/2$). These oscillations are due to gravitational instabilities in the film, because the acceleration of gravity points from the free surface toward the outside of the fluid. These developing instabilities of the film are clearly visible for $Fr = 0.25$ in Fig. 5.23 (right). At $Fr = 0.5$, the film seems to be stable, but small oscillations in the skin friction coefficient suggest that the instability could develop. Note that this instability could be avoided if surface tension was taken into account, which is not the case in the present simulation, mainly to achieve an easier convergence of the nonlinear algorithm.

Finally, it should be mentioned that the small amplitude oscillations of the skin friction coefficient on the upper part of the cylinder at $Fr = 0.25$ and $y_c \geq 1$ are mostly due to a mesh in the film that is slightly too coarse. The need for a higher resolution at small Froude number is due to the corresponding smaller viscous to gravity length scale $L_\nu/a = \sqrt[3]{\frac{4FR}{Re_D}}$. The correspondingly larger contribution of the gravity acceleration leads to a larger draining velocity of the film and thus a more rapid film thinning. This illustrates that, when prescribing the mesh resolution in the film, not only the Reynolds number, but also the Froude number, should be considered.

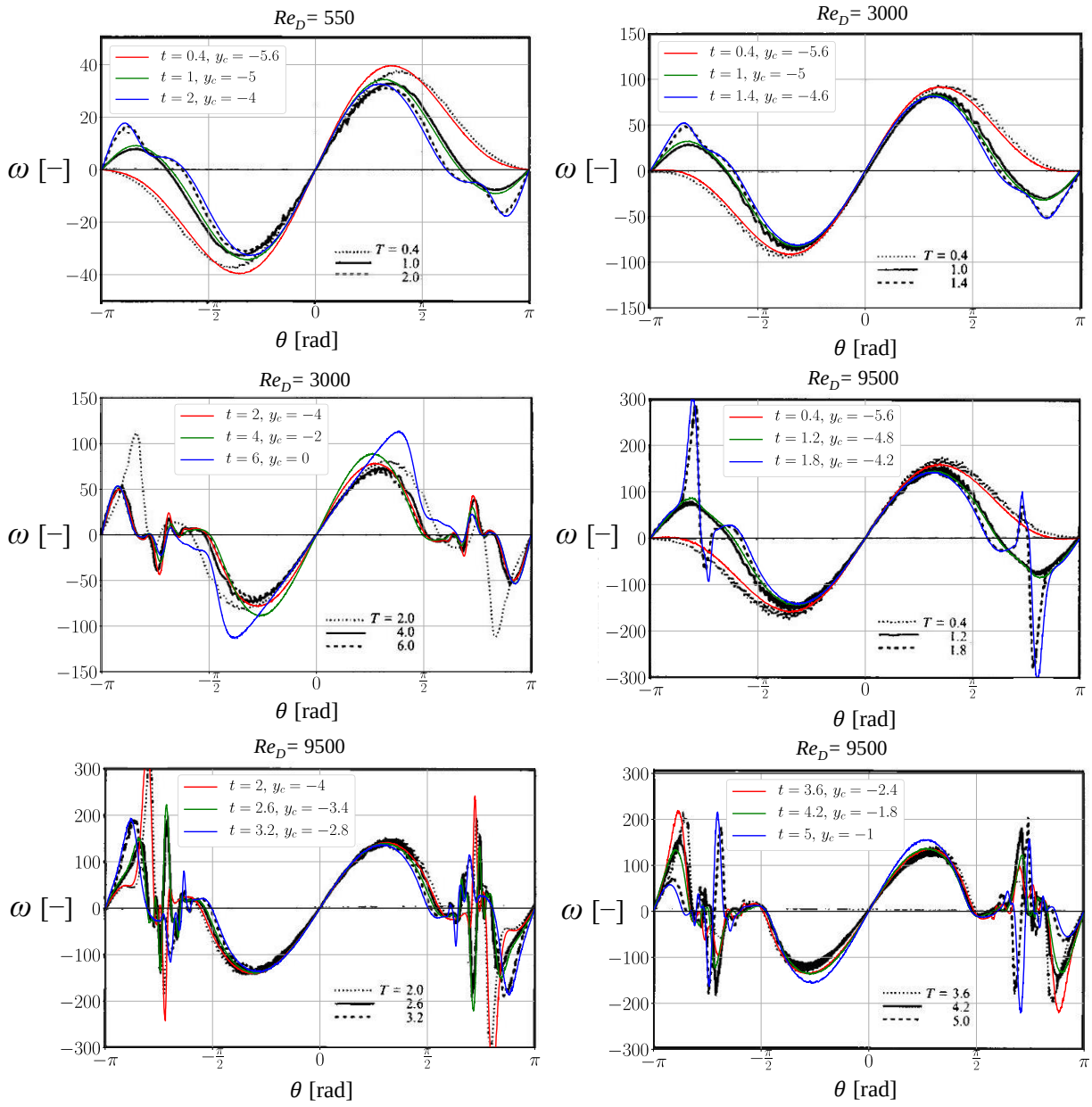


Figure 5.20: Non-dimensional vorticity ω as a function of the position angle θ along the cylinder surface at different time instants after the impulsive start ($Fr = 0.5$, simulation set 1 in Table 5.6). The cylinder top corresponds to $\theta = 0$. The results of Koumoutsakos et al. [93] for an impulsively started cylinder without free surface are shown as black lines.

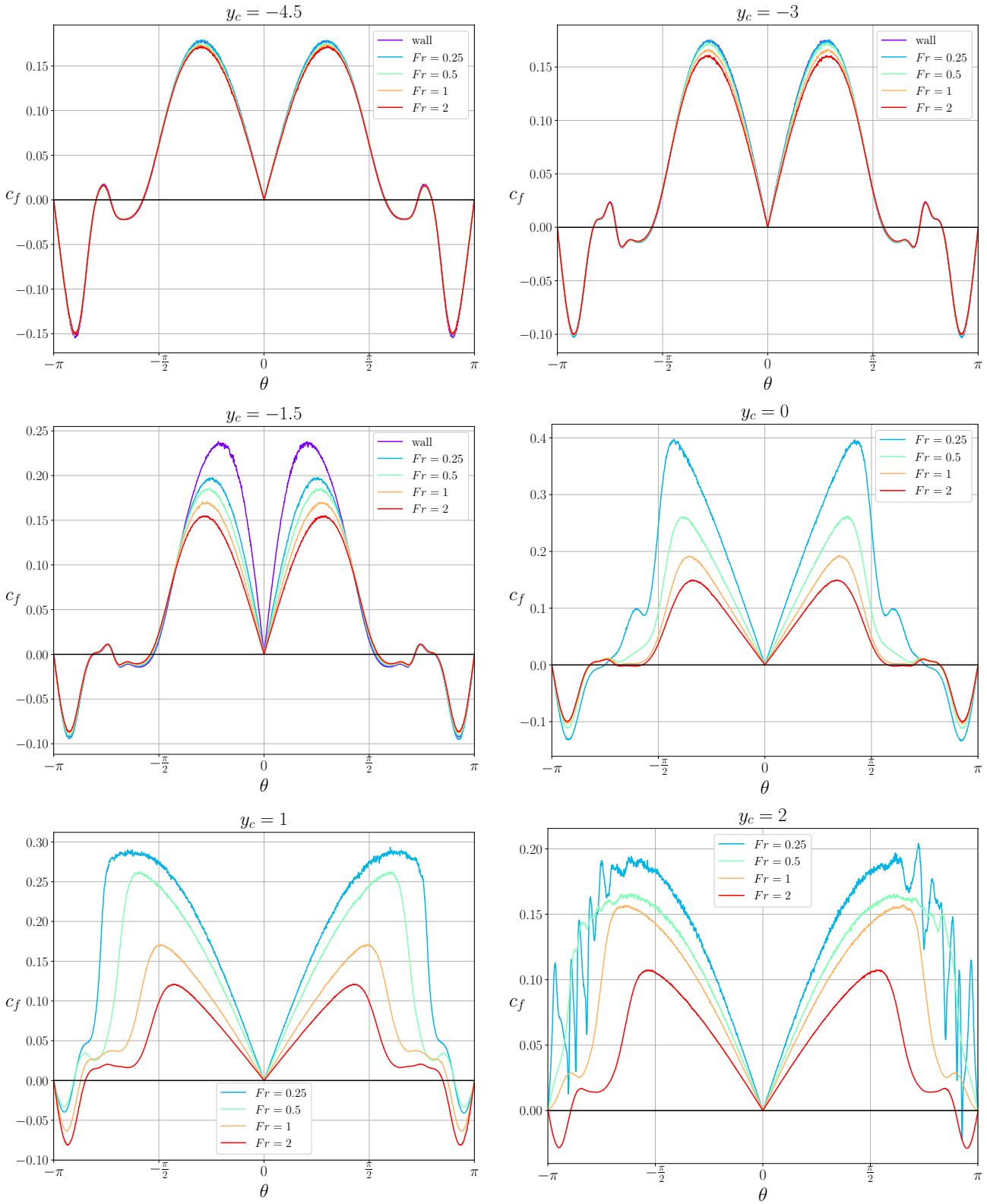


Figure 5.21: Skin friction coefficient on the cylinder surface at different positions $y_c = \{-4.5, -3, -1.5, 0, 1, 2\}$ for different Froude numbers ($Re_D = 1000$, simulation set 2 in Table 5.6) and for the case of a cylinder rising towards a wall ($y_c \leq -1.5$ only). The top of the cylinder corresponds to the value $\theta = 0$.

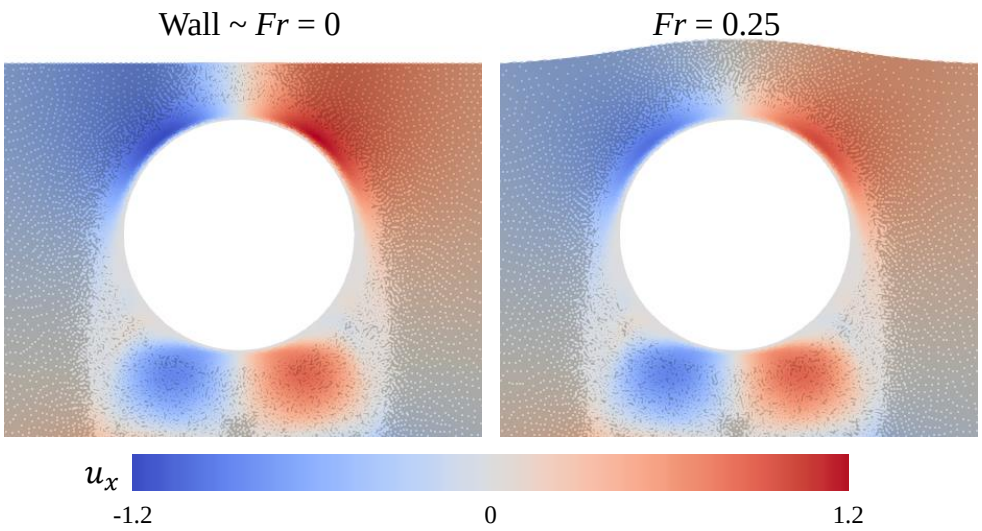


Figure 5.22: Contour of the horizontal component of the velocity at $y_c = -1.5$ for the cylinder rising toward a wall ($Fr \to 0$, left), and the cylinder rising toward the free surface at $Fr = 0.25$ (right), both at $Re_D = 1000$.

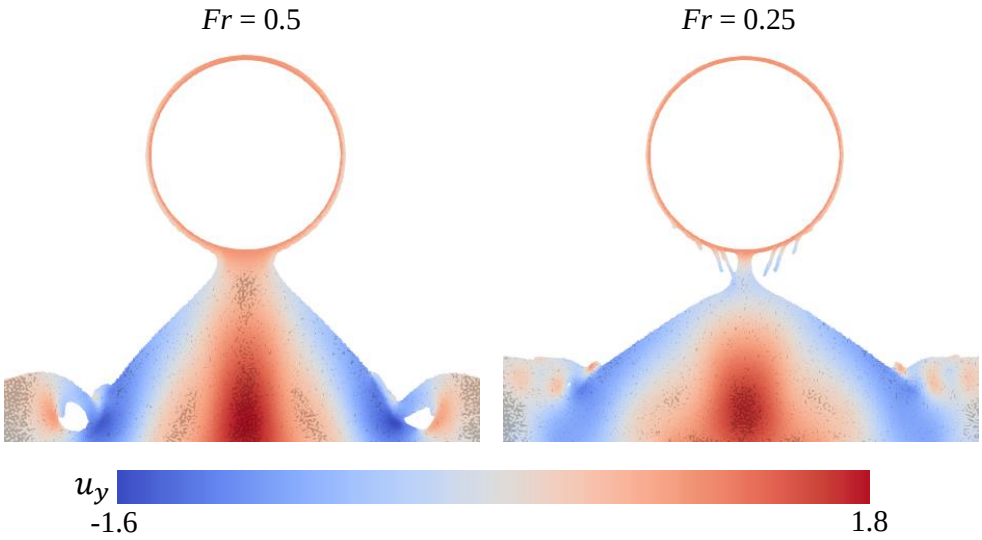


Figure 5.23: Contour of the vertical component of the velocity at $y_c = 2$ (after the free surface crossing) for the cylinder rising toward the free surface at $Fr = 0.5$ (left) and $Fr = 0.25$ (right), both at $Re_D = 1000$.

Overall cylinder drag in the immersed phase.

The evolution of the overall drag acting on the impulsively started cylinder rising towards the free surface ($y_c \leq -1$) is now analyzed for $Re_D = 550$, 3000 and 9500 (simulation set 1 in Table 5.6). To further assess the impact of the free surface, simulations are performed at a low ($Fr = 0.5$) and high ($Fr = 10$) Froude number and compared to simulations in a bounded domain (i.e., without free surface). The corresponding drag coefficient C_D is shown in Figs. 5.24 and 5.25 and shows two opposite trends. The results in a bounded domain are also compared to those of Koumoutsakos et al. [93], as already discussed in Section 4.1.4.

For the low Froude number (Fig. 5.24), the drag coefficient C_D is systematically larger for the free-surface case (continuous blue lines) than in the bounded domain (dashed blue lines). Moreover, the "drag excess" (continuous red lines) increases as the cylinder approaches the free surface. Interestingly, the qualitative behavior of C_D is nevertheless the same in both cases until shortly before $y_c = -1$ at which point C_D rapidly increases in the free-surface case, while it decreases in the bounded-domain case (note that this behavior might depend on the starting depth).

While the drag coefficient in a bounded domain at $Re_D = 550$ is very close to that of Koumoustakos et al. (dashed green lines), discrepancies appear at $Re_D = 3000$ and are even more prominent at $Re_D = 9500$. They are mainly due to the limited mesh resolution in the wake. However, both drag curves seem to be on the same level of magnitude, such that, regardless of temporal oscillations, the global scaling is correctly captured. Moreover, because the drag excess is assessed on the same PFEM mesh, it can be expected that a better mesh resolution would change the individual curves but only slightly the drag excess itself.

On the other hand, for the large Froude number (Fig. 5.25), the drag coefficient in the free-surface case is systematically lower than in the bounded domain, despite again the same qualitative behavior of the two curves. In other words, while the free surface induces a drag excess at small Froude number, it leads to a "drag deficit" at large Fr.

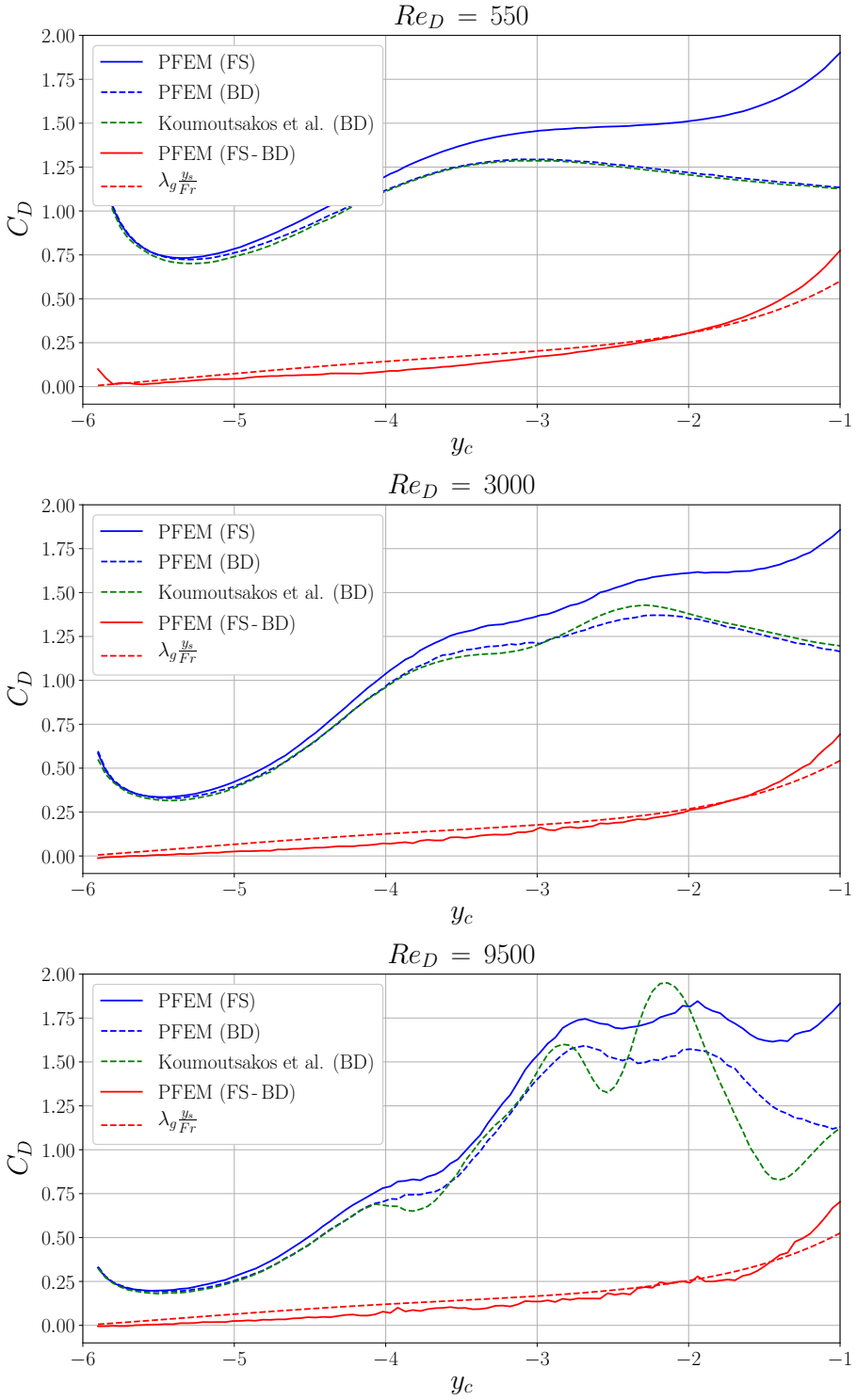


Figure 5.24: Drag coefficient for an impulsively started cylinder at $Re_D = 500$, 3000 and 9500. Comparison between PFEM simulations in a bounded domain (BD, blue dashed line) and a free surface (FS, blue continuous line, $Fr = 0.5$). The results of Koumoutsakos et al. [93] (BD, green dashed line) obtained in a bounded domain with a vortex method are also shown. The curves are plotted with respect to y_c, which is not relevant for the results in a bounded domain (a depth y_c corresponds to the time $t = y_c + 6$ from the impulsive start). The continuous red line is the measured excess of the PFEM drag coefficient with a free surface (FS) compared to the drag coefficient obtained in a bounded domain (BD). The red dashed line shows the linear relation of this excess drag with the free-surface elevation y_s (Eq. (5.15)).

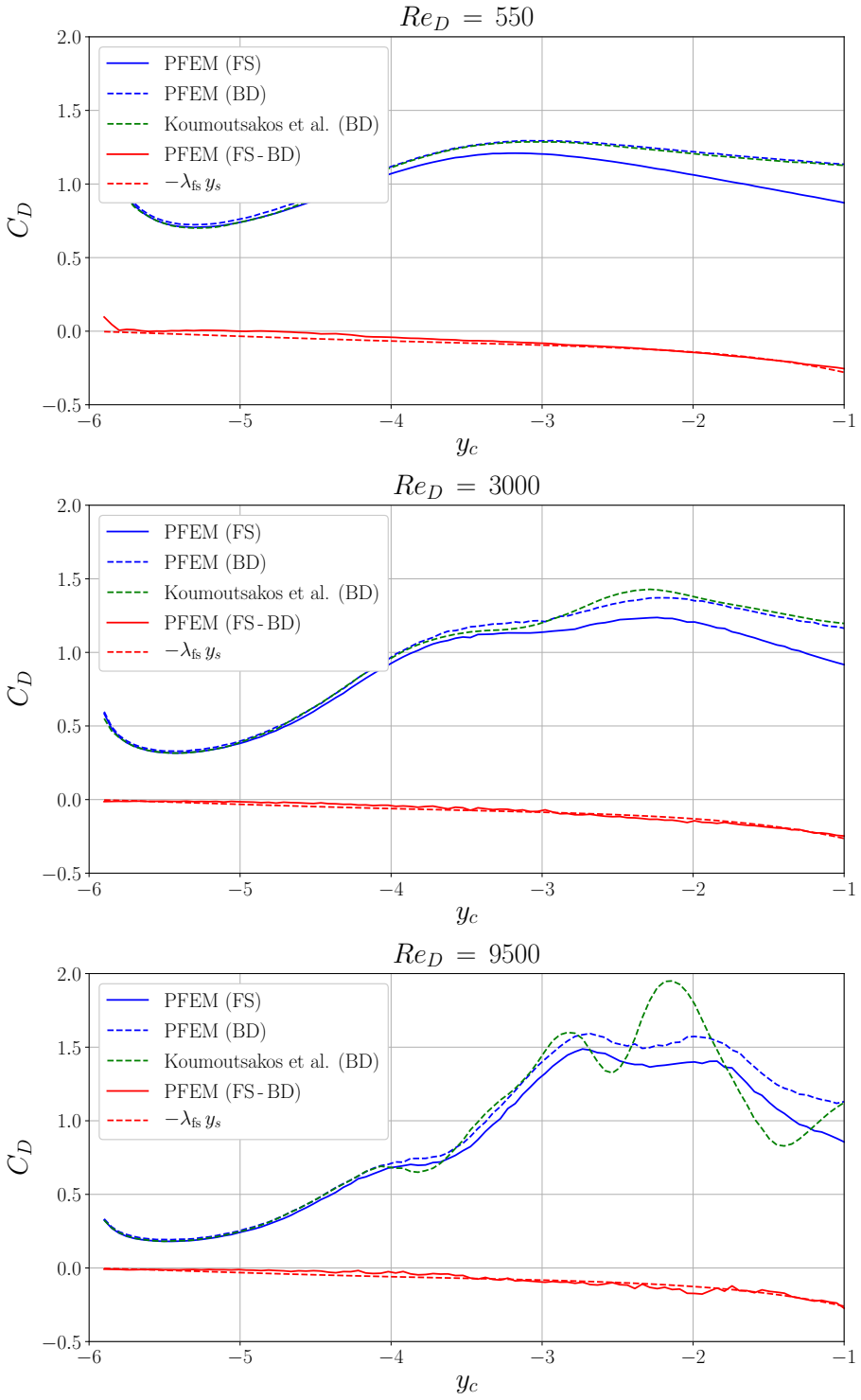


Figure 5.25: Drag coefficient for an impulsively started cylinder at $Re_D = 500$, 3000 and 9500. Comparison between PFEM simulations in a bounded domain (BD, blue dashed line) and a free surface (FS, blue continuous line, $Fr = 10$). The results of Koumoutsakos et al. [93] (BD, green dashed line) obtained in a bounded domain with a vortex method are also shown. The curves are plotted with respect to y_c, which is not relevant for the results in a bounded domain (a depth y_c corresponds to the time $t = y_c + 6$ from the impulsive start). The continuous red line is the measured deficit of the PFEM drag coefficient with a free surface (FS) compared to the drag coefficient obtained in a bounded domain (BD). The red dashed line shows the linear relation of this drag deficit with the free-surface elevation y_s (Eq. (5.16)).

As illustrated above, the drag excess is directly related to the free surface deformation. At low Fr, the free surface deformation is small and the free surface acts like a wall. The fluid between the cylinder and the free surface can only escape on the side and the pressure increases accordingly in this region, thereby increasing the drag. This drag excess can be quantified by considering the weight of the fluid above the initial free surface as additional contribution to the drag, as schematically shown in Fig. 5.26. The drag excess can thus also be understood as "gravity drag". The over time increasing weight of the fluid above the cylinder should scale as $aY_s\rho g$, where $Y_s = y_s a$ is the dimensional free-surface elevation and $y_s = h + 1 + y_c$. Consequently, the drag excess (or gravity drag) coefficient can be approximated as

$$C_g = \lambda_g \frac{a^2\, y_s\, \rho\, g}{\rho\, U^2\, a} = \lambda_g \frac{y_s}{Fr}\,, \tag{5.15}$$

where λ_g is a proportionality constant. A regression analysis on the parameter λ_g gives the values 0.6, 0.57 and 0.56 for $Re_D = 550$, $Re_D = 3000$ and $Re_D = 9500$, respectively. The weak variation of λ_g across different Reynolds numbers provides support to this simple model.

It should be emphasized that y_s itself also depends on the Froude number, as clearly shown by the aforementioned dependence of $h(y)$ on Fr. Specifically, at $y_c = -1$ Eqs. (5.8) or (5.10) for $h^* = y_s(y_c = -1)$ suggest that the increase of y_s with Fr is weaker than the explicit $1/Fr$ decrease of C_g in Eq. (5.15), so that overall, the drag excess coefficient decreases with the Froude number. Unfortunately, the drag difference between free-surface and confined domain is not available at other intermediate Froude numbers so that this model prediction cannot be further validated. The only other Froude number for which results are available is the case $Fr = 10$. At this larger Froude number, the drag excess model predicts a smaller but positive drag excess. However, the results just discussed above show a drag deficit in this case. This indicates that another competing effect is at play.

The physical mechanism for this drag deficit is not directly related to gravity, but rather to the mere presence of the free surface, i.e., a material line at constant atmospheric pressure. In particular, the lower drag observed is due to the absence of fluid above the free surface that needs to be displaced when the cylinder moves. Results in Fig. 5.25 suggest that the drag deficit (or free-surface drag) coefficient could be modeled as

$$C_{\text{fs}} = -\lambda_{\text{fs}}\, y_s\,, \tag{5.16}$$

where λ_{fs} is another constant of proportionality. In this case, a regression analysis gives $\lambda_{\text{fs}} = 0.56$, 0.55 and 0.55 for $Re_D = 550$, $Re_D = 3000$ and $Re_D = 9500$, respectively. Figure 5.25 shows a very good match between this simple model and the data with an even lower variation of λ_{fs} across Reynolds numbers compared to the drag excess model. Surprisingly, the calibrated value of λ_{fs} in Eq. (5.16) is almost identical to λ_g in Eq. (5.15). Because this drag deficit model is mostly ad-hoc without any physical justification, this similarity between λ_{fs} and λ_g in the two models in most likely fortuitous. Nonetheless, it suggests that there is neither drag excess nor drag deficit at $Fr \approx 1$. Finally, note that here again y_s depends on Fr so that C_{fs} also does.

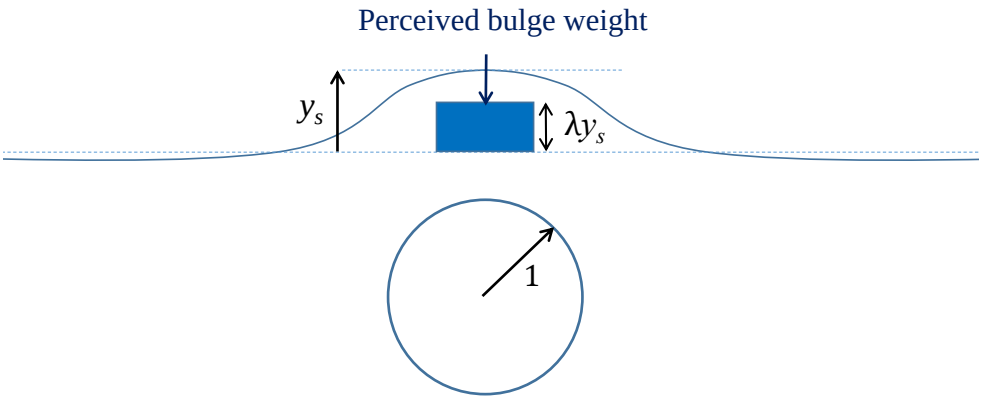


Figure 5.26: Illustration of the bulge weight effect increasing the drag above the cylinder rising toward the free surface.

In summary, the drag acting on a cylinder rising vertically towards a free surface at constant velocity, and its deviation from the drag of the similar flow in a bounded domain, stems from two opposite effects. On the one hand, the absence of fluid above the free surface has the effect to decreases the drag coefficient, leading to a negative contribution called the free-surface drags C_{fs}, related to the free-surface elevation y_s through Eq. (5.16). On the other hand, the bulge of height y_s forming at the free surface has the effects of increasing the drag coefficient through its weight in the gravitational field, leading to a positive drag contribution called the gravity drag C_g given by Eq. (5.15). It should be emphasized that these results are only a very preliminary step towards a predictive model, for several reasons:

- For the considered Froude numbers ($Fr = 0.5$ and $Fr = 10$), gravity is not completely dominant, nor completely negligible, such that C_{fs} and C_g should have been considered simultaneously when fitting the constants λ_g and λ_{fs}.
- The assumption that C_g and C_{fs} are both linear in y_s is not well justified, and more complex relation may fit better, especially for C_g as the fit is not as good as for C_{fs}.
- The actual dependence of C_g and C_{fs} on the Froude number could unfortunately not be verified.
- While a correlation for $h^*(Fr)$ has been proposed, the dependence of y_s on Fr for other cylinder positions ($y_c < -1$) is not available.

Nonetheless, these relations between drag excess/deficit and free-surface elevation are an important result as they directly relate the dynamics occurring in depth with observable kinematics of the free surface. In the next chapter, the main attention is on the prediction of $h(t)$ (and therefore $y_s(t) = h(t) + y_c(t) + 1$) based on models of the averaged forces and convective terms along the cylinder surface. From there, better models could be later obtained to predict C_g and C_{fs}, without any a priori knowledge of $y_s(t)$, by considering simulations at different Froude numbers, and simultaneous fits of the C_g and C_{fs} models. However, these possible extensions have not been considered in the present work.

Finally, it is important to consider how C_g and C_{fs} fit into the existing literature. Tyvand and Miloh [109] identified three contributions to the zeroth-order forces (with respect to time) on a rising cylinder submerged in an inviscid fluid [109]:

- A downward dynamic-pressure force.
- A downward force due to free-surface nonlinearity.
- An upward force due to geometric nonlinearity.

In our case, the net upward force corresponds to C_{fs}. Note that the terminology behind the different contributions in Tyvand and Miloh comes from the different terms within the potential flow formalism of the problem, and are not directly applicable to the present full-time computation of the problem using the Navier-Stokes equations. On the other hand, C_g corresponds to the effect of the gravity, which was also studied by Tyvand and Miloh in the small time limit. For this term, they have found an analytical expression based on a Fourier series, involving the non-dimensional cylinder radius $\epsilon = \frac{a}{d}$, i.e., the radius normalized by the cylinder initial depth. However, their formula is only valid in the small time limit, while the present work investigates the behavior of C_g and C_{fs} during the overall rising phase of the cylinder below the initial free surface.

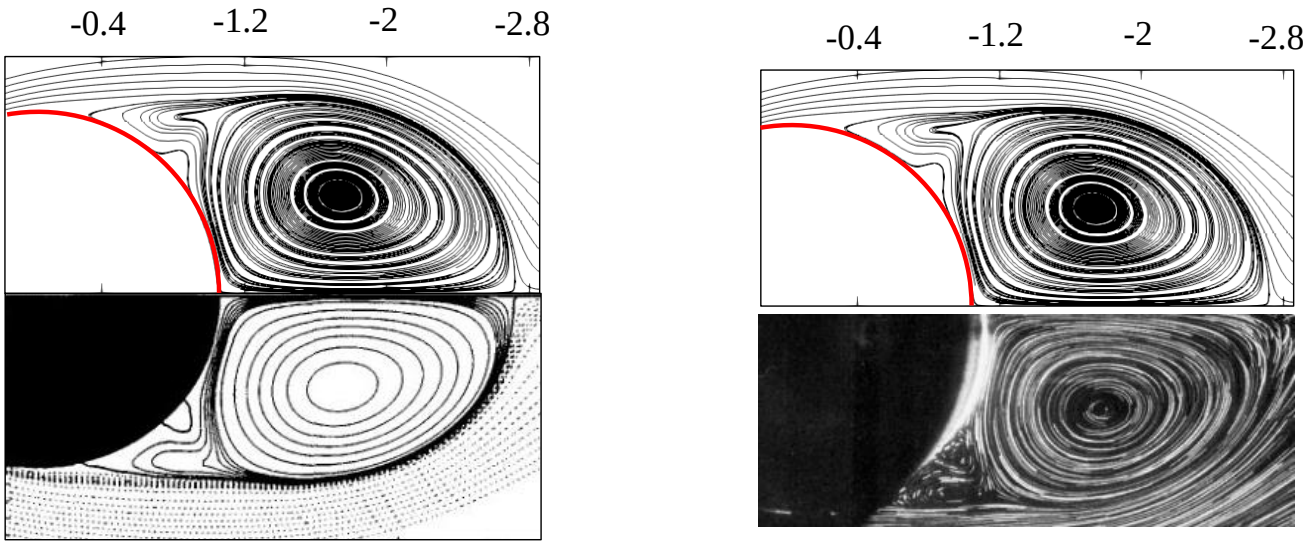
Wake dynamics

The results for the friction coefficient and the surface vorticity suggest that the free surface does not impact the wake. This is further analyzed by considering the streamlines of the wake, in the cylinder frame of reference, for $Re_D = 550$, 3000 and 9500, and $Fr = 0.5$, in Figs. 5.27, 5.28 and 5.29, respectively. The PFEM simulations are compared with the numerical results of Koumoutsakos et al. [93] (Figs. 5.27a and 5.29a) and the experimental results of Bouard et al. [129] (Figs. 5.27b, 5.28a and 5.29b). The PFEM streamlines (top image of the respective figures) are very similar to those obtained in the literature, despite the presence of the free surface. In particular, a good qualitative match is found regarding the

size and position of the different vortices. This further demonstrates that the free-surface deformation has a negligible impact on the wake dynamics (note that the opposite is not true, as shown by Vincent et al. [123]). A first possible explanation is that the wake is shielded from the free surface by the body itself and thus feels only weakly its presence. Another explanation is that the wake development takes place during the entire motion of the cylinder, from its start. If the starting depth is sufficiently large, the deformation of the free surface remains very limited during most of this wake formation process and only matters when the cylinder reaches the free surface.

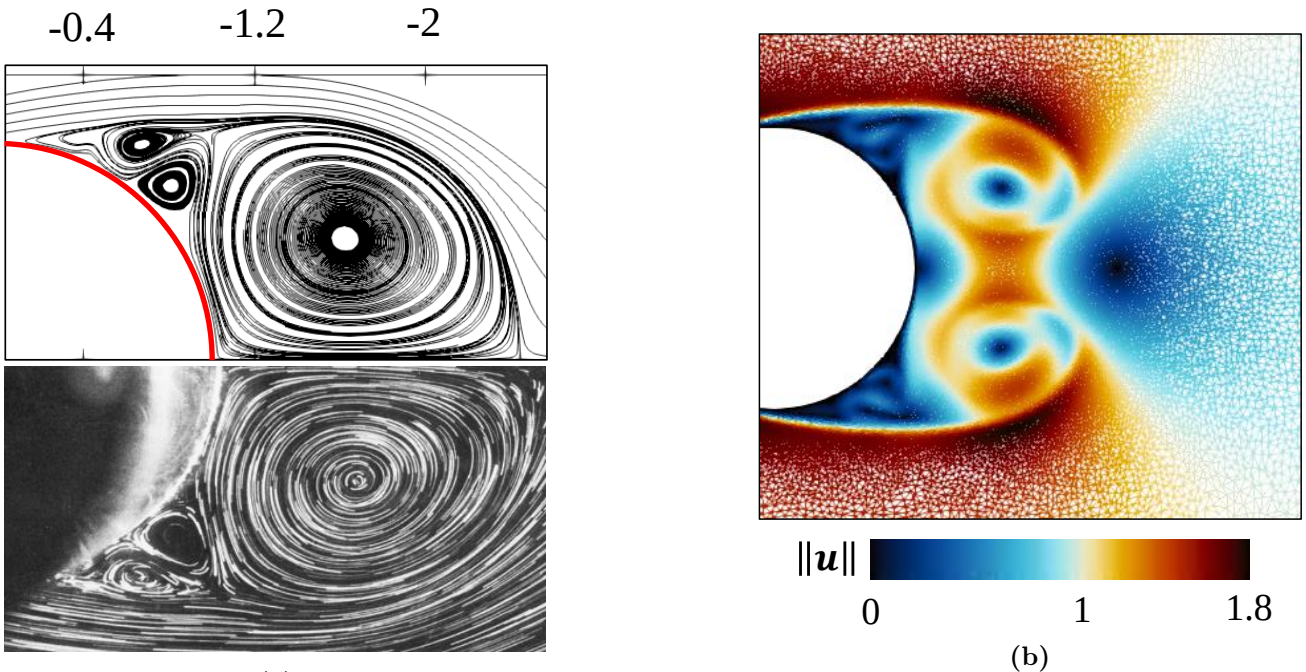
This has a practical consequence for the predictivity of flows around bodies crossing a free surface. Before the interface crossing, the wake can be more or less assumed to be in a state that only depends on its release depth[7], and which can be computed in a bounded domain, for which more efficient classical CFD solvers are available, or directly obtained from the literature. Furthermore, the free-surface deformation could be approximated through an inviscid calculation with a pseudo-body that includes the body and its wake. Note that the behavior of the wake during and after the interface crossing is briefly discussed in Appendix G.

[7] At least for the range of parameters considered here.



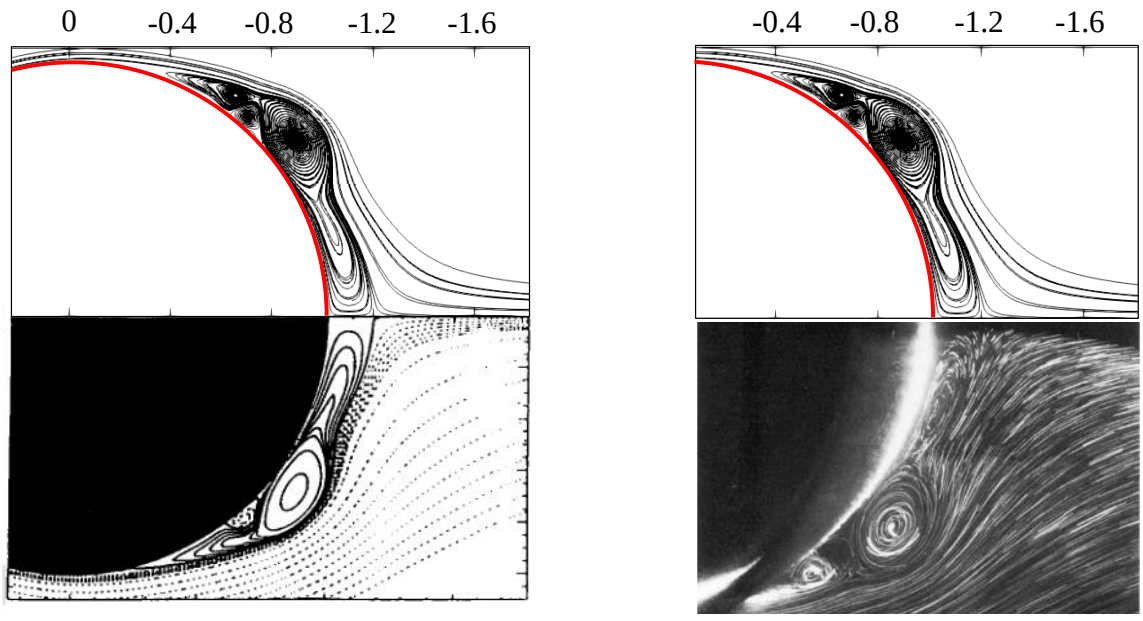
(a) **(b)**

Figure 5.27: Streamlines in the wake of the impulsively started cylinder at $Re_D = 550$ after a non-dimensional time $t = 5$ (i.e., $y_c = -1$ for the PFEM simulation with $Fr = 0.5$). (**a**) Comparison between the results obtained with the PFEM (top) and the numerical results of Koumoutsakos et al. [93] (bottom); (**b**) comparison between the results obtained with the PFEM (top) and the experimental measurements of Bouard et al. [129] (bottom). The cylinder boundary in the PFEM images (top) is represented by a thick red curve.



(a) **(b)**

Figure 5.28: (**a**) Streamlines in the wake of the impulsively started cylinder at $Re_D = 3000$ after a non-dimensional time $t = 5$ (i.e., $y_c = -1$ for the PFEM simulation with $Fr = 0.5$), obtained with the PFEM (top), and experimentally by Bouard et al. [129] (bottom). The cylinder boundary in the PFEM image (top) is represented by a thick red curve. (**b**) PFEM mesh wireframe colored by the velocity magnitude (in the cylinder frame of reference) at the same time as in (**a**).



(a) **(b)**

Figure 5.29: Streamlines in the wake of the impulsively started cylinder at $Re_D = 9500$ after a non-dimensional time $t = 5$ (i.e., $y_c = -1$ for the PFEM simulation with $Fr = 0.5$). (**a**) Comparison between the results obtained with the PFEM (top) and the numerical results of Koumoutsakos et al. [93] (bottom). (**b**) Comparison between the results obtained with the PFEM (top) and the experimental measurements of Bouard et al. [129] (bottom). The cylinder boundary in the PFEM images (top) is represented by a thick red curve.

5.2.3 Comparisons with experiments

To further validate the numerical results, they are compared to in-house experiments that have been conducted for the case of a cylinder pulled out of a liquid bath at constant velocity [125]. Because in experiments real fluids are used, any variation of the Froude number (i.e., of the cylinder velocity) implies also a corresponding variation of the Reynolds number. Therefore, to separate the two effects, two different fluids have been considered: water and oil. As the viscosity of the latter is much larger than that of water, lower Reynolds numbers can be achieved for a given cylinder velocity (Froude number), which has the advantage that the spatial resolution in simulations can be lower. Therefore, all comparisons are done for the oil case here, and only this case is discussed.

First, the experimental measurements are briefly described. Then, the numerical setup is summarized. Finally, numerical results for the surface elevation and the force acting on the cylinder are compared to experimental results.

5.2.3.1 Experimental setup

The experimental setup for oil consists of a rectangular tank with a length of 72 cm, a width of 30 cm and a height of 79 cm, filled with silicon oil (50 cSt), in which a cylinder is pushed (for surface elevation and film thickness measurements) or pulled (for force measurements) from an initial depth at constant velocity (typically between 0.1 to 1 m/s[8]). Oil experiments were conducted with two different cylinders: a thicker cylinder of radius $a = 2.5$ cm and aspect ratio $L/a = 10$ and a thinner cylinder of radius $a = 1.25$ cm and aspect ratio $L/a = 12$, whose axes were aligned with the length of the tank and parallel to the free surface, as shown in Fig. 5.30. To further reduce 3D effects in the experiment, end-plates were used for the surface deformation (and film thickness measurements) but not for the force measurements. The initial depth varied depending on the measurement type and the presence or not of end-plates.

The surface deformation during the initial phase was obtained using high-speed imaging of the liquid-air interface and the film thickness during drainage using a Chromatic Confocal Point Sensor (CCPS). Note that for this second measurement, the cylinder was at rest above the free surface. The force acting on the cylinder during its vertical motion was measured with a strain gauge. Two-dimensional planar Particle Image Velocimetry (PIV) was also used but the results are not considered here. More informations about the experimental setup, protocols and results can be found in Dorbolo et al. [125].

Numerical setup

Because simulations are two-dimensional, a 2D cross-section of the experimental setup with the same tank width and height is considered. The geometrical, mesh and mesh adaptation parameters are given in Table 5.7. The initial meshes are shown in Fig. 5.31. For the mesh adaptation, a geometrical mesh refinement is used to impose a small progression of the target mesh size from $L^*_{\text{min,GEO}}$ to $L^*_{\text{max,GEO}}$. The progression starts from $L^*_{\text{min,GEO}}$ at the cylinder surface and initial free-surface level, to $L^*_{\text{max,GEO}}$ at the respective distances of $W - 1$ and $W/6$ from them. In addition, the solution-based mesh adaptation of Eq. (3.23) is used, whose the parameters are defined in Table 5.7; it is referred here as SOL.

5.2.3.2 Results

Comparison between simulations and experiments are first done with the thicker cylinder ($L/a = 10$, $d = 19.8$) for the thickness $h(y_c)$ of the oil column above the cylinder, and then with the thinner cylinder ($L/a = 12$, $d = 37.6$) for the force F acting on the cylinder. Finally, simulations with the thicker cylinder are analyzed and compared to a simple model. The thicker cylinder is considered in this case because its wake is more stable, as discussed below.

[8]The cylinder reaches its steady target velocity after a short acceleration period. In particular, a velocity $U = 1$ m/s is reached after the cylinder has moved 12 cm from its initial position.

Figure 5.30: Picture of the first experimental setup. Three instruments are shown on this picture, and are framed by yellow rectangles. On the top right, there is the force sensor; on the lower left, there are the high speed camera and the laser used for PIV. Courtesy of S. Dorbolo.

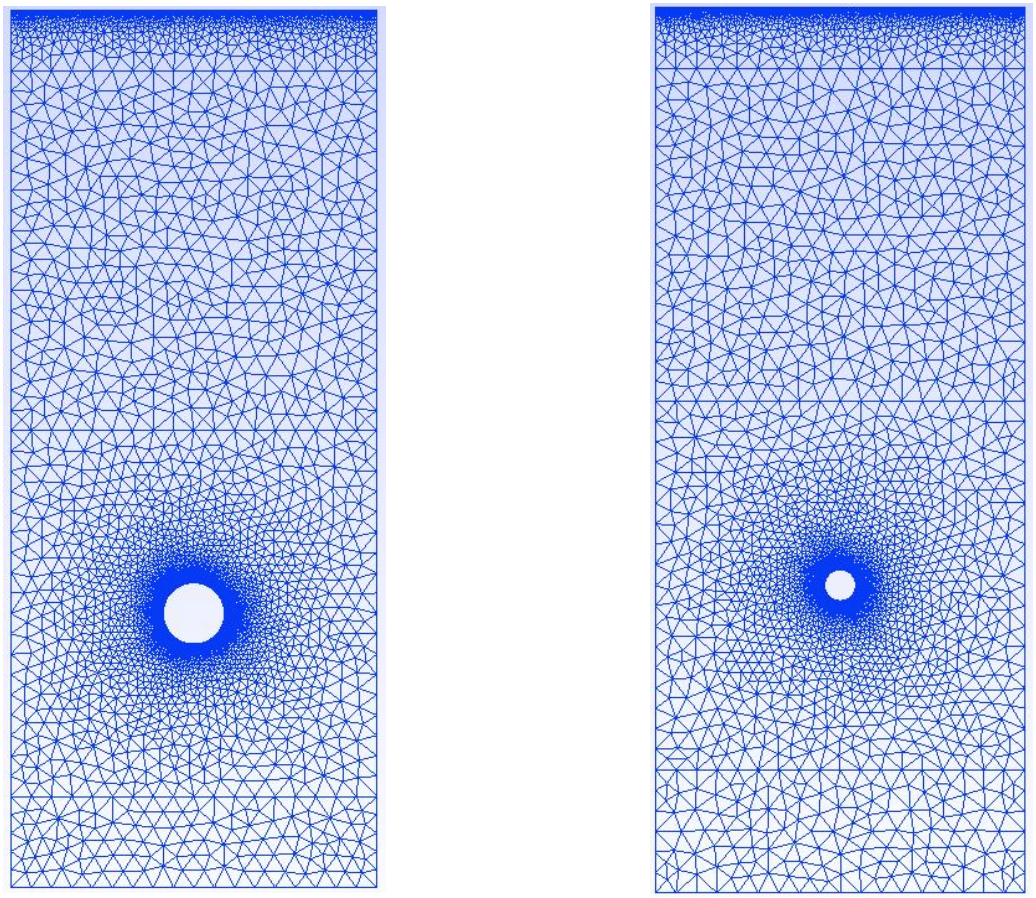
Column height for the thicker cylinder in oil

The results for the height $h(y_c)$ of the oil column above the thicker cylinder ($L/a = 10$, $d = 19.8$) are shown in Fig. 5.32 where the experimental and numerical results are respectively represented by the dashed and the continuous lines. Globally, a very good agreement is observed for all cylinder velocities. Note that simulations can reach larger y_c, i.e., smaller values of h, where experimental results are either hard or impossible to obtain because of the limited possible vertical displacement of the camera and measurement uncertainties at small h. In the experiment, the film thickness is then measured much more precisely with the CCPS, but this requires to bring the cylinder to rest. This specific case has not been reproduced with simulations. The exponential decreasing rate of $h(y_c)$ decreases as the film gets thinner, because of viscous effects. This is confirmed by the fact that the curves approach, or reach, the viscous-gravity length scale l_ν, where the viscous-to-gravity balance becomes dominant over inertial effects.

Additionally, images of the numerically simulated flows are shown in Fig. 5.33 for different values of the velocity U and cylinder positions y_c. One can observe that the wakes are similar, but are slightly longer for smaller velocities (smaller Reynolds number), which is a well-known effect in the literature on viscous flows around a cylinder [87]. For the larger velocity (larger Reynolds number), the wake starts to become unstable, as indicated by its small asymmetry (right column of Fig. 5.33). The free-surface elevation and the film thickness are larger for larger velocities, which is consistent with the results for $h(y_c)$. It is also interesting to note that, after the interface crossing, the film thickness seems to be uniform along the cylinder surface (i.e., for a significant range of angles θ from the cylinder apex), as already observed in the literature (see Section 5.1). This observation is used when modeling the film thickness in Chapter 6.

Vertical force on the thinner cylinder.

The second quantity of comparison between the experiments and the simulations is the downward force opposing the cylinder motion. While it is usually referred to as the drag force in the literature, it is



(a) **(b)**

Figure 5.31: Initial meshes for the simulation of the thicker **(a)** and the thinner **(b)** cylinders.

		Thicker cylinder ($L/a = 10$)	Thinner cylinder ($L/a = 12$)
Tank half width	W	6	12
Tank height	H	28.8	57.6
Initial depth of the cylinder	d	19.8	37.6
Maximal mesh size	$L^*_{\text{max,GEO}}$	0.6	1.2
Minimal mesh size	$L^*_{\text{min,GEO}}$	0.0033	
Time step size (*)	Δt	0.0033	
Maximal mesh size of SOL	$L^*_{\text{max,SOL}}$	0.45	0.9
Minimal mesh size of SOL	$L^*_{\text{min,SOL}}$	0.03	
Minimal velocity gradient (*)	$\|\|\nabla\mathbf{u}\|\|_{\text{min}}\|\|$	0.006	
Maximal velocity gradient (*)	$\|\|\nabla\mathbf{u}\|\|_{\text{max}}\|\|$	2	
Power parameter	β	1/3	

Table 5.7: Geometrical, mesh and mesh adaptation (see Eq. (3.23)) parameters for the simulations. See Fig. 5.1 for the definition of the different length scales, which are all non-dimensionalized by the cylinder radius a. (*) Quantities involving the time are also non-dimensionalized by the cylinder velocity U.

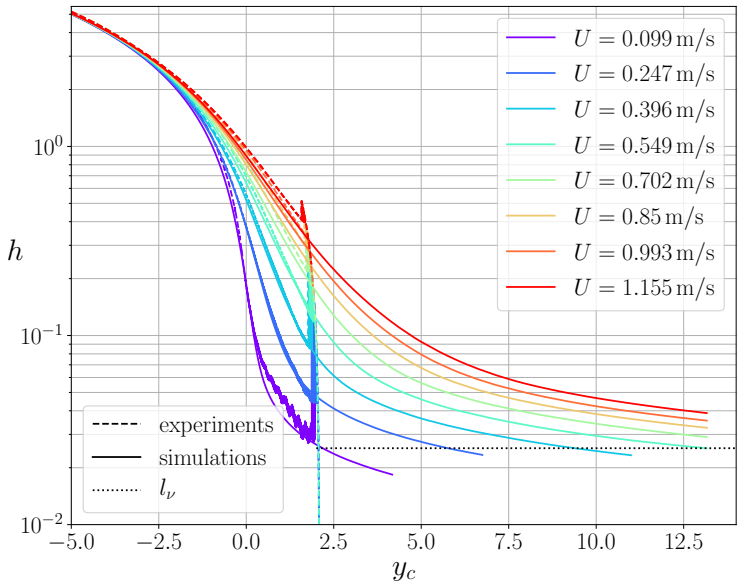


Figure 5.32: Comparison of the height h of the oil column above the thicker cylinder ($L/a = 10$) as a function of its vertical position y_c between experiments (with end-plates, dashed lines) and simulations (continous lines) for different cylinder velocities U ($Fr = 0.04$ to 5.44, $Re = 99$ to 1155). The initial cylinder depth in the experiment is $d = 15$. Note that in the experiment the thickness of the oil film can only be measured until shortly after the free-surface crossing. The viscous-gravity length scale l_ν is indicated by the horizontal black dotted line.

preferred to avoid this terminology here as this force includes different contributions, including the drag induced by the cylinder motion, but also the buoyancy (or Archimedes') force which is oriented upward and non-zero even at rest. One could try to remove the buoyancy contribution, as it is well defined by the body's volume and the fluid density. However, during the interface crossing, the cylinder is formally surrounded by fluid everywhere while some parts of his volume is located above the initial free-surface level. In these conditions, it is not clear how to define the buoyancy force and how to isolate it from the other contributions along the vertical position[9]. This difficulty of clearly defining the different force contributions at the interface crossing is symptomatic of the problem complexity, and it is expected that the level of surface deformation plays a particular role here. While the surface deformation is intuitively lower at smaller Froude numbers, its impact on the force exerted on the cylinder, and on the flow above it, is actually stronger through the higher effect of gravity. This larger effect at small Froude number was already shown through the gravity drag C_g introduced in Section 5.2.2.2. Its effect on the flow above the cylinder is investigated in more detail in Chapter 6.

Before showing the forces, the procedure to obtained them is described. When comparing the force from experiments and simulations, two important aspects have to be taken into account. First, the force sensor in the experiment measures all contributions at play, which includes, as previously mentioned, the dynamic drag, the static buoyancy, the added-volume and inertial effects during the acceleration phase, but also the cylinder weight. The zero-level thus depends on which contributions are included when reporting the force. Because the cylinder weight is irrelevant in the simulations, it is here subtracted from the measured force. Consequently, the reported force corresponds to minus the static buoyancy force at $t = 0$ and vanishes at large times when the film has fully drained. Secondly, as the simulations are two dimensional, the force is defined per unit length, and should therefore be multiplied by the length of the cylinder. However, the sectional force in the experiment is obviously not constant along the cylinder length because of three-dimensional effects towards the cylinder ends. This is expected to be a major source of discrepancies between experiments and simulations.

[9]Note that the "drag" terminology was used in the previous sections because there, it was only referring to the dynamics occurring below the free surface, where the buoyancy force is assumed to be well defined and can be removed from the other contributions to the overall vertical force.

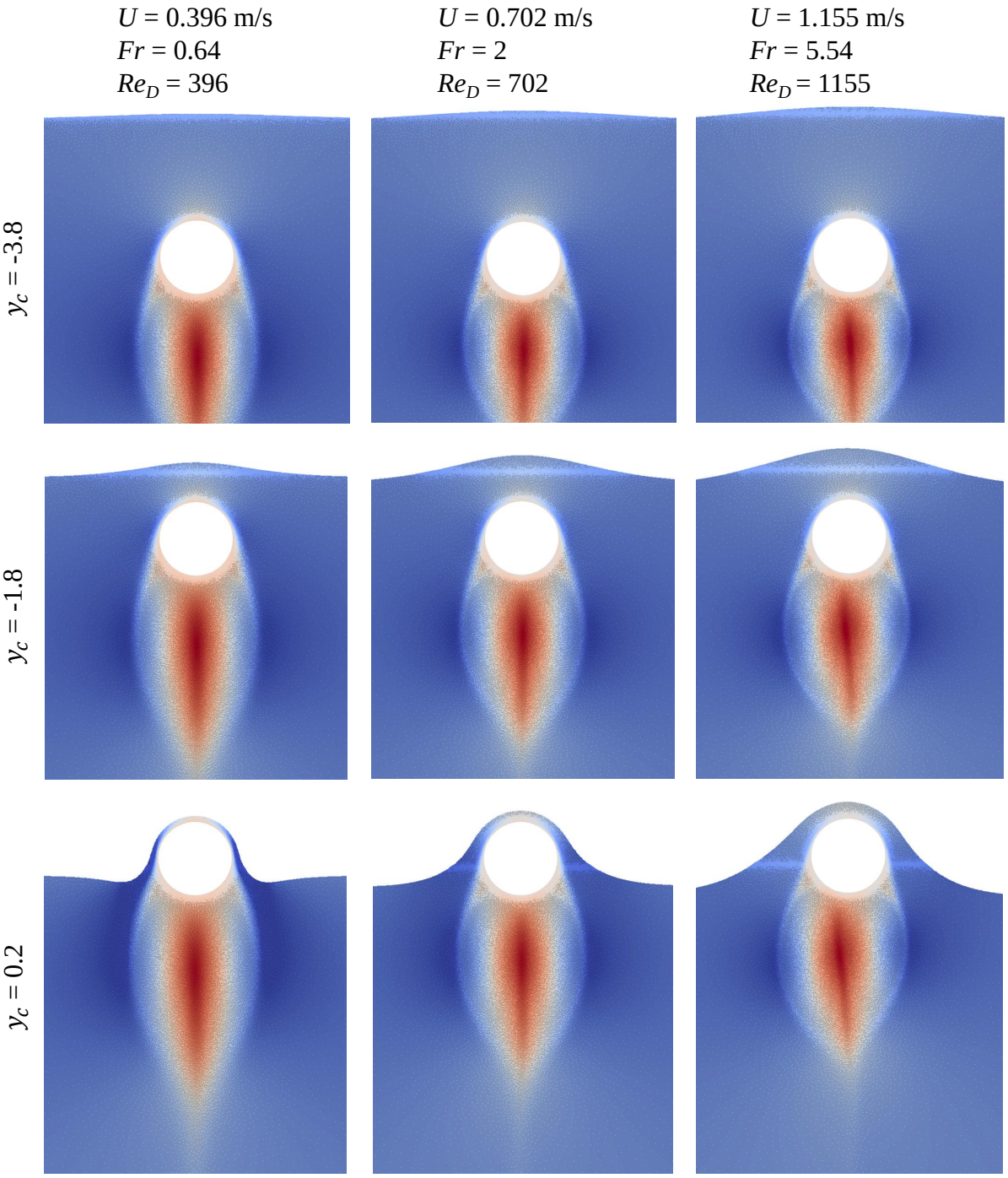


Figure 5.33: Contour plots of the vertical component of the velocity around the rising thicker cylinder for different velocities U and positions y_c obtained from simulations.

Figure 5.34 shows the force (from which the cylinder weight has been subtracted) on the thinner cylinder ($L/a = 12$) for different velocities ranging from 0.1 m/s (purple lines) to 1.3 m/s (red lines) by steps of 0.1 m/s from the experiment (dashed lines, $d = 38$) and simulations (continuous lines). In the simulations, after an initial spurious peak due to the infinite acceleration of the impulsive start, the force rapidly increases, while the wake is developing, before reaching a maximum value around $y_c = -34$. Then, the force slowly decreases until the cylinder approaches the free surface. The rapid force growth and its maximum value increase with the cylinder velocity. At very low velocities, they are not distinguishable and the force remains more or less constant during the immersed phase. This

behavior is also observed experimentally, but at a larger y_c. Specifically, the force increase is much more gradual, the maximum value is reached just after the end of the acceleration phase and is much lower than in the simulations. Interestingly, this maximum value seems to almost lie on the numerical curves and, after this point, a very good match is observed between numerical and experimental results. The discrepancies observed during the initial phase are thus mostly due to the different kinematics imposed to the cylinder, but once the cylinder has reached a constant speed and its wake has formed, a much better agreement is found. At the largest velocity, a large and rapid increase of F is observed around $y_c = -12$ in the simulations. This significant quantitative and qualitative change compared to lower velocities is likely caused by the destabilization of the wake and the inception of vortex shedding, as illustrated in Fig. 5.35. Wake destabilization is observed for the four largest values of the velocity, and is significantly larger for the largest velocity. The more stable experimental drag curves are possibly due to a more progressive development of the wake (much weaker acceleration) such that instabilities have not sufficient time to develop (constant velocity is maintained over a shorter distance until the free surface). Additionally, three-dimensional effects can also play an important role.

Other discrepancies are found when the cylinder approaches and crosses the free surface. For low velocities, the drag starts to rapidly increase because the buoyancy force decreases to zero when the cylinder emerges out of the bath. It then reaches a positive maximum value that corresponds approximately to the weight of the entrained liquid, as discussed in the next section. With the subsequent drainage of the film the force then continuously decreases, and should eventually vanish after full drainage. With increasing velocity, the time at which the maximum force occurs and its magnitude increase because more liquid is entrained. The same behavior is observed experimentally and numerically, but the peak value of the force is lower and occurs at larger y_c in the experiment. Moreover, the force during drainage is consistenly larger in the simulations, most likely due to the two-dimensional simplification.

For the larger velocities, simulations show a rapid decrease of the force, that departs from the experimental results, before the interface crossing. After reaching a minimum at $y_c \geq 1$, the force increases back to the experimental value. This behavior could again be due to the instabilities developing in the cylinder wake at larger velocities, but more conclusive analyses are necessary to ascertain it. Nonetheless, the drainage phase seems to be again similar to lower velocities. This last phase is further investigated in the next section.

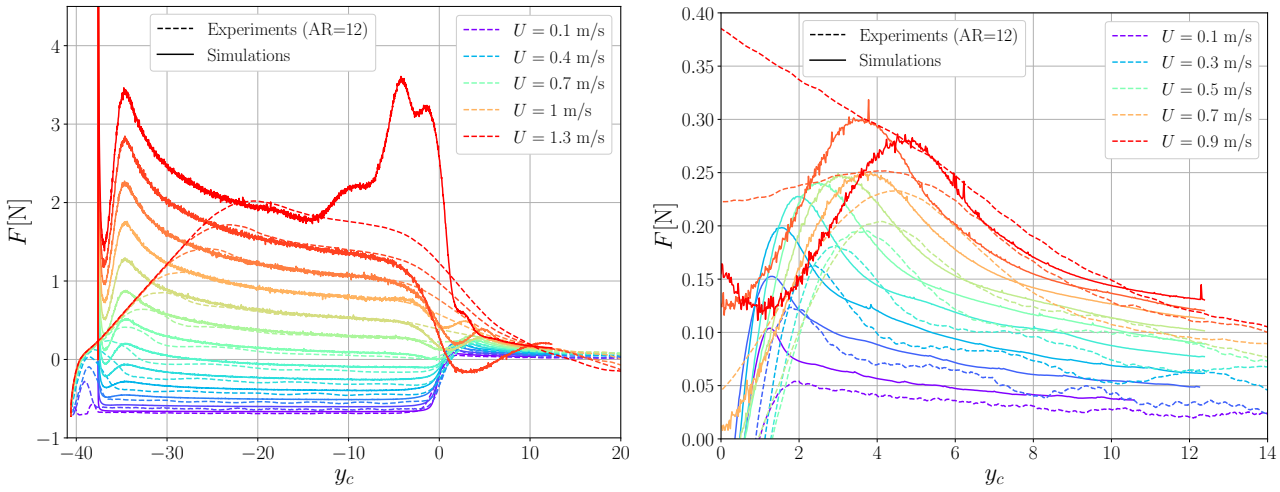
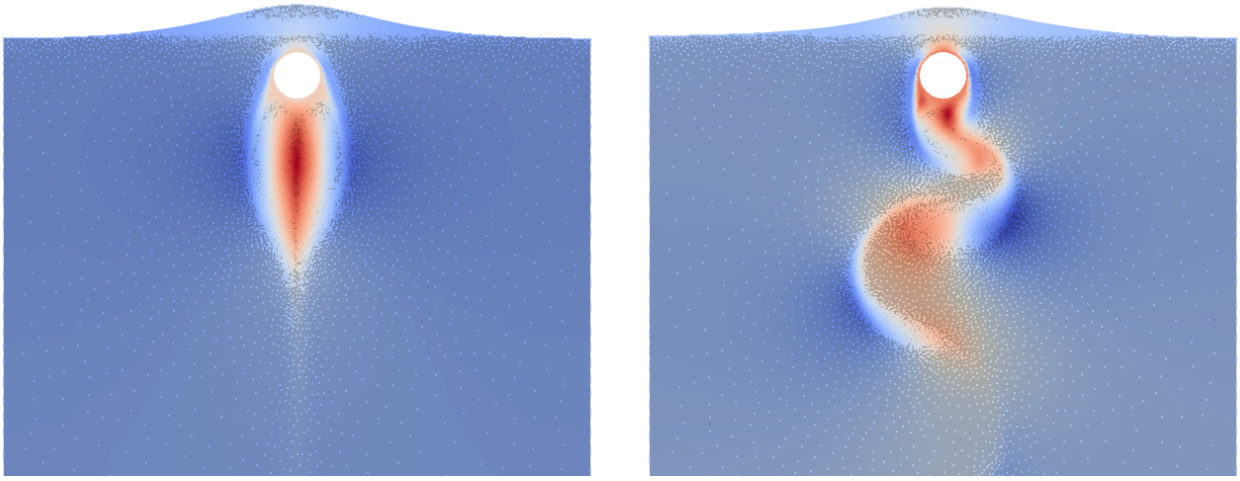


Figure 5.34: Force F acting on the thinner cylinder ($L/a = 12$, the cylinder weight has been subracted) in oil as a function of the cylinder vertical position y_c obtained numerically (continuous lines) and experimentally ($d = 38$, dashed lines) for velocities ranging from 0.1 m/s (purple lines) to 1.3 m/s (red lines) by steps of 0.1 m/s. It corresponds to Reynolds numbers ranging from 50 to 650 and Froude numbers ranging from 0.0815 to 13.78. (**a**) Entire y_c range and (**b**) close-view on values $y_c > 0$. For visibility, the color range has been reduced for (**b**), such that it does not directly correspond to the curves in (**a**).

Vertical force on the thicker cylinder



(a) **(b)**

Figure 5.35: Contour of the vertical component of the velocity, when the thinner cylinder is at position $y_c = -2$. (**a**) $U = 1.2$ m/s, $Re_D = 600$, $Fr = 11.74$: the wake remains almost perfectly symmetrical. (**b**) $U = 1.3$ m/s, $Re_D = 650$, $Fr = 13.78$: the wake is unstable and vortex shedding has appeared.

To check the assumption that the force at $y_c > 0$ is mainly due to the weight of the entrained liquid, a closer look is taken at the force obtained numerically for the thicker cylinder. The rational for considering the thicker cylinder is that its wake is stable for almost all velocities, despite a larger Reynolds number, because the initial location of the cylinder is closer to the free surface ($d = 19.8$). Unfortunately, experimental force measurements for the thicker cylinder are not available for this setup so that only numerical results are used.

The force (per unit length) obtained from simulations is shown in Fig. 5.36 for different velocities. For $y_c < 0$ (Fig. 5.36a), the observations are similar to those for the thinner cylinder. Although the wake seems to be more stable in this case, the rapid drop of the force just before interface crossing is visible for the three largest velocities, suggesting that small wake asymmetries could nonetheless be present, as previously illustrated in Fig. 5.33.

To correlate the force with the weight of the entrained liquid for $y_c > 0$, the weight (per unit length) W of the annulus of fluid surrounding the cylinder can be considered, assuming the thickness of the annulus to be uniform along the cylinder surface and equal to $h(y_c)$, as illustrated in Fig.5.37. It is thus

$$W = \pi \rho g a^2 \left((1+h)^2 - 1)\right) . \tag{5.17}$$

Figure 5.36b compares the actual force $F(y_c)$ with the scaled weight $1.2W(y_c)$. The clear correlation between the two quantities for all velocities supports the assumption that the force for $y_c > 0$ is mainly due to the entrained liquid, plus some contribution of the stretching of the emerging wake filament below it, which also seems to be proportional to the weight of the annulus of fluid.

Furthermore, the black dots in Fig. 5.36b show the value $1.4W(y_c)$ at the position y_c where F reaches its maximum. Again, a strong correlation is found between the maximum force and the amount of entrained liquid. The corresponding flow field at maximum F is shown in Fig. 5.38. Some wake asymmetry can be observed for the largest velocities. More importantly here, the horizontal thickness of the entrained liquid below the cylinder is of the same order for all velocities. This is further quantified in Fig. 5.39, where the minimal wake thickness Δ when the force is maximal is plotted as a function of the Froude number. For $Fr \geq 0.5$, Δ is more or less constant. On the other hand, the evolution in time (not shown) indicates that the entrained liquid filament below the cylinder shrinks, while the pressure on the lower cylinder surface decreases, inducing a suction akin to the well-known Coanda effect (Fig. 5.40). This suction reaches a maximum because of the opposite effect of a decreasing pressure and a thinner wake filament, i.e., a smaller wetted cylinder surface. Additionally, one can observe that the shrinking of the wake filament occurs at later times for higher velocities. Because more liquid is entrained in this case, it takes longer for the fluid to flow until this maximum suction is reached.

It is remarkable that this suction force is approximately proportional to the weight of the annulus

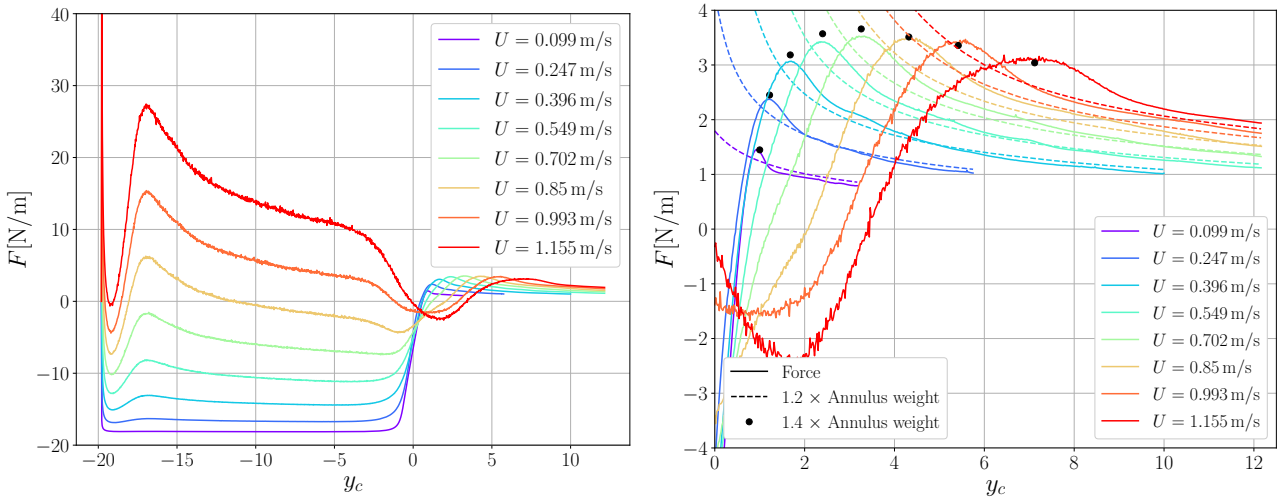


Figure 5.36: Force per unit length on the thicker cylinder in oil as a function of the cylinder vertical position y_c obtained numerically for different cylinder velocities (Reynolds number ranging from 100 to 1150 and Froude number from 0.04 to 5.4). (**a**) Entire range of positions y_c and (**b**) close view on $y_c > 0$. In (**b**) dashed lines are placed at 1.2 times the weight of the annulus of fluid surrounding the cylinder and black dots indicate the y_c position of the local maximal forces at a level corresponding to 1.4 times the weight of the annulus of fluid at that time.

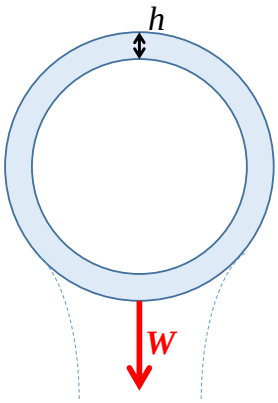


Figure 5.37: Illustration of the annulus of fluid draining around the cylinder after the interface crossing.

of fluid around the cylinder. In fact, it can be shown that both these quantities (i.e., the maximum suction force and the annulus weight) are related to the wake minimal thickness Δ at the time when the drag is maximal ($y_c > 0$), so that these two quantities are proportional to each other by a factor 1.4. In particular, a linear fit can be obtained for both, as illustrated in Fig. 5.39.

Finally, from the time $\overline{t}$ when the drag is maximum, corresponding to the cylinder elevation $\overline{y_c}$, a collapse of all force curves can be obtained by rescaling the force F by the annulus weight at that initial time, $W(\overline{y_c})$, and plotting their evolution with respect to the gravity−normalized time $t_g = \sqrt{\frac{g}{a}}t$. The result is shown in Fig. 5.41. This collapse, in addition to show that the force is proportional to the film weight all along its drainage, highlights the fact that the drainage is purely gravity-driven, and that the cylinder velocity does not play any role here. Nonetheless, for smaller velocities, the force (i.e., also the film thickness) decreases more slowly, which is likely due to the relatively stronger effects of viscosity occuring for a smaller film thickness (which is an indirect effect of using smaller velocities).

Conclusions

By comparing simulations and experiments of a cylinder rising toward and crossing the free-surface, for Reynolds number ranging from 50 (thin cylinder at $U = 0.1$ m/s) to 1155 (thick cylinder at $U =$

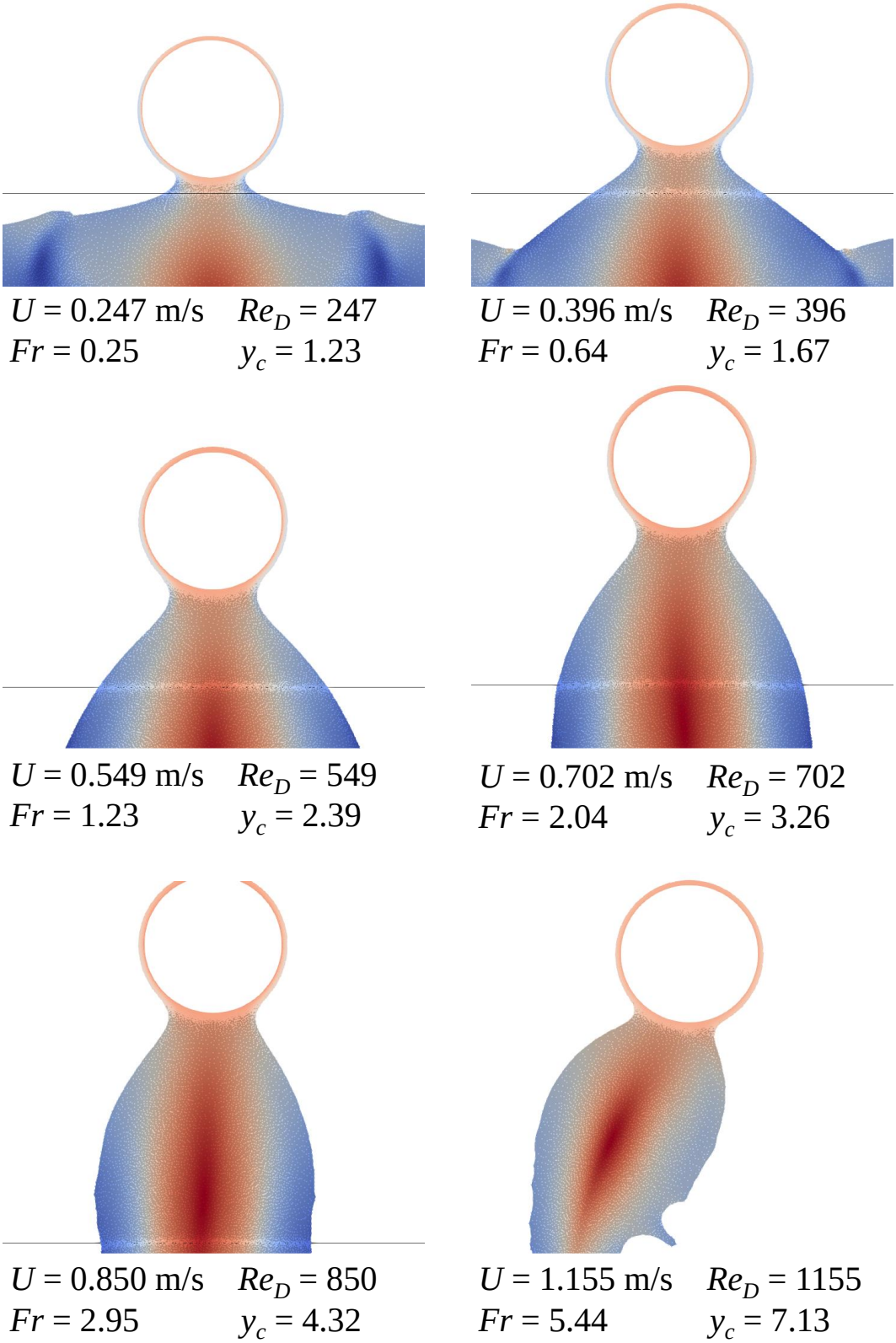
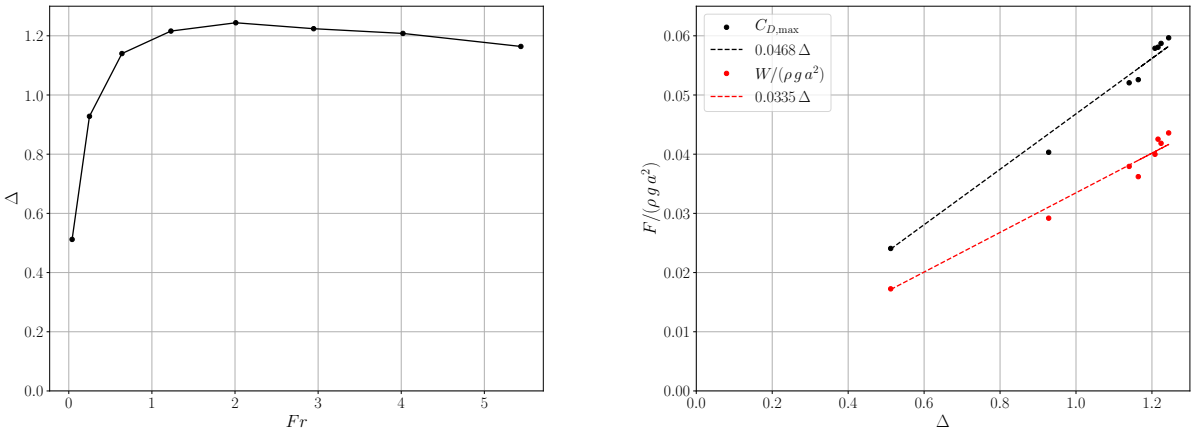


Figure 5.38: Contour of the vertical component of the velocity (from $-1.5U$ in blue to $2U$ in red) around the thicker cylinder at the position $y_c > 0$ when the downward force is maximal, indicated by the black dots in Fig. 5.36b.

1.155 m/s), and Froude number ranging from 0.04 (thick cylinder at $U = 0.1$ m/s) to 13.78 (thin cylinder at $U = 1.3$ m/s), it has been shown that:

- The drainage of the annulus of fluid around the cylinder is governed by an inertia-to-gravity balance, resulting in an exponential decrease of the film thickness $h(y_c)$. The exponential decrease



(a) **(b)**

Figure 5.39: (**a**) Dependency of the minimal wake thickness Δ when the suction effect is maximal on the Froude number. (**b**): Dependency of the strength of the suction effect and of the weight of the annulus with respect to the wake minimal thickness Δ. The linear fits between each of these two quantities and Δ suggest that they are proportional to each other with a factor 1.4.

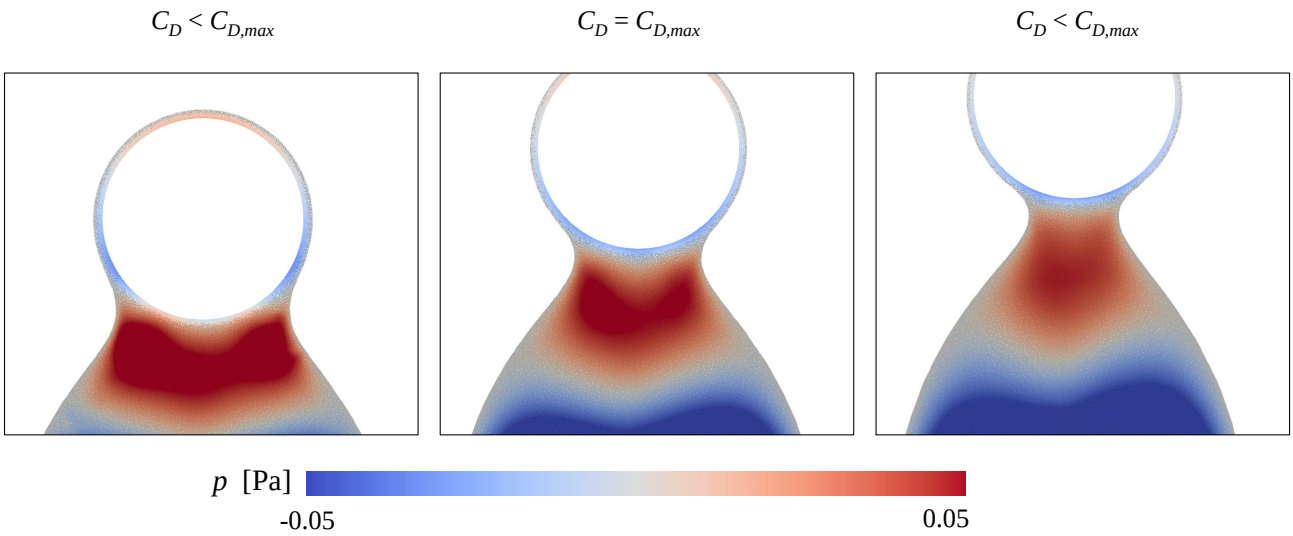


Figure 5.40: Contour of pressure before (left) and after (right) the time when the drag is maximal for the thicker cylinder at $U = 0.702$ m/s. Before, the pressure below the cylinder is high and starts to progressively decrease as there is less flowing fluid on each side. The maximal drag is then obtained by suction resulting from the lower pressure below the cylinder combined with "still" large wetted cylinder surface by the wake filament. Afterwards, the pressure remains low but the suction effect weakens because of the decreasing wake thickness (i.e., smaller wetted area on which the low pressure acts).

becomes slower as the film thickness approaches the viscous-to-gravity length scale l_ν where viscous effects become relatively more important.

- The force acting downward on the cylinder after the interface crossing is proportional to the weight of the annulus of fluid surrounding the cylinder, and is maximal when the wake minimal thickness is around 1.2 radius ($1 < Fr < 5$ in Fig. 5.39(**a**)). It results from a suction effects due to a small depression below the cylinder.

- The evolution of that force is mainly gravity-dependent, and the cylinder velocity has only an indirect influence on it through the thicker or thinner annulus of fluid around the cylinder. This dependence on gravity is shown through a collapse of the force evolution at all cylinder velocities, normalized it by the annulus weight and using a time description normalized by a gravitational time scale $t_g = \sqrt{\frac{a}{g}}$ (see Fig. 5.41).

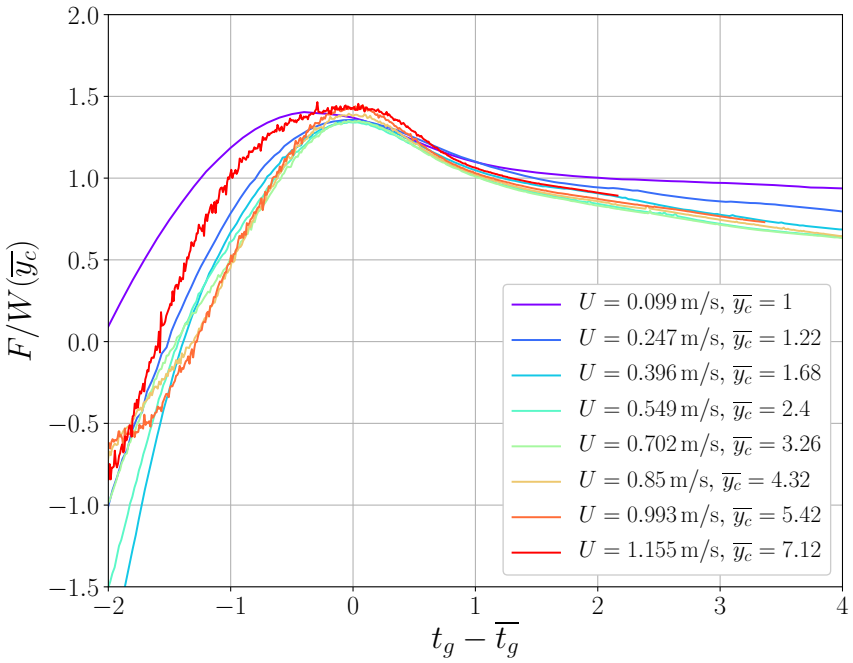


Figure 5.41: Drag force on the rising cylinder, normalized by the annulus weight W at the time $\overline{t_g}$ and cylinder position $\overline{y_g}$, when the drag force is maximal. The evolution is expressed in terms of the non-dimensional time $\overline{t_g} = \sqrt{\frac{g}{a}}t$.

To summarize, this section enabled to validate results of the previous one (evolution of film thickness $h(y_c)$) while complementing others. Specifically, while the behavior of the drag was investigated for $y_c < 0$ in the precedent section, the force after the interface crossing $y_c > 0$ has been investigated here.

5.2.4 Wake dynamics and fluid entrainment at very small Reynolds number ($Re_D \leq 1$)

So far, typical flow regimes at Froude number around 1 and Reynolds number around $1000 - 10000$ have been investigated. It has been shown that the surface elevation is mainly influenced by the Froude number, while the Reynolds number only plays a secondary role in the free-surface deformation. In the film dynamics, the Reynolds number only starts to matter as the film thickness gets closer to the viscous-gravity length scale l_ν.

In this section, the effect of the Reynolds number in the highly viscous regime ($Re \leq 1$) is investigated at a constant Froude number of 1 (fixed gravity and velocity). In this case the Reynolds number is expected to have a significant effect on the free-surface deformation. Moreover, the lower Reynolds number also results in an almost linear dynamics (very weak effect of the nonlinear convective terms) and therefore to highly symmetrical flows. This provides the possibility of a direct study of the vertical filament dynamics that drains and is stretched below the cylinder[10].

The specific nomenclature for this particular analysis is given in Fig. 5.42, where the minimal width of the filament below the cylinder is noted w. Unlike in the previous sections, all length scales are here non-dimensionalized by the cylinder *diameter*, except for the Froude number that is still defined using the cylinder radius a. All these non-dimensional parameters are summarized in Table 5.8. At the time where $w = 1$, the characteristic length L_w of the filament can be defined as $y_c|_{w=1} + 1$. The following analysis attempts to relate this characteristic filament length L_w to the Reynolds number.

Numerical setup

At such low Reynolds numbers, the effect of viscosity does not only impact the boundary layers but also the entire flow. Consequently, the boundaries have a significant impact on the flow. In order

[10]By direct, it is meant that only one simulation per case is necessary because of the wake stability, while statistics would be required to study the entrained fluid at higher Reynolds number for which random asymmetries are unavoidable.

Cylinder diameter	D	1
Initial depth of the cylinder	d	1
Pool depth	H	51
Half pool width	W	50
Reynolds number	Re_D	$\in [0.01, 1]$
Froude number	Fr	1
Minimal mesh size	L^*_{min}	0.0075
Maximal mesh size	L^*_{max}	2.25
Time step size	Δt	0.045

Table 5.8: Simulation parameters for the case of the rising cylinder in the highly viscous regime ($Re_D \leq 1$). All quantities are non-dimensionalized by the cylinder velocity U and diameter D, except for the Froude number which uses the radius a instead of D (as in all previous sections). The Reynolds numbers considered are in the set $\{0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1\}$.

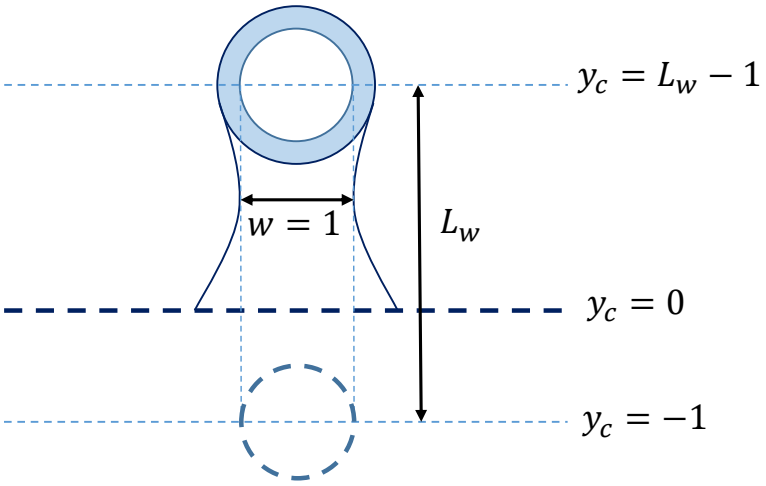


Figure 5.42: Definition of key quantities for the analysis of the wake filament at low Reynolds number, in complement to those defined in Fig. 5.1a. All lengths are here non-dimensionalized using the cylinder diameter D.

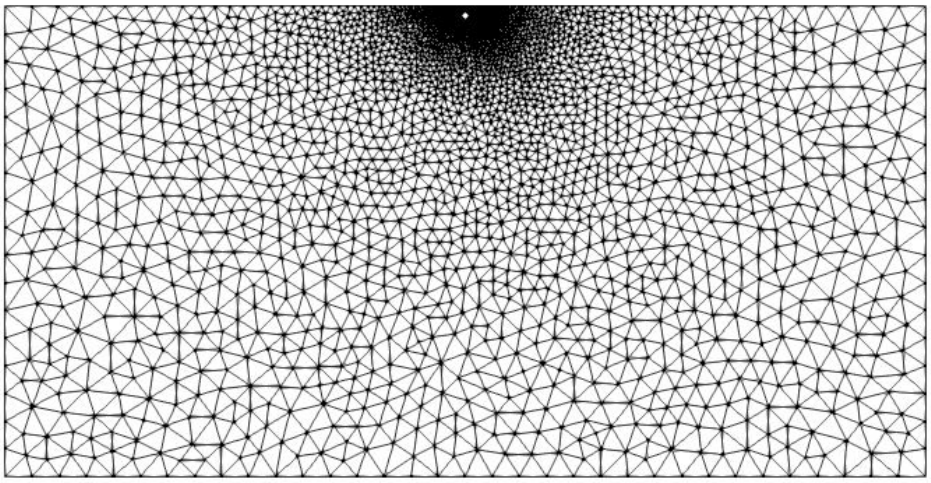


Figure 5.43: Initial mesh for the rising cylinder in an highly viscous fluid. The corresponding simulation parameters are given in Table 5.8.

to mitigate this effect, a very large domain (see the simulations parameters in Table 5.8) is used, as illustrated in Fig. 5.43 ($W = 50$ and $H = 51$). The computational cost remains nonetheless limited owing to mesh adaptation, as very large elements ($L^*_{\max} = 2.25$) can be used near the tank walls. Moreover, since we are not interested in the effect of the pool walls, a free-slip boundary condition is imposed on each walls, such that no shearing occurs there[11].

A benefit of simulating highly viscous fluid flows is that there is no boundary layer per se (the viscous region includes the entire domain). There is thus less stringent requirements in terms of the spatial resolution, as small flow features are either nonexistent or very quickly diffused. For that reason, a rather large minimal target mesh size ($L^*_{\min} = 0.0075$) is used, compared to the other cases at higher Reynolds number[12]. However, the mesh size must still be small enough to actually capture the wake filament below the rising cylinder and the film around the cylinder.

The target mesh size is imposed through a linear progression from $L^*_{\min}$ at the cylinder surface, to $L^*_{\max}$ at a distance of $10D$ from the cylinder center. Because of the high shearing occurring everywhere, no solution-based refinement is used here, as it would lead to unnecessary refinement in many regions of the computational domain[13]. Furthermore, the mesh coarsening is deactivated during the cylinder rise, such that the film developing below the cylinder remains well refined during its rise.

Finally, because of the absence of boundary layer, the time step can be slightly larger than what is usually expected for flows with boundary layers (see Fig. 3.19 in Section 3.3). It is set here to $6\frac{L^*\min}{U} = 0.045\frac{D}{U}$, which is sufficient to simulate all flows without convergence issues until $y_c = 7$, as illustrated in Fig. 5.44. Convergence issues start to appear for larger simulation times, all the earlier at higher Reynolds number. All simulation parameters are summarized in Table 5.8.

Results

Figure 5.45 shows the free-surface nodes for the different Reynolds numbers considered, at the times when the cylinder successively reaches the positions $y_c = 0$ (Fig. 5.45a), $y_c = 1$ (Fig. 5.45b), $y_c = 2$ (Fig. 5.45c) and $y_c = 3$ (Fig. 5.45d). As expected, the amount of fluid entrained around the cylinder is larger at smaller Reynolds number because of the increased friction. While a wake filament starts to stretch below the cylinder for the higher Reynolds numbers ($Re_D \geq 0.5$) in the considered range of y_c (see Figs. 5.45c and 5.45d), this is not the case for smaller Reynolds numbers, for which this filament

[11] While this is not representative of the real highly viscous flow in a tank of finite length, it is likely closer to what one should expect in a pool of infinite size, which is more consistent with the fact that the effects of the walls are not studied in the present analysis.

[12] Note that here the mesh sizes are non-dimensionalized by the cylinder diameter instead of the radius, such that a factor of two should be applied to the present mesh sizes for a fair comparison with resolutions used in the precedent sections.

[13] It is nonetheless possible to adapt the solution-based criterion, Eq. (3.23), to this particular case.

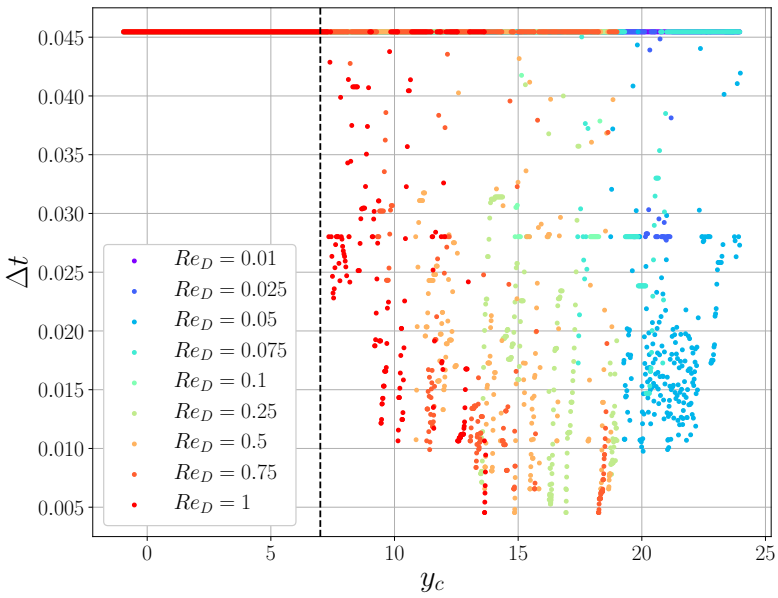


Figure 5.44: Evolution of the time step size as the simulation proceeds (i.e., increasing y_c) for the highly viscous fluid flow around a cylinder rising toward a free surface. To ensure convergence of the nonlinear solver (Picard algorithm), the time step must be decreased sooner for larger Reynolds number because the wake filament thinning occurs more rapidly. For all Reynolds numbers considered, no time step size adaptation is required until $y_c > 7$.

forms at later times. To quantify this effect, the value of L_w is plotted in Fig. 5.46b with respect to the Reynolds number, where a quasi-power fit of the data can be performed. In Fig. 5.46a, the corresponding free-surface deformation is shown for all the Reynolds numbers considered. In addition the evolution of the minimal wake filament thickness w (continuous lines), once it has reached the value $w = 1$, is shown in Fig. 5.47a in a log-log plot, where its rate of decrease follows an arithmetic law of the form[14]

$$w(t) = C(t + L_0)^{-k}\,, \tag{5.18}$$

that is fitted (dashed lines) to the data (fitting has been performed on the time range during which $w \in [0.04, 0.4]$). The two calibration parameters C and k are plotted in Fig. 5.48 for the different Reynolds numbers, with L_0 fixed to $0.9L_w(Re_D)$, for which a correlation has been found. Figure 5.48 suggests a logarithmic dependence on the Reynolds number, as the points are almost aligned.

The behavior described by Eq. (5.18) can be easily interpreted by considering mass conservation, leveraging some basic assumptions regarding the flow. As illustrated in Fig. 5.47b, the vertical velocity component u_y at a fixed time t increases with y (at least locally near the position where the wake filament is the thinnest). Moreover, as the cylinder rises, the velocity gradient $\frac{\partial u_y}{\partial y}$, and therefore u_y, decreases at a fixed position y (the velocity u_y increases from the bath where it is more or less 0 to the cylinder where it is equal to 1). As a consequence, the velocity evolves inversely proportionally to the film elongation: $u_y \propto \frac{1}{t+L_0}$, where L_0 is a constant to determine, and t is the (non-dimensional) time which has its 0 when $w = 1$. In mathematical terms, one therefore has

$$u_y(y,t) \simeq k\frac{y}{t + L_0}\,, \tag{5.19}$$

from which it directly follows from mass conservation that

$$u_x(x,t) \simeq -k\frac{x}{t + L_0}\,. \tag{5.20}$$

In other words, the wake filament thinning corresponds to an extensional flow in the y-direction. Replacing x by $w/2$ in the above expression yields

$$\frac{d(w/2)}{d\,t} \simeq -k\frac{w/2}{t + L_0}\,, \tag{5.21}$$

[14] All lengths are normalized by the cylinder diameter D and the time by the characteristic time D/U.

or equivalently,

$$\frac{d\,w}{d\,t} \simeq -k\frac{w}{t+L_0}\,, \tag{5.22}$$

whose solution is given by Eq. (5.18). From there, L_0 can be intuitively related to the initial length of the filament, at the time $t = 0$ (i.e., when $w = 1$). It is therefore related to L_w and the value $L_0 = 0.9\,L_w$ has been found to provide good fits for $w \in [0.04, 0.4]$.

As a conclusion, it is clear that the behavior of the free-surface elevation is strongly affected by the Reynolds number in the highly viscous limit, as it was expected. It has been shown that, on the one hand, the characteristic wake filament length L_w is related to the Reynolds number through a correlation close to a power law. On the other hand, it has been shown that the time evolution of the filament thickness below the cylinder follows a decreasing power law.

Additional cases could have been considered, such as a larger initial depth d for instance. However, the surface deformation is already very large with a rather small release depth of 1 (i.e., only one radius of fluid initially above the cylinder), as illustrated in Fig. 5.45, such that considering higher release depths would not provide much more insight: the qualitative findings, i.e., the shape of the $L_w(Re_D)$ correlation and the power law decrease of w would likely remain the same.

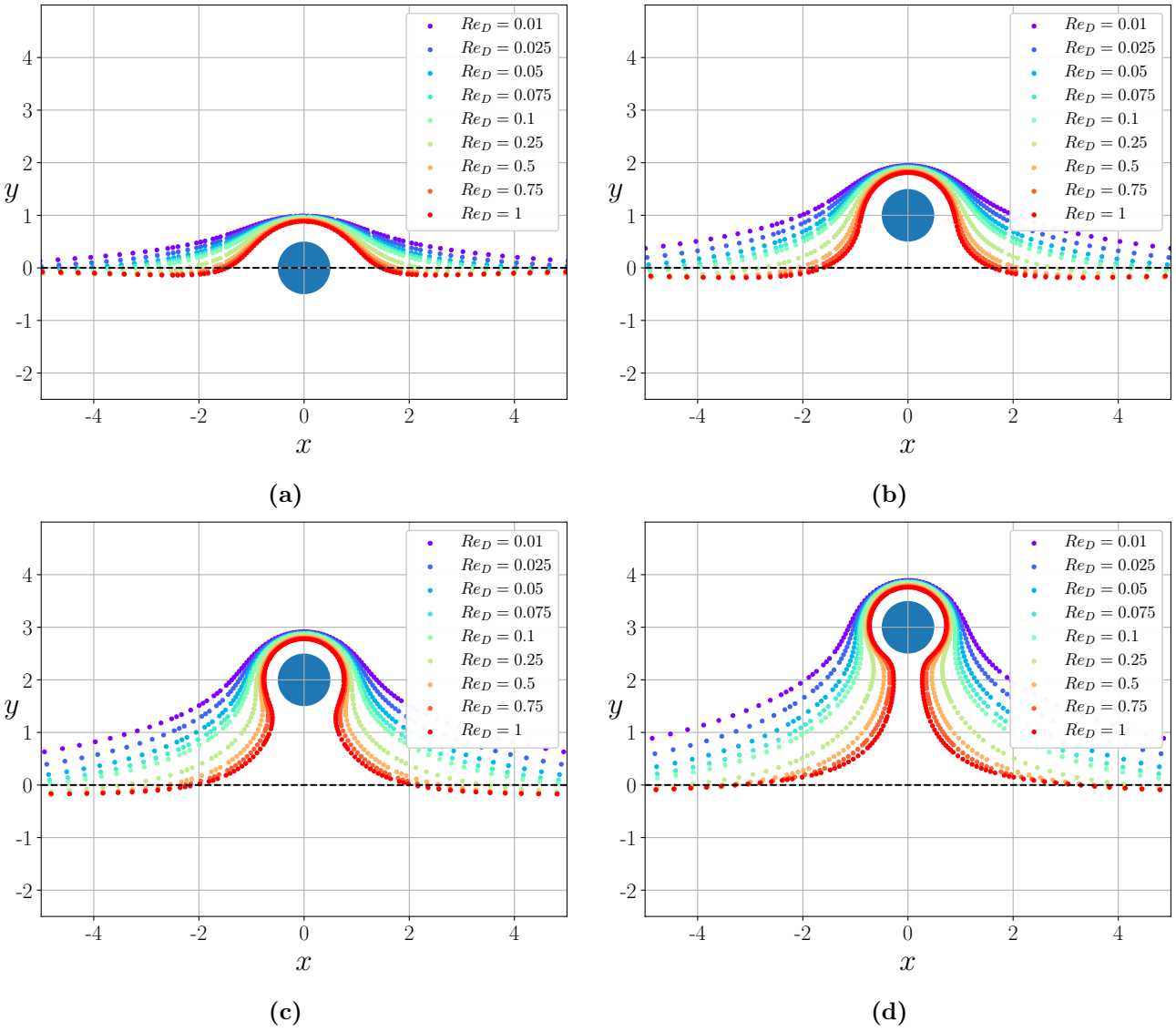
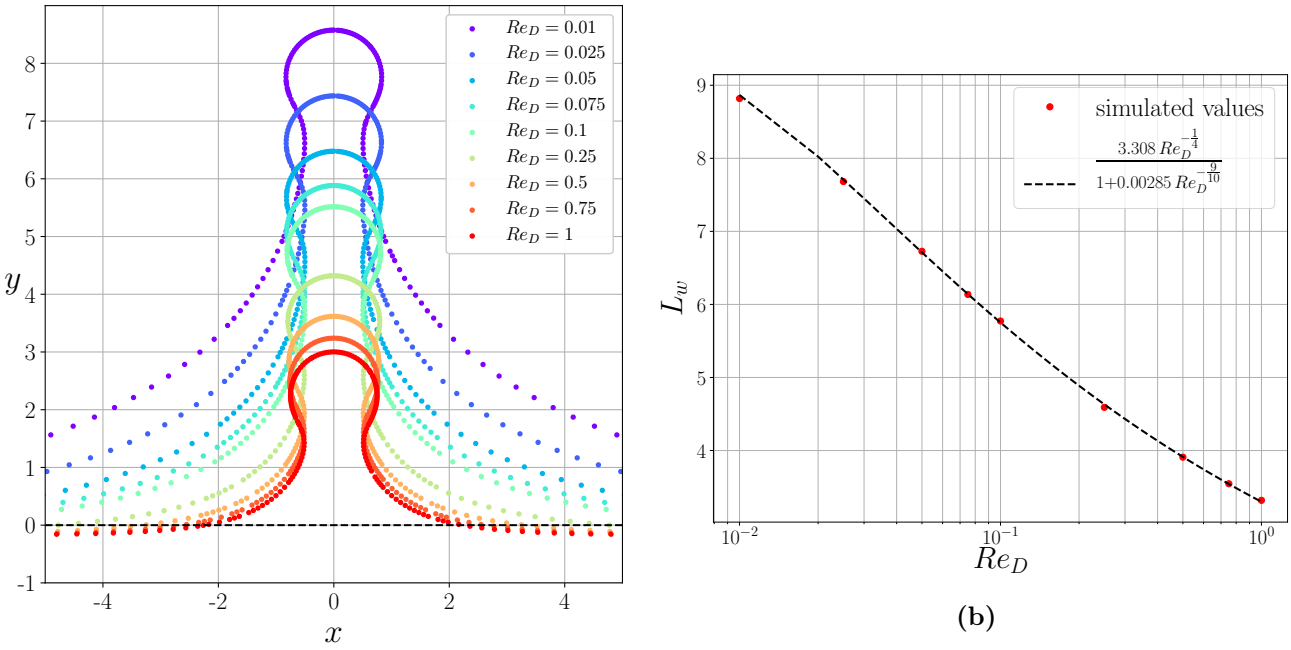
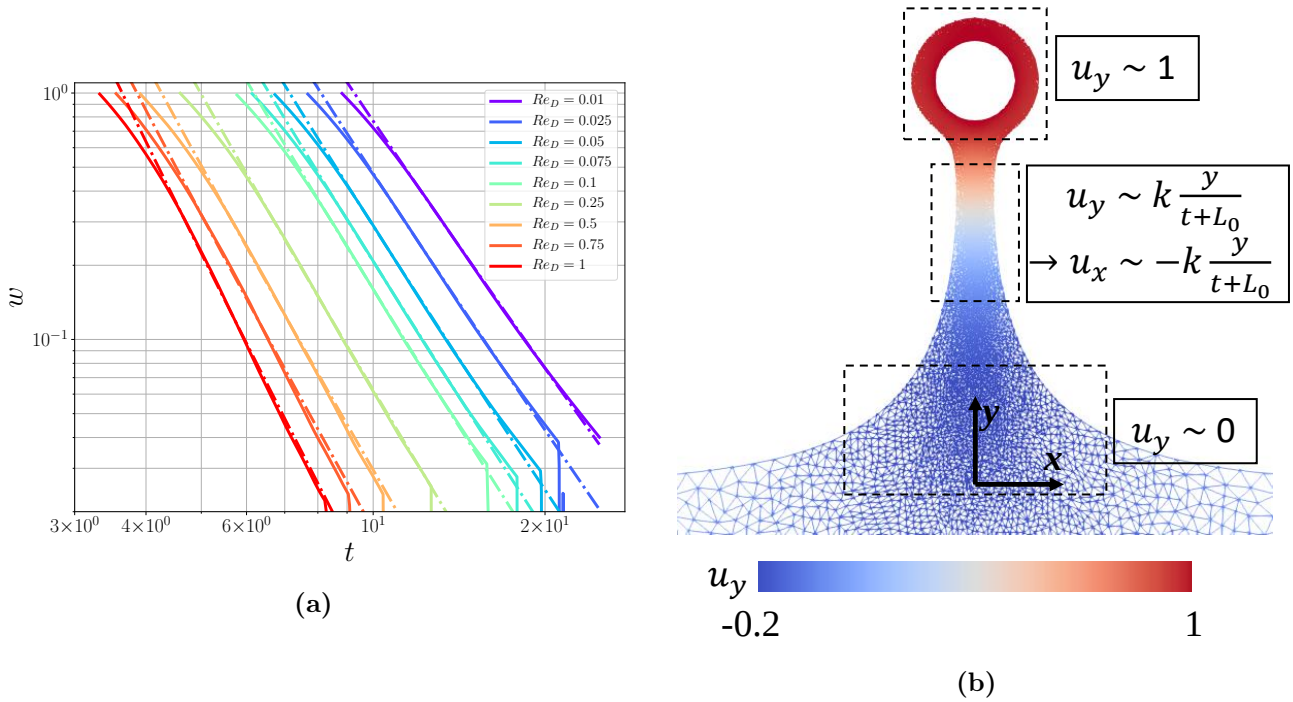


Figure 5.45: Free-surface nodes (i.e., deformation) in the highly-viscous case when the cylinder is at the successive positions $y_c = 0$ (**a**), $y_c = 1$ (**b**), $y_c = 2$ (**c**) and $y_c = 3$ (**d**) for the different Reynolds numbers considered.



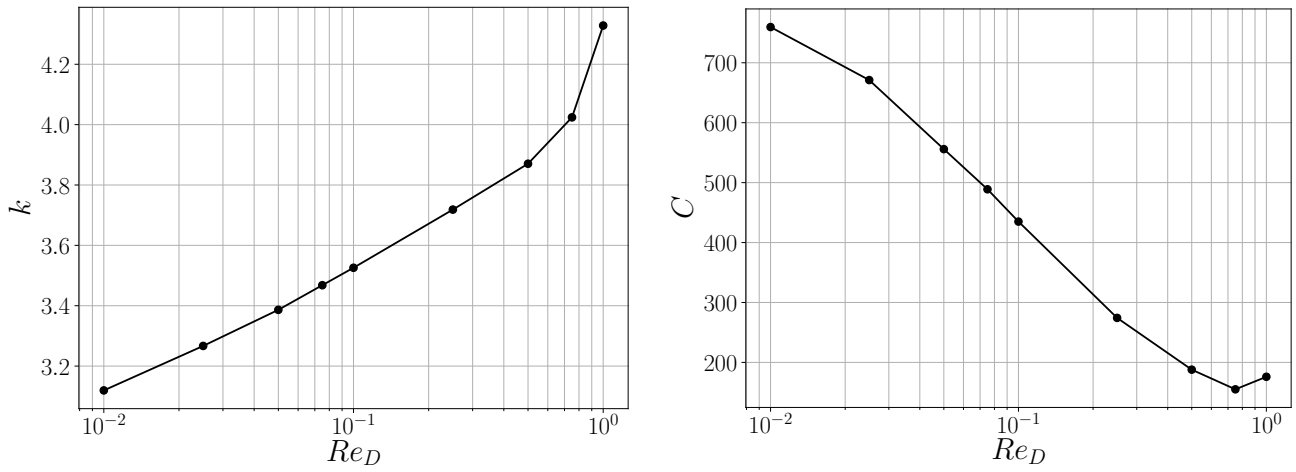
(a) (b)

Figure 5.46:
(**a**) Free-surface nodes (i.e., deformation) when the wake filament minimal thickness reaches $w = 1$ for the different Reynolds numbers considered. (**b**) Corresponding characteristic length L_w of the wake filament below the cylinder when its minimal thickness reaches $w = 1$. An empirical correlation with Re_D successfully describes L_w for $Re_D \in [0.01, 1]$.



(a) (b)

Figure 5.47:
(**a**) Evolution of the wake filament thickness w (continuous lines), in a log-log plot, starting from the time when it is equal to 1, for the different Reynolds numbers. The decrease of w is well described by the power law (dash-dotted lines) given by Eq. (5.18). (**b**) Mesh wireframe colored by the vertical velocity component u_y at $Re_D = 0.25$ and $y_c = 5$. The vertical velocity in the wake filament varies approximately between 0 and 1 over a length scale that increases with time, such that the vertical velocity gradient in the y-direction slowly decreases with time. Note that the velocity is below 0 in some regions, meaning that the wake filament shrinks under the effect of both the rising cylinder and the large amount of fluid close to the initial free surface, falling back into the bath.



(a) **(b)**

Figure 5.48: Calibration parameters k (**a**) and C (**b**) of the wake filament thinning model given by Eq. (5.18) with $L_0 = 0.9L_w(Re_D)$, used to obtain the fitted curves (dash-dotted lines) in Fig. 5.47a, as a function of the Reynolds number. The log-lin plots suggests that $k \propto \ln(Re_D)$ and $C \propto -\ln(Re_D)$.

5.3 The rising sphere

The case of a rising axisymmetric sphere is interesting because it highlights the fundamental differences between flows around infinitely long bodies (2D) and bodies with a limited extent (3D).

The simulations for a rising sphere are performed with the same parameters as for the cylinder, summarized in the first column of Table 5.4 (set F1), using the nomenclature of Fig. 5.1b. The results for $h(y_c)$ and $h^*(Fr)$ are respectively given in Figs. 5.49 and 5.50. In the second figure, the results are directly compared to those obtained with the cylinder. Globally, the same qualitative observations as for the cylinder case can be made. For values of y_c around -1 (interface crossing), an exponential decrease of $h(y_c)$ is observed, followed by a decrease of its decreasing rate due to viscous forces, down to the viscous-gravity length scale l_ν. The difference with the cylinder is that the exponential decrease is more abrupt, due to the fact that the fluid can flow along two space dimensions instead of one, as it is discussed in Chapter 6. As a consequence, the transition to the viscous regime is more abrupt than for the cylinder case, and the viscous-gravity length scale, l_ν, is reached sooner. These qualitative similarities are also observed for the free-surface deformation $h^*(Fr)$ when the sphere reaches the initial interface, as shown in Fig. 5.50 (**b**). The surface deformation at a given Fr is expectedly smaller than for the cylinder case, for the same reasons as mentioned above. Nevertheless, a similar fit based on the Lambert $\mathcal{W}$ function, Eq. (5.10), can be performed with satisfying agreement for Froude numbers below 1.

In conclusion, the qualitative behavior of the flow around the axisymmetric sphere is very similar to that around the 2D cylinder, despite quantitative differences arising from the different numbers of space dimensions. Quite intuitively, very similar models should be derivable for both cases, in which the effect of the different geometries, and the different numbers of dimensions, should be highlighted. This is the subject of the next chapter. Finally it should be mentioned that, although much fewer analyses have been performed for the case of a sphere, because fewer simulations and experimental measurements were available, the modeling effort in Chapter 6 considers both almost equally.

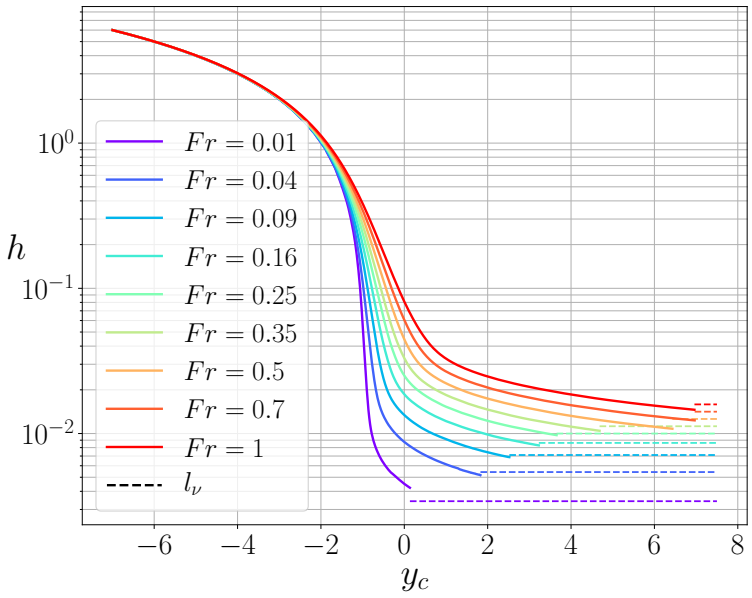


Figure 5.49: Height h of the liquid column above a rising axisymmetric sphere as a function of its vertical position y_c for different Froude numbers ($Re_D = 1000$, initial depth $d = 7$). The horizontal dashed lines indicate the viscous-gravity length scale l_ν.

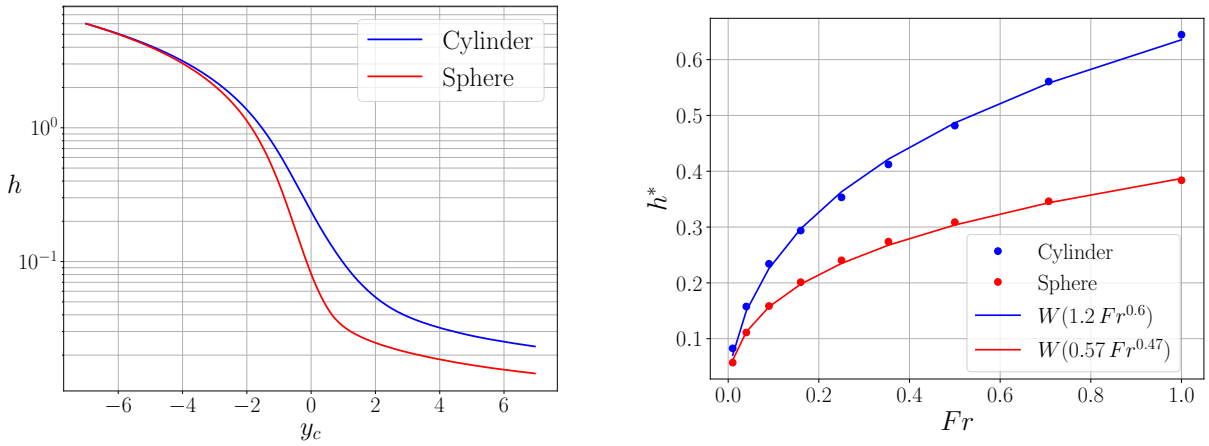


Figure 5.50: Comparison of the film drainage around a rising axisymmetric sphere and an infinitely long cylinder for a Reynolds number $Re_D = 1000$ and an initial depth $d = 7$. (**a**) Height h of the liquid column above the cylinder/sphere as a function of its vertical position y_c, for a Froude number $Fr = 1$. (**b**) Free-surface deformation $h^* = h(y_c = -1)$ when the cylinder/sphere reaches the initial interface as a function of the Froude number (dots) and best fit with the Lambert $\mathcal{W}$ function (continuous line).

5.4 Conclusions

In this chapter, the flow around a cylinder pulled out of liquid bath has been considered. First, through a mesh convergence analysis, the minimum resolution required was determined such that a smooth variation of the friction coefficient c_f is observed, and remains almost unchanged when the mesh is further refined. Then, it has been shown that the time step should be set based on a compromise between mesh deformation and pressure oscillations: a too large time step leads to large numerical mass conservation errors and convergence issues while a too small time step leads to pressure oscillations, because of the remeshing. This analysis therefore enabled to select appropriate space and time discretizations for the study.

Afterwards, the physical features of the considered flow were investigated. The first part of the study considered the bulge formation above the cylinder, and the subsequent film drainage around it. In particular, the bulge height at the time when the top of the cylinder reaches the initial free surface ($y_c = -1$), h^*, was shown to follow a Lambert $\mathcal{W}$ law for $Fr \leq 1$ (the $\mathcal{W}$ function is applied on an algebraic expression of the Froude number Fr). Note that according to Vincent et al. [123], a logarithmic law can be used for larger Froude numbers ($Fr \in [1 - 10]$). In addition, Vincent et al. have shown that h^* results from two contributions: one "wake-independent" contribution, and one contribution $\delta\, h^*$ scaling as $\sqrt{j^*}$, j^* being the wake area at that time. This scaling has been verified with the present simulations.

Another finding concerning the bulge formation concerns its initial vertical velocity at its apex. Vincent et al. have identified two regimes. An unconfined regime where the cylinder initial depth d is smaller than the half pool width W and a confined regime, where d is larger (or W smaller). They have shown that the initial surge speed at the apex, c^*, was given by two different expressions depending on these regimes. In this work, their findings have been confirmed for both regimes with only a few % of error.

After the bulge formation, the film drains around the cylinder and we can trace the evolution of its thickness h with time or cylinder elevation. Initially, the drainage follows an exponential decrease that is driven by a balance between inertia and gravity forces. When the film gets thinner, viscous effects become relatively more important and the film thickness decreases more slowly, approaching the viscous-to-gravity length scale l_ν. In that regard, results for a rising sphere (in axisymmetric formulation) have been compared to those of the cylinder. The exponential decrease rate appears to be more important

for the sphere, leading to a more abrupt and sooner transition to the viscous regime. The drainage dynamics and the transition between the two regimes are investigated in more details in Chapter 6.

The effects of the free-surface and gravity field on the aerodynamic loads have also been investigated. When the cylinder approaches the free-surface, two additional effects are present compared to similar flows in an infinite domain. At low Froude number, the weight of the bulge tends to increase the pressure above the cylinder and therefore increases the drag, vorticity and skin friction. At high Froude number, the free-surface elevation tends to decrease the pressure above the cylinder, therefore reducing the drag, vorticity and skin friction. Both effects are present and should cancel each other for some Froude number around 1 leading to a flow very similar to that in an infinite domain. On the other hand, when the Froude is very low, the flow is very similar to that obtained when the cylinder approaches a wall (with free-slip BC). While the Froude number is a key parameter governing these effects, the Reynolds number does not play a significant role here. Specifically, the surface deformation has been shown to be only indirectly affected by the Reynolds number, through the size of the wake. By opposition, the impact of the free-surface on the wake formation is almost negligible for the considered range of Froude numbers $Fr \in [\ 0 - 10]$, provided that the release depth is sufficiently large, and the wake dynamics in the presence of a free-surface is almost identical to that observed in infinite domains, until the cylinder gets very close to the free-surface.

By comparing experiments and simulations of the rising cylinder in oil, a good agreement has been observed for the vertical force after the acceleration phase (cylinder was impulsively started in the simulations), provided that the wake remains stable. For the evolution of the film thickness h, a good agreement was also observed, even if some scaling differences are present due to different confinement effects between 2D simulations and 3D experiments [125].

For the evolution of the force after the interface crossing, it has been shown that a maximum peak occurs at different cylinder elevations y_c (larger y_c for larger cylinder velocity), and that the force was proportional to the weight of the annulus of thickness h of the fluid surrounding the cylinder. This maximum peak is due to a suction effect caused by a depression below the cylinder, at the time when the smallest wake thickness along the horizontal direction is about 1.2 times the cylinder radius for Fr between 1 and 5.

From the time this maximum peak occurs at $t_g = 0$ until $t_g \simeq 2$, the evolution of the vertical force only depends on gravity: the velocity of the cylinder determines only the initial amount of fluid around the cylinder at $t_g = 0$.

The case of a rising cylinder at $Re_D <= 1$ was finally investigated for a constant Froude number $Fr = 1$. In this case, the stretching of the filament below the rising cylinder after the interface crossing has been shown to follow a power law evolution, the two calibration parameters of this extensional flow having a logarithmic dependency on the Reynolds number. In addition, the characteristic filament height, i.e., the height at which the minimum horizontal wake thickness equals the cylinder diameter, has been shown to follow a quasi-power algebraic dependency (power law) with respect to the Reynolds number.

The chapter ended by comparing results of the 2D cylinder and the 3D axisymmetric sphere, as discussed above. The similar results between both cases in terms of film drainage motivates the use of a similar strategy for the derivation of models for the film drainage dynamics. This is the subject of Chapter 6.

Chapter 6

Simplified model for the dynamics of the film between a rising cylinder/sphere and a free surface

The purpose of this chapter is to model the evolution of the film thickness above the apex of a cylinder/-sphere crossing a liquid/gaz interface at different Froude and Reynolds numbers. The symmetry of the geometries considered (cylinder/sphere) is leveraged by expressing the conservation laws in a moving frame of reference, using a polar or an axisymmetric spherical coordinate system whose origin is located at the cylinder or the sphere centre, respectively.

The corresponding two-dimensional Navier-Stokes equations, expressed in terms of the radial and azymuthal directions, r and θ, and time t, are then simplified. Because the interest is in predicting the film thickness $h(\theta, t)$ around the apex, the radial dependence of the solution is first eliminated by integrating the equations in this direction from the body surface to the free surface. This approach is often used when the thin film approximation can be made, i.e., when the length scale of the dimension along which the integration is performed is much smaller than the other(s). Typical examples are boundary layer integral methods [130] or lubrication theory [126, 127]. In the thin film limit, the film thickness $h(\theta, t)$ is linked to the radially integrated volume flow rate $q(\theta, t)$ through the thin film continuity equation:

$$\frac{\partial h}{\partial t} = -\frac{\partial q}{\partial \theta} \tag{6.1}$$

The other equations in thin film models are the momentum conservation equations integrated over the film thickness. For some reviews on the subject, the reader is referred to Chang et al. [131], Craster et al. [132] or Ruyer-Quil et al. [128].

The aim here is however to predict the film thickness around the apex for all Froude numbers, i.e., also when h cannot be considered small compared to the cylinder/sphere radius, so that the thin film assumption cannot be invoked. Consequently, Eq. (6.1) must be adapted. The proposed approach is to complement the radially-integrated equations with an assumption on the shape of the velocity profile (similarly to some boundary layer integral methods). Concretely, the assumed velocity profile shape is proposed based on PFEM simulation results. Additionally, another major simplification is obtained by considering a uniform film thickness around the body apex. Using a Taylor expansion in the θ-direction and keeping only the first non-zero terms of the expansion eliminates the explicit θ-dependence of the solution.

It is shown that this approach allows predicting the exponential decrease of $h(t)$ observed in the precedent chapter, when $h(t)$ is of the order of one radius or less. Furthermore, special functions involving the exponential function, to which they simplify in the asymptotic limit of small film thickness h, are introduced to capture the trends for larger values of h. These special functions enable to greatly simplify the mass conservation equations, leaving all the complexity in the momentum conservation equations.

The chapter is divided as follows. First, the key assumption of uniform film thickness is discussed. Then, the non-dimensionalization of the equations is recalled. This is followed by the mathematical development of the model, explicitly detailing the integration of the equations in the radial direction,

the introduction of special functions to partially address the nonlinearity of the problem, and the small angle approximation. These developments are made for the 2D cylinder and the 3D axisymmetric sphere in parallel, the different steps being very similar in both cases. In the subsequent section, the θ-momentum equation evaluated at the free surface is introduced along further assumptions regarding the velocity profile to replace the integrated θ-momentum equation. This enables the derivation of the final time-dependant differential equations. Their numerical resolution and the corresponding required initial conditions are then described. At last, the model is validated by comparing its predictions to the results obtained with other PFEM simulations.

6.1 Main assumption

Unlike most works associated with film dynamics, the thin film assumption is not a priori invoked here. The main justification for this choice is that it allows the study of $h < h^*(Fr)$ for all Froude numbers, whereas the thin film assumption is not valid for large Froude numbers ($h^*(Fr = 1) \simeq 0.63$ for the cylinder and 0.38 for the sphere).

The key assumption of the model relies on the observation, in the literature and in our simulations, that, from the beginning of the interface crossing, the film thickness $h(\theta, t)$ is approximately uniform in the neighborhood of the apex:

$$\frac{\partial h}{\partial \theta} \simeq 0 \,\text{for}\, \theta \in [-\theta_{\max}(t), \theta_{\max}(t)]\,, \tag{6.2}$$

where θ is measured clockwise from the vertical axis and $\theta_{\max}(t) < \pi$ is some increasing function of time t, as illustrated in Fig. 6.1. In other words, when the body is sufficiently close to the free surface, the evolution of the film thickness at the apex, $h(\theta = 0, t)$, is representative of the evolution of the film thickness $h(\theta, t)$ in its neighborhood. Moreover, the region of uniform film thickness increases with time.

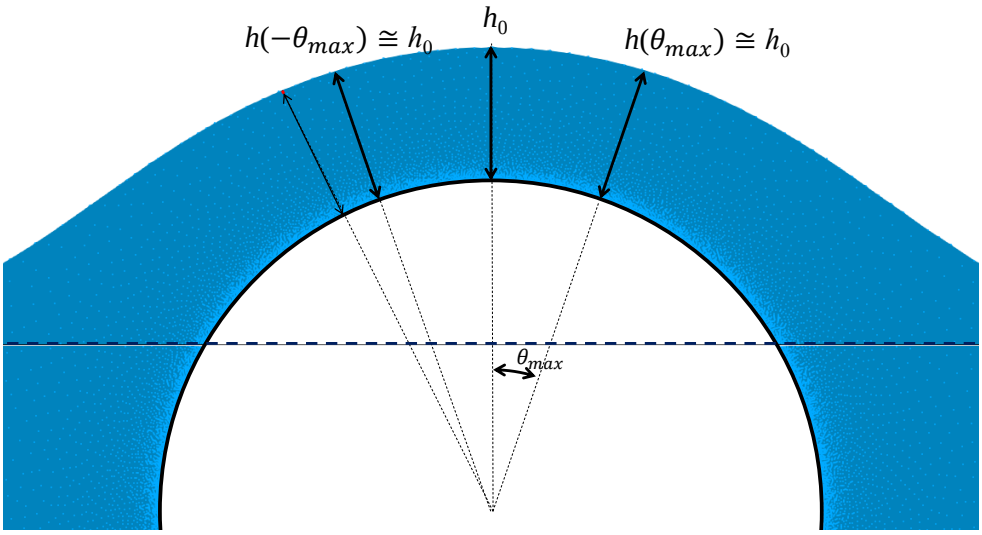


Figure 6.1: Illustration of the range $[-\theta_{\max}(t), \theta_{\max}(t)]$ over which the film thickness can be assumed to be uniform. The example shown is for $Fr = 1$, $Re_D = 1000$, $y_c = 0.25$ and $\theta_{\max}(y_c) = 19°$. The initial free surface is indicated by the dark blue horizontal dashed line. On the left, the film thickness increment with respect to h_0 observed outside the range $[-\theta_{\max}(t), \theta_{\max}(t)]$ is highlighted by the small red segment. Note that even in the range $[-\theta_{\max}(t), \theta_{\max}(t)]$, $h(\theta)$ is not rigorously uniform but the assumption remains a good approximation.

If surface tension is neglected (it has only a non-negligible impact in the late phase of the process), and because the pressure is uniform at the free surface, the uniform film thickness implies that

$$\left.\frac{\partial p}{\partial \theta}\right|_{r=1+h} \simeq 0 \,\text{for}\, \theta \in [-\theta_{\max}(t), \theta_{\max}(t)]\,. \tag{6.3}$$

Note that, in addition, the uniform thickness assumption could also be combined with the vanishing shear stress at the free surface to express the velocity gradient $\partial_r u_\theta|_{r=1+h}$ in terms of other quantities.

However, as discussed and justified later, this gradient is here rather obtained from the assumed velocity profile inside the film.

6.2 Non-dimensionalization of the equations

To identify and study this specific flow, and characterize how it behaves at different Froude and Reynolds numbers, a non-dimensionalization of the equations should be performed. This is done using the following characteristic physical quantities:

- The length scales, such as the radial position and the film and boundary layer thicknesses, are all non-dimensionalized by the radius a of the cylinder/sphere: $r = a\hat{r}$, $\delta = a\hat{\delta}$, $h = a\hat{h}$, where $\hat{\cdot}$ represents a dimensionless quantity.
- The velocity variables are non-dimensionalized by the cylinder/sphere velocity U: $\mathbf{u} = U\hat{\mathbf{u}}$. Note that the choice $\mathbf{u} = \sqrt{gh}\hat{\mathbf{u}}$ could also be used, as the velocity inside the film is larger when the effect of gravity is stronger, but then the Froude number would not explicitly appear in the equations.
- The pressure variable is non-dimensionalized by the factor ρU^2 where ρ is the fluid density: $p = \rho U^2 \hat{p}$.
- The time variable is non-dimensionalized by the ratio a/U: $t = \frac{a}{U}\hat{t}$.

For the remaining sections of this chapter, all variables are non-dimensional and the $\hat{\cdot}$ is dropped to simplify the notation.

It is well-known that, with the above non-dimensionalization, the viscous terms are multiplied by the inverse of the Reynolds number Re_a and the gravity term is replaced by the inverse of the Froude number Fr, with

$$Fr = \frac{U^2}{ga}\,, \tag{6.4}$$

$$2Re_a = \frac{UD}{\nu} = Re_D\,, \tag{6.5}$$

where g is the gravity acceleration, ν the kinematic viscosity of the fluid and $D = 2a$ the cylinder/sphere diameter.

6.3 Mathematical development of the simplified model

The first step of the present mathematical derivation consists in expressing the Navier-Stokes equations either in polar (for the 2D cylinder) or in axisymmetric spherical coordinates (for the 3D axisymmetric sphere), such that quantities are expressed as functions of the spatial variables r and θ and the time variable t. Afterwards, the variable r is eliminated from the equations by integrating in r over the film thickness $h(\theta, t)$. Note that only the mass conservation and the θ-momentum equation are considered. The integrated r-momentum equation would lead to other unclosed terms that would be difficult to deal with, but this option could nevertheless be considered for future work. Doing so, one is left with two equations for two variables: the film thickness $h(\theta, t)$ and the volume flow rate $q(\theta, t)$ along θ, as illustrated in Fig. 6.2a and 6.2b. Nonetheless, these two equations involve unclosed terms which are further discussed in Section 6.4. Eventually, a special composed function is used to simplify the equations, and the limit of small angles θ is considered, such that one is left with only time-dependent quantities describing the dynamics near the cylinder/sphere apex.

6.3.1 Integrated form of the mass and momentum conservation equations

In this section, the integrated non-dimensional equations of mass and momentum conservation are formulated in terms of the volume flow rate $q(\theta, t)$ and the thickness $h(\theta, t)$. They are expressed in polar and axisymmetric spherical coordinates for the 2D and the 3D cases, respectively.

First, the volume flow rate is defined as

$$q(\theta,t) = \int_1^{1+h(\theta,t)} u_\theta \,\mathrm{d}r \quad \text{in 2D}\,, \tag{6.6}$$

$$q(\theta,t) = \int_1^{1+h(\theta,t)} 2\,\pi\, r \sin\theta\, u_\theta \,\mathrm{d}r \quad \text{in 3D}\,, \tag{6.7}$$

where u_θ is the fluid velocity along θ, as shown in Figs. 6.2a and 6.2b.

Then, the mass conservation equations,

$$\frac{1}{r}\frac{\partial r\, u_r}{\partial r} + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} = 0 \quad \text{in 2D}\,, \tag{6.8}$$

$$\frac{1}{r^2}\frac{\partial r^2\, u_r}{\partial r} + \frac{1}{r\sin\theta}\frac{\partial (u_\theta \sin\theta)}{\partial \theta} = 0 \quad \text{in 3D}\,, \tag{6.9}$$

after multiplication respectively by r and $2\,\pi\, r^2 \sin\theta$, can be integrated in r over the thickness $h(\theta,t)$:

$$\left[r\, u_r\right]_1^{1+h} = -\int_1^{1+h(\theta,t)} \frac{\partial u_\theta}{\partial \theta}\mathrm{d}r \quad \text{in 2D}\,, \tag{6.10}$$

$$2\pi\, \sin\theta \left[r^2\, u_r\right]_1^{1+h} = -\int_1^{1+h(\theta,t)} \frac{\partial (2\pi\, r \sin\theta u_\theta)}{\partial \theta}\mathrm{d}r \quad \text{in 3D}\,. \tag{6.11}$$

The non-penetration boundary condition at the cylinder/sphere surface yields $u_r|_{r=1} = 0$, while at the film surface, i.e., at $r = 1 + h$, u_r corresponds to the total derivative of h, such that

$$u_r|_1^{r=1+h} = \frac{\mathrm{d}h}{\mathrm{d}t} = \frac{\partial h}{\partial t} + \frac{u_\theta|_{r=1+h}}{1+h}\frac{\partial h}{\partial \theta}\,. \tag{6.12}$$

These results can be introduced into Eqs. (6.10) and (6.11), where the derivative with respect to θ has been taken out of the integral using Leibniz integral rule, leading to an additional term on the right hand side:

$$(1+h)\left(\frac{\partial h}{\partial t} + \cancel{\frac{u_\theta|_{r=1+h}}{1+h}\frac{\partial h}{\partial \theta}}\right) = -\frac{\partial}{\partial \theta}\left(\int_1^{1+h(\theta,t)} u_\theta \,\mathrm{d}r\right) + \cancel{\frac{\partial h}{\partial \theta}\, u_\theta|_{r=1+h}}\,, \tag{6.13}$$

$$2\pi\, \sin\theta(1+h)^2\left(\frac{\partial h}{\partial t} + \cancel{\frac{u_\theta|_{r=1+h}}{1+h}\frac{\partial h}{\partial \theta}}\right) = -\frac{\partial}{\partial \theta}\left(\int_1^{1+h(\theta,t)} 2\,\pi\, r \sin\theta\, u_\theta \,\mathrm{d}r\right) + \cancel{2\,\pi\,(1+h)\sin\theta \frac{\partial h}{\partial \theta}\, u_\theta|_{r=1+h}}\,. \tag{6.14}$$

Cancelling identical terms on both sides and using respectively the volume flow rate definitions (Eqs. (6.6) and (6.7)) finally yields

$$(1+h)\frac{\partial h}{\partial t} = -\frac{\partial q}{\partial \theta} \quad \text{in 2D}\,, \tag{6.15}$$

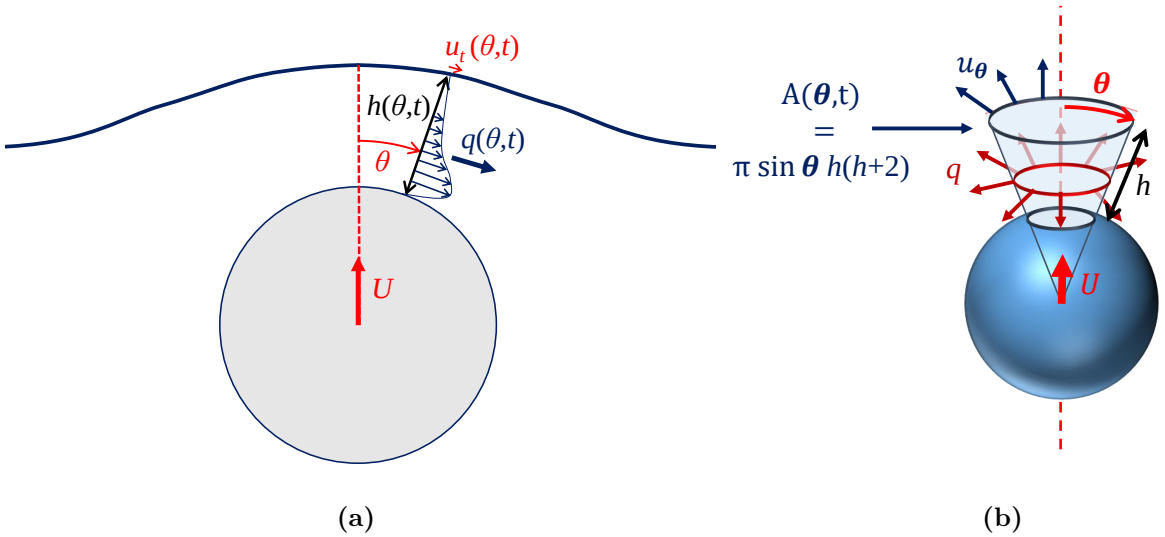
$$2\pi\, \sin\theta(1+h)^2\frac{\partial h}{\partial t} = -\frac{\partial q}{\partial \theta} \quad \text{in 3D}\,. \tag{6.16}$$

These equations look very similar to the thin film continuity equation, Eq. (6.1), and only differ from it by the terms in front of $\partial_t h$. In particular, the first converges towards Eq. (6.1) in the thin film limit.

At this stage, one has one equation for two unknowns, such that one additional equation is needed. To progress further, the θ-momentum equation is considered. Because it is quite complex in either polar or axisymmetric-spherical coordinates, all terms but the temporal one are for now lumped together on the right-hand side. The integration over the film thickness yields

$$\int_1^{1+h(\theta,t)} \frac{\partial u_\theta}{\partial t}\mathrm{d}r = \chi \quad \text{in 2D}\,, \tag{6.17}$$

$$\int_1^{1+h(\theta,t)} \frac{\partial (2\pi\, r \sin\theta u_\theta)}{\partial t}\mathrm{d}r = \chi \quad \text{in 3D}\,, \tag{6.18}$$



(a) **(b)**

Figure 6.2: **(a)** Illustration of the variables of the 2D integral model. The blue solid line represents the free surface and u_t the θ-component of the velocity at the free surface. **(b)** Illustration of the variables of the 3D integral model. The volume flow rate q is the integral of the velocity along θ passing through the truncated cone of area A.

where χ denotes the integrated form of the remaining lumped terms of the momentum equation, i.e., the integrated stresses minus the integrated convective terms. Rewriting these equations in terms of the volume flow rate using Leibniz integral rule leads to

$$\frac{\partial q}{\partial t} - u_t \frac{\partial h}{\partial t} = \chi \quad \text{in 2D}\,, \tag{6.19}$$

$$\frac{\partial q}{\partial t} - 2\pi \sin\theta u_t (1+h) \frac{\partial h}{\partial t} = \chi \quad \text{in 3D}\,, \tag{6.20}$$

where u_t is the θ-component of the velocity at the free surface:

$$u_t(\theta, t) \equiv u_\theta|_{r=1+h}\,. \tag{6.21}$$

It is useful to introduce the average azimuthal velocity

$$\overline{u_\theta}(\theta, t) = \frac{q(\theta, t)}{h(\theta, t)} \quad \text{in 2D}\,, \tag{6.22}$$

$$\overline{u_\theta}(\theta, t) = \frac{q(\theta, t)}{A(\theta, t)} \quad \text{in 3D}\,. \tag{6.23}$$

where the flow cross-section areas are, respectively,

$$h(\theta, t) = \int_1^{1+h} \mathrm{d}r \quad \text{in 2D}\,, \tag{6.24}$$

$$A(\theta, t) = \int_1^{1+h} 2\,\pi\, r \sin\theta \mathrm{d}r = \pi\, h\,(h+2) \sin\theta \quad \text{in 3D}\,. \tag{6.25}$$

Secondly, one can introduce a so-far unknown radial variation parameter α that relates the azimuthal velocity at the film surface to the averaged velocity:

$$\alpha(\theta, t) = \frac{u_t - \overline{u_\theta}}{1+h}. \tag{6.26}$$

Equations (6.19) and (6.20) can be rewritten using these new quantities. After some manipulations, one obtains

$$\frac{\partial\left(\overline{u_\theta}\right)}{\partial t} = \overline{\chi} + \alpha\left(\frac{1+h}{h}\right)\frac{\partial h}{\partial t} \quad \text{in 2D}\,, \tag{6.27}$$

$$\frac{\partial\left(\overline{u_\theta}\right)}{\partial t} = \overline{\chi} + \alpha\left(\frac{2(1+h)^2}{h(2+h)}\right)\frac{\partial h}{\partial t} \quad \text{in 3D}\,, \tag{6.28}$$

where $\overline{\chi}$ is the average of the stresses and convective terms in the film:

$$\overline{\chi} = \frac{\chi}{h} \quad \text{or} \quad \overline{\chi} = \frac{\chi}{A}\,. \tag{6.29}$$

6.3.2 Geometrical ansatz

One has now two nonlinear partial differential equations, Eqs. (6.15) and (6.27) in the 2D case and Eqs. (6.16) and (6.28) in the axisymmetric 3D case, for the two unknowns $q(\theta,t)$ (or $\overline{u_\theta}(\theta,t)$) and $h(\theta,t)$. Nevertheless, it is important to emphasize that the two unclosed terms $\overline{\chi}(\theta,t)$ and $\alpha(\theta,t)$ still need to be addressed. Before discussing this issue, additional mathematical manipulations and approximations are performed to further simplify the equations.

First, the following ansatz is leveraged, which consists in expressing $h(\theta,t)$ in terms of some special functions, to help cancelling the nonlinear terms:

$$h(\theta,t) = \mathcal{W}(\exp\left(-f(\theta,t)\right)) \quad \text{in 2D}\,, \tag{6.30}$$

$$h(\theta,t) = \mathcal{F}(\exp\left(-f(\theta,t)\right)) \quad \text{in 3D}\,, \tag{6.31}$$

where $\mathcal{W}(x)$ and $\mathcal{F}(x)$ are functions defined by their reciprocal:

$$\mathcal{W}^{-1}(x) = x\mathrm{e}^x\,, \tag{6.32}$$

$$\mathcal{F}^{-1}(x) = \mathrm{e}^{2x}\frac{x}{x+2}\,. \tag{6.33}$$

In particular, $\mathcal{W}(x)$ is known in the literature as the Lambert $\mathcal{W}$ function, or product logarithm. The derivative of the functions $\mathcal{W}(x)$ and $\mathcal{F}(x)$ are given respectively by

$$\frac{\mathrm{d}\mathcal{W}(x)}{\mathrm{d}x} = \frac{1}{x}\frac{\mathcal{W}(x)}{\mathcal{W}(x)+1}\,, \tag{6.34}$$

$$\frac{\mathrm{d}\mathcal{F}(x)}{\mathrm{d}x} = \frac{1}{x}\frac{\mathcal{F}(x)(\mathcal{F}(x)+2)}{2(\mathcal{F}(x)+1)^2}\,. \tag{6.35}$$

The relation between $\partial_t h$ and $\partial_t f$ is obtained by deriving expressions (6.30) and (6.31) using the chain rule to obtain

$$\frac{\partial h}{\partial t} = -\left(\frac{h}{h+1}\right)\frac{\partial f}{\partial t} \quad \text{in 2D}\,, \tag{6.36}$$

$$\frac{\partial h}{\partial t} = -\left(\frac{h(h+2)}{2(h+1)^2}\right)\frac{\partial f}{\partial t} = -\left(\frac{h}{h+1}\right)\left(\frac{2+h}{2+2h}\right)\frac{\partial f}{\partial t} \quad \text{in 3D}\,. \tag{6.37}$$

This result can be used to simplify the mass conservation equations (6.15) and (6.16), which yields

$$h\frac{\partial f}{\partial t} = \frac{\partial q}{\partial \theta} \quad \text{in 2D}\,, \tag{6.38}$$

$$A\frac{\partial f}{\partial t} = \frac{\partial q}{\partial \theta} \quad \text{in 3D}\,. \tag{6.39}$$

On the other hand, the momentum conservation equations (6.27) and (6.28) also simplify to

$$\frac{\partial \overline{u_\theta}}{\partial t} = \overline{\chi} - \alpha\frac{\partial f}{\partial t} \quad \text{in 2D}\,, \tag{6.40}$$

$$\frac{\partial \overline{u_\theta}}{\partial t} = \overline{\chi} - 2\alpha\frac{\partial f}{\partial t} \quad \text{in 3D}\,. \tag{6.41}$$

Equations (6.38) and (6.39) can finally be manipulated to explicitly involve the derivative of the mean velocity $\overline{u_\theta}$,

$$\frac{\partial f}{\partial t} = \frac{\partial \overline{u_\theta}}{\partial \theta} + \overline{u_\theta}\frac{1}{h}\frac{\partial h}{\partial \theta} \quad \text{in 2D}\,, \tag{6.42}$$

$$\frac{\partial f}{\partial t} = \frac{\partial \overline{u_\theta}}{\partial \theta} + \overline{u_\theta}\frac{1}{A}\frac{\partial A}{\partial \theta} \quad \text{in 3D}\,, \tag{6.43}$$

where the equation for the axisymmetric 3D case can be rewritten as

$$\frac{\partial f}{\partial t} = \frac{\partial \overline{u_\theta}}{\partial \theta} + \overline{u_\theta}\left(\cot\theta + \frac{1}{h(h+2)}\frac{\partial\left(h(h+2)\right)}{\partial \theta}\right)\,. \tag{6.44}$$

It is interesting to note that the equations for the 2D and axisymmetric 3D cases have now the same form.

6.3.3 Small angle approximation

Equations (6.42) and (6.43) are now almost linear, only the last term on the right-hand side remains nonlinear. This last remaining nonlinearity can be eliminated by considering the limit of small angles θ. One first assumes that the unknown functions ($\overline{u_\theta}$, q, h, $\overline{\chi}$, $\alpha\,, f$) are analytical, which means that they are locally infinitely differentiable, both in θ and t, such that they can be written as a Taylor series,

$$\xi(\theta, t) = \sum_i \sum_j a_{i,j}\theta^i t^j\,, \tag{6.45}$$

where ξ represents any of the variables and the coefficients $a_{i,j}$ depend on the cross-derivatives of i^{th} order in θ and j^{th} order in t. However, because of the symmetry of the problem, some of the unknown variables are odd with respect to θ ($\overline{\chi}$, $\overline{u_\theta}$, q, α, A), and can therefore be written in the form

$$\xi_{\text{odd}}(\theta, t) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{2i+1,j}\theta^{2i+1}t^j\,, \tag{6.46}$$

while the others are even (h,f) and can be written in the form

$$\xi_{\text{even}}(\theta, t) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} a_{2i,j}\theta^{2i}t^j\,. \tag{6.47}$$

The problem is now further simplified by considering the equations near the position $\theta = 0$ to obtain the evolution of the film thickness just above the apex of the symmetrical body. Considering the almost uniform film thickness around the apex, only the zeroth order and first order terms in θ are kept for the odd and even expansions given by Eqs. (6.46) and (6.47). Specifically,

$$\xi_{\text{odd}}(\theta, t) \simeq \left(\sum_{j=0}^{\infty} a_{1,j}t^j\right)\theta = \xi_1(t)\theta \tag{6.48}$$

for the odd functions and

$$\xi_{\text{even}}(\theta, t) \simeq \left(\sum_{j=0}^{\infty} a_{0,j}t^j\right) = \xi_0(t) \tag{6.49}$$

for the even functions.
Using these approximations in the vicinity of $\theta = 0$ leads to

$$q(\theta, t) \simeq q_1(t)\theta\,, \tag{6.50}$$

$$\overline{u_\theta}(\theta, t) \simeq \overline{u_\theta}_1(t)\theta\,, \tag{6.51}$$

$$\overline{\chi}(\theta, t) \simeq \overline{\chi}_1(t)\theta\,, \tag{6.52}$$

$$\alpha(\theta, t) \simeq \alpha_1(t)\theta\,, \tag{6.53}$$

$$h(\theta, t) \simeq h_0(t)\,, \tag{6.54}$$

$$f(\theta, t) \simeq f_0(t)\,, \tag{6.55}$$

$$A(\theta, t) \simeq A_1(t)\theta = \pi h_0(h_0+2)\theta\,. \tag{6.56}$$

Note that, because h is even by symmetry, one has

$$\lim_{\theta\to 0}\frac{\partial h(\theta,t)}{\partial\theta}=0\,. \tag{6.57}$$

As pointed out in the literature and at the beginning of Section 6.1, this relation should also be valid for an increasing range of angle θ around the apex, which is our key assumption. This means that in the mathematical developments below, terms of $\mathcal{O}(\theta^2)$ and higher can be neglected for all variables (i.e, not only for h). In the limit $\theta\to 0$, the mass conservation equations, Eqs. (6.42) and (6.43), thus simplify[1] to

$$\frac{\partial f_0}{\partial t}=\overline{u_{\theta 1}}(t)\quad \text{in 2D}\,, \tag{6.58}$$

$$\frac{\partial f_0}{\partial t}=2\overline{u_{\theta 1}}(t)\quad \text{in 3D}\,. \tag{6.59}$$

The same can be done for Eqs. (6.40) and (6.41), which have been divided by θ to avoid the singularity, to have

$$\frac{\partial\overline{u_{\theta 1}}}{\partial t}=\overline{\chi}_1-\alpha_1\frac{\partial f_0}{\partial t}\,, \tag{6.60}$$

for both 2D and 3D cases. The variable $\overline{u_{\theta 1}}(t)$ can at last be eliminated to obtain the final differential equations

$$\frac{\partial^2 f_0}{\partial t^2}=\overline{\chi}_1-\alpha_1\frac{\partial f_0}{\partial t}\quad \text{in 2D}\,, \tag{6.61}$$

$$\frac{\partial^2 f_0}{\partial t^2}=2\,\overline{\chi}_1-2\,\alpha_1\frac{\partial f_0}{\partial t}\quad \text{in 3D}\,. \tag{6.62}$$

These equations are unfortunately unclosed, as the terms $\overline{\chi}_1$ and α_1 are still unknown.

To summarize, the use of a geometrical ansatz enabled to get simpler forms of the radially integrated equations of continuity and momentum conservation along θ, for both the 2D cylinder and the axisymmetric sphere. While the derived equations are formally valid in the small angle limit **without any assumption on the flow**, they are assumed to hold for an increasing range of angle θ around the apex, which is the main assumption of the present model (see Section 6.1). In summary, the model is at this stage represented by the following equations, in which the terms in red remain unclosed (they are discussed in the next section):

	2D cylinder	3D axisymmetric sphere	
Mass conservation :	$\dfrac{\partial f_0}{\partial t}=\overline{u_{\theta 1}}(t)$	$\dfrac{\partial f_0}{\partial t}=2\overline{u_{\theta 1}}(t)$	(6.63)
Momentum conservation :	$\dfrac{\partial^2 f_0}{\partial t^2}=\overline{\chi_1}-\alpha_1\dfrac{\partial f_0}{\partial t}$	$\dfrac{\partial^2 f_0}{\partial t^2}=2\,\overline{\chi_1}-2\,\alpha_1\dfrac{\partial f_0}{\partial t}$	(6.64)
Geometric ansatz :	$\mathrm{e}^{-f_0}=h_0\mathrm{e}^{h_0}$	$\mathrm{e}^{-f_0}=\dfrac{h_0}{h_0+2}\mathrm{e}^{2h_0}$	(6.65)
Rel. between derivatives :	$\dfrac{\partial h_0}{\partial t}=-\left(\dfrac{h_0}{h_0+1}\right)\dfrac{\partial f_0}{\partial t}$	$\dfrac{\partial h_0}{\partial t}=-\left(\dfrac{h_0(h_0+2)}{2(h_0+1)^2}\right)\dfrac{\partial f_0}{\partial t}$	(6.66)

[1] In Eq. (6.44), the derivative of h with respect to θ vanishes following the main assumption (Eq. (6.2)). Moreover, the first order term in θ of $\overline{u_{\theta,1}}$ and the cotangent, cancel each other in the small angle approximation: $\lim_{\theta\to 0}(\theta\cot\theta)=1$. This explains the factor 2 instead of 1 on the RHS of the second equation.

6.4 Film model based on the free-surface θ-momentum equation

In the above equations (6.63) to (6.66), the two terms in red remain unclosed. The first one, $\overline{\chi}$, contains the averaged balance between stresses and convective terms along the direction θ. In particular, it includes the pressure gradient along θ. To eliminate the pressure, the r-momentum equation should be used, which however strongly complicates the problem. Moreover, it requires assumptions on how the velocity u_θ varies along θ. The other unclosed term, α, also requires the knowledge of the velocity profile to relate the free-surface velocity u_t to the average velocity $\overline{u_\theta}$. Once this relation is given, α is simply closed by using Eq. (6.26). Alternatively, assuming the relation between u_t and $\overline{u_\theta}$, and differentiating it with respect to time, would relate $\partial_t u_{t,1}$ to $\partial_t \overline{u_\theta}$, such that the inviscid momentum conservation equation along θ (the impact of dismissing the viscous terms in this equation is further discussed below),

$$\frac{\partial u_\theta}{\partial t} = -u_r \frac{\partial u_\theta}{\partial r} - \frac{u_\theta}{r}\frac{\partial u_\theta}{\partial \theta} - u_\theta \frac{u_r}{r} - \frac{\partial p}{\partial \theta} + \frac{\sin\theta}{Fr}\,, \tag{6.67}$$

can be evaluated at the free-surface, as a replacement of Eq. (6.64). At first order in θ, it gives

$$\frac{\partial u_t}{\partial t} = -\left[u_r \frac{\partial u_\theta}{\partial r}\right]_{r=1+h} - \frac{u_t}{r}\frac{\partial u_t}{\partial \theta} - u_t\left[\frac{u_r}{r}\right]_{r=1+h} - \left[\frac{\partial p}{\partial \theta}\right]_{r=1+h} + \frac{\theta}{Fr}\,. \tag{6.68}$$

where u_t is by definition $u_\theta|_{r=1+h}$.[2] Using the results found in Section 6.1 with the assumption of a uniform film thickness, $\partial_\theta p|_{r=1+h}$ vanishes from the equation. If one also notes that, according to the main assumption ($\partial_\theta h \ll 1$),

$$u_r|_{r=1+h} = \frac{dh}{dt} = \frac{\partial h}{\partial t} + \frac{u_t}{1+h}\frac{\partial h}{\partial \theta} \simeq \frac{\partial h}{\partial t}\,, \tag{6.69}$$

Eq. (6.68) becomes

$$\frac{\partial u_t}{\partial t} = -\frac{u_t}{1+h}\frac{\partial u_t}{\partial \theta} - \frac{\partial h}{\partial t}\frac{u_t}{1+h} + \frac{\theta}{Fr} - \frac{\partial h}{\partial t}\left.\frac{\partial u_\theta}{\partial r}\right|_{r=1+h}\,, \tag{6.70}$$

and with the small angle approximation, $u_t(\theta,t) \simeq u_{t,1}(t)\theta$ and $h(\theta,t) \simeq h_0(t)$, one obtains

$$\frac{\partial u_{t,1}}{\partial t} = -\frac{u_{t,1}^2}{1+h_0} - \frac{\partial h_0}{\partial t}\frac{u_{t,1}}{1+h_0} + \frac{1}{Fr} - \frac{\partial h_0}{\partial t}\left.\frac{\partial u_{\theta,1}}{\partial r}\right|_{r=1+h_0}\,, \tag{6.71}$$

The alternative to solving Eq. (6.64) is thus to solve Eq. (6.71) and use some relation between u_t and $\overline{u_\theta}$ to obtain the latter. This approach has the advantage of not requiring the closure of $\overline{\chi}$ and of eliminating the pressure, as no pressure term appears in the overall set of equations.

6.4.1 Assumed velocity profile approach

As mentioned above, to solve the global problem, one still needs a relationship between $u_{t,1}$ and $\overline{u_{\theta,1}}$, i.e., an expression for $\alpha_1 = \alpha/\theta$ (see Eq. (6.26)). Additionally, the term

$$\left.\frac{\partial u_{\theta,1}}{\partial r}\right|_{r=1+h_0} \tag{6.72}$$

in Eq. (6.71) remains so far unknown. One way to address these two issues is to assume the shape of the u_θ velocity profile, and thus of $u_{\theta,1}$, in the radial direction. For the method to be useful, the dependence of $u_{\theta,1}$ on r, using an adequate parametrization, should be the same for different times and for different values of the global parameters (Fr, Re_D). An a priori determination of this r-dependence is not trivial, so that, to make further progress, it is useful to consider actual velocity profiles from simulations.

Figure 6.3 represents the profile of $u_{\theta,1} = u_\theta/\theta$ as a function of the radial position r at two different angles, $\theta = 9.5°$ and $\theta = 18.4°$, and different positions of the cylinder, i.e., different times, for both

[2] Note that this equation holds for both polar and axisymmetric spherical coordinates.

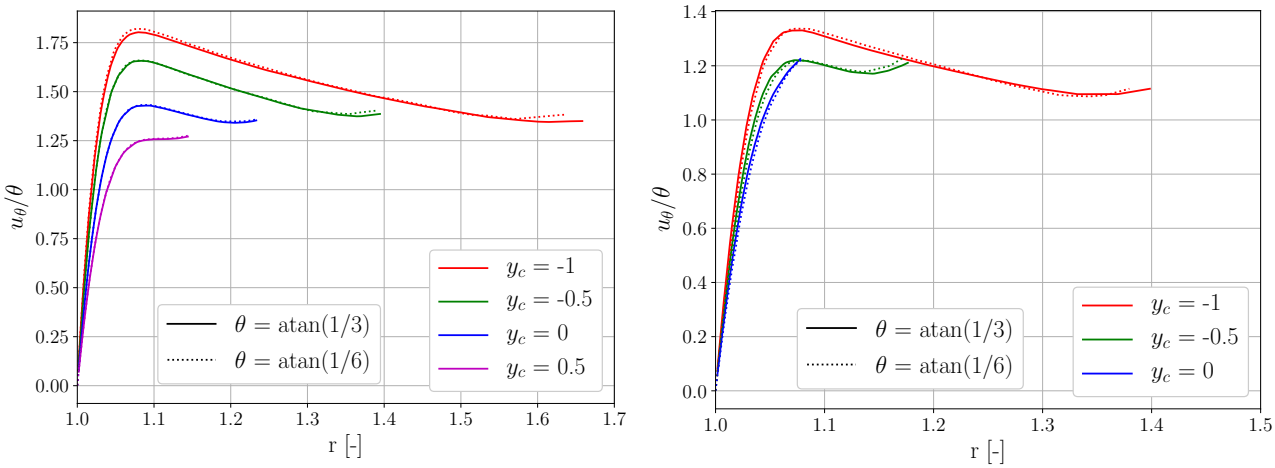


Figure 6.3: Velocity profiles in the thin film above the 2D cylinder (**a**) and the 3D axisymmetric sphere (**b**), at two different angles θ from the apex and different positions y_c; $Fr = 1$, $Re_D = 1000$.

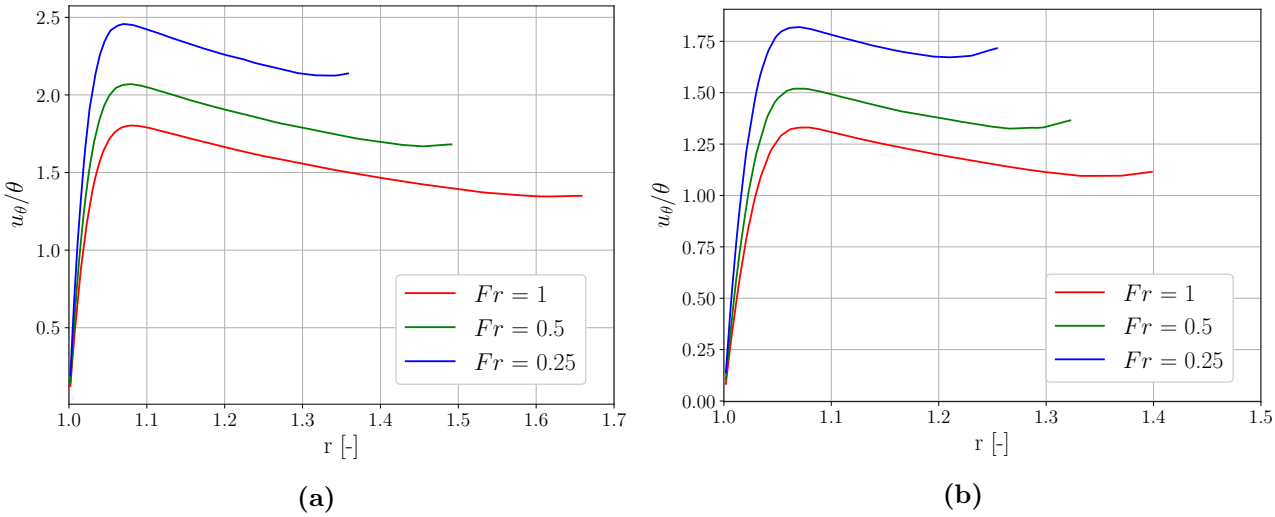
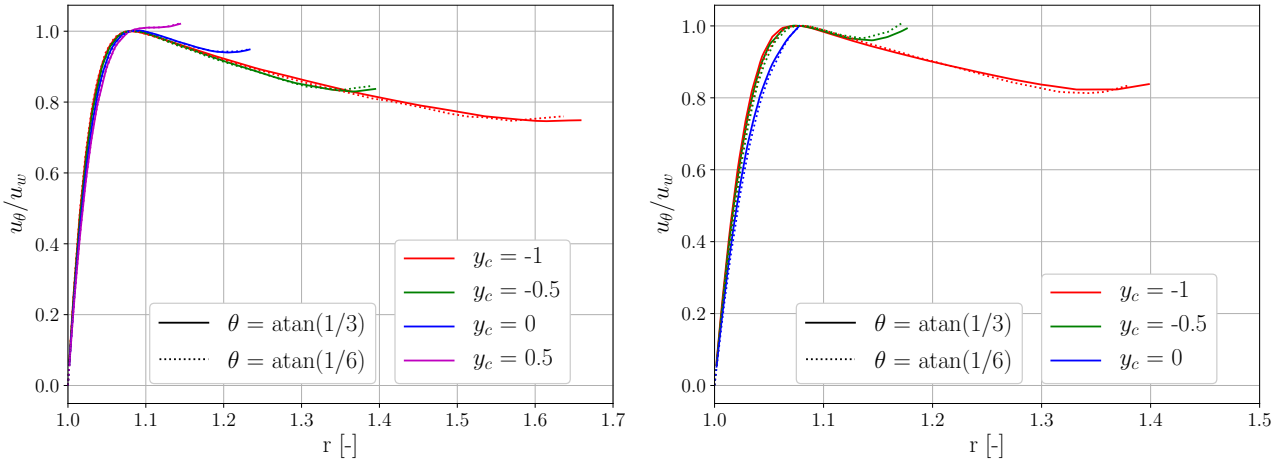


Figure 6.4: Velocity profiles in the thin film above the 2D cylinder (**a**) and the 3D axisymmetric sphere (**b**) for three different Froude numbers; $y_c = -1$, $Re_D = 1000$.

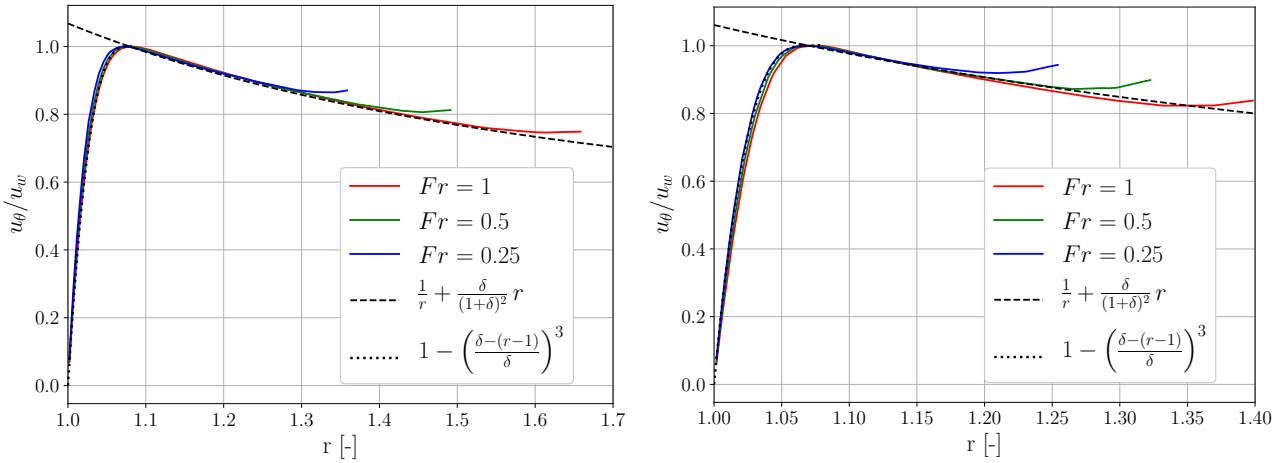
the cylinder and the axisymmetric sphere with $Fr = 1$ and $Re_D = 1000$. Similar velocity profiles are also obtained for different Froude numbers, as shown in Fig. 6.4. First, one observes that almost the same velocity profile of $u_{\theta,1}$ is obtained for different angles θ, which provides support to the main model assumption of a uniform film thickness. One can also identify three regions along the radial direction from these velocity profiles. The region directly adjacent to the cylinder/sphere corresponds to the viscous boundary layer. The velocity vanishes at the solid surface ($r = 1$), because of the no-slip boundary condition expressed in the frame of reference moving with the cylinder/sphere, and increases up to a maximum value, $u_w(t)$, at the boundary layer edge ($r = 1+\delta$, where δ is the boundary layer thickness). At a given Reynolds number, it seems that δ remains fairly constant in time and is independent of the Froude number during the interface crossing. Note also that, in the inviscid limit, the maximum velocity would be found at the cylinder/sphere surface, i.e., there would be no viscous boundary layer. Therefore, the maximum velocity $u_w(t)$ can also be seen as the wall velocity in an inviscid flow. Past the boundary layer edge, the velocity decreases with increasing r until shortly before reaching the free surface. This corresponds to the bulk flow. Finally, in a region of thickness of the order of δ adjacent to the free surface, the velocity stops decreasing with r and even slightly increases when reaching the free surface. This is referred to the viscous free-surface layer and is discussed in more details below. As y_c (i.e., time) increases, both the radial position of the free surface, i.e., h, and the maximum velocity $u_w(t)$ decrease. On the other hand, the velocity at the free surface seems to slightly

increase (more for the sphere) before decreasing when the cylinder/sphere crosses the initial free surface ($y_c \approx 0$). It is also interesting to mention that, during the interface crossing, the thickness of the viscous boundary and viscous free-surface layers remains more or less constant, while the bulk layer, on the other hand, becomes thinner as h decreases. It is expected that, once $h \sim \delta$, only the boundary layer region remains.



(a) **(b)**

Figure 6.5: Velocity profiles in the thin film above the 2D cylinder (**a**) and the 3D axisymmetric sphere (**b**) normalized by their respective maximum velocity u_w, for two different angles θ from the apex and different positions y_c; $Fr = 1$, $Re_D = 1000$.



(a) **(b)**

Figure 6.6: Velocity profiles in the thin film above the 2D cylinder (**a**) and the 3D axisymmetric sphere (**b**) normalized by their respective maximum velocity u_w for three different Froude numbers; $y_c = -1$, $Re_D = 1000$. The black dotted line and the black dashed line represent the assumed velocity profile inside and above the boundary layer, respectively. Those are respectively given by Eqs. (6.80) and (6.77). Cylinder: $\delta = 0.079$, Sphere: $\delta = 0.07$

Interestingly, the different velocity profiles seem to all have a similar shape in r, which suggests that an assumed velocity profile approach might be successful. This is better illustrated in Figs. 6.5 and 6.6, where the velocity u_θ is normalized by the corresponding maximum velocity u_w. One observes in this case that the velocity profiles $u_\theta/u_w(t)$ at different times and Froude numbers collapse on a single curve, at least for radial positions below the viscous free-surface layer. The goal is thus to find a parametric representation of this radial profile, considering each region separately, as depicted in Fig. 6.7. The same velocity profile is assumed for the cylinder and the sphere.

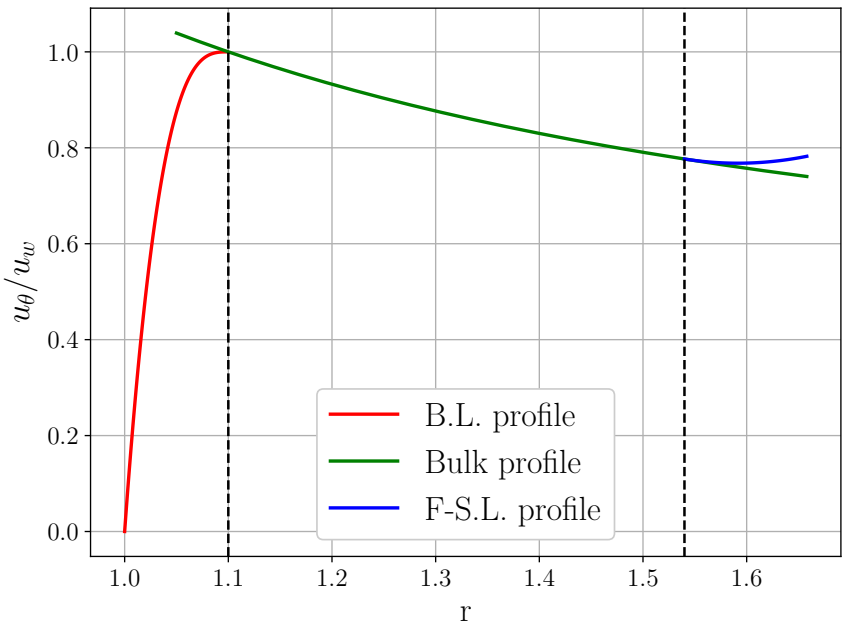


Figure 6.7: Illustration of the three regions defining the overall velocity profile. The first region is the viscous boundary layer, whose velocity profile (red) is defined by an half-cubic branch given by Eq. (6.80); the second region is the bulk layer, with a velocity profile (green) defined by Eq. (6.77); the last region corresponds to the viscous free-surface layer that is neglected in the present model, but whose exact velocity profile (blue) is given in Appendix H. For this example, $\delta = 0.1$, $h = 0.66$ and $\delta_{FS} = 0.12$ (see Appendix H).

6.4.1.1 Viscous free-surface layer

The shear stress tangential to the free surface should vanish. Under the uniform film thickness assumption, this writes

$$\sigma_{r\theta}|_{r=1+h_0} = \frac{2}{Re_D}\left[r\frac{\partial}{\partial r}\left(\frac{u_\theta}{r}\right) + \frac{1}{r}\frac{\partial u_r}{\partial \theta}\right]_{r=1+h_0} \simeq 0\,. \tag{6.73}$$

Moreover, because u_r is symmetric in θ, $\partial_\theta u_r \approx u_{r,2}$, which is assumed small. In other words, at the free surface one has approximately,

$$\frac{\partial}{\partial r}\left(\frac{u_{\theta,1}}{r}\right)\bigg|_{r=1+h_0} \simeq 0\,, \tag{6.74}$$

or equivalently,

$$\frac{\partial u_{\theta,1}}{\partial r}\bigg|_{r=1+h_0} \simeq \frac{u_{\theta,1}}{r}\bigg|_{r=1+h_0} = \frac{u_{t,1}}{1+h_0}\,, \tag{6.75}$$

which could be used to close Eq. (6.71). This results also indicates that $\partial_r u_{\theta,1}$ is positive at the free surface and increases with decreasing h_0, as it can be seen in Fig. 6.3 for instance. The change in the r-dependence of the velocity profile in the viscous free-surface layer thus stems from viscous effects: the velocity must adapt to ensure that the shear stress at the free surface vanishes. This effect is however limited to a thin region of thickness δ_{FS} similar to the boundary layer thickness δ. Moreover, in the inviscid limit, this viscous region would not exist and the velocity profile at the free surface would correspond to that in the bulk. Consequently, and to keep the model simple, the viscous free-surface layer is neglected here and it is assumed that the bulk region extends up to the free surface. Accordingly, and to be consistent with this simplification, the derivative $\partial_r u_{\theta,1}|_{r=1+h_0}$ in Eq. (6.71) is calculated from the assumed velocity profile rather than from Eq. (6.75).

6.4.1.2 Bulk layer

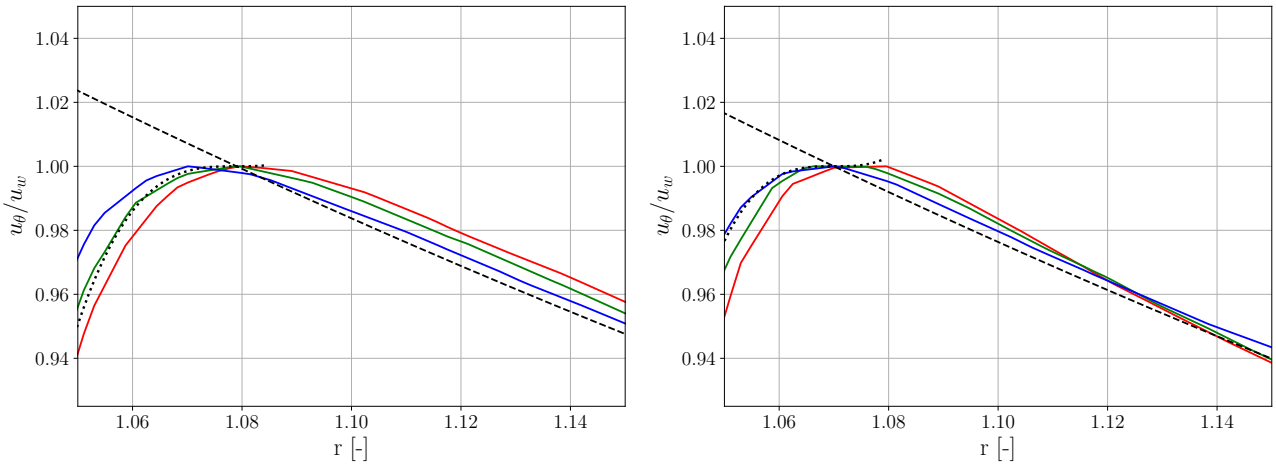
In the bulk layer, trial and error has shown that a good fit of simulation results can be obtained with

$$\frac{u_{\theta,1}(r,t)}{u_{w,1}(t)} = \frac{c_1}{r} + c_2 r\,, \tag{6.76}$$

where $u_{w,1}(t) = u_w(\theta, t)/\theta$, and c_1 and c_2 should be independent of time t, angle θ and Froude number Fr. In particular, taking $c_1 = 1$ and using $u_{\theta,1}(r = 1 + \delta, t) = u_{w,1}(t)$, one obtains $c_2 = \delta/(1 + \delta^2)$ and

$$u_{\theta,1}(r, t) = u_{w,1}(t) \left(\frac{1}{r} + \frac{\delta}{(1+\delta)^2} r \right) \qquad \text{for } 1 + \delta \leq r \leq 1 + h_0(t)\,. \tag{6.77}$$

The fit is illustrated by the dashed black line in Figs. 6.6 and 6.8, in which $\delta = 0.079$ and 0.07 for the cylinder and the sphere, respectively.



(a) **(b)**

Figure 6.8: Closer views of Figs. 6.6a and 6.6b to highlight the junction between the velocity profiles in the boundary layer and in the bulk. Note that, while the velocity is continuous at the junction, its slope is not.

It is interesting to note that, assuming a negligible $\partial_\theta u_r$, the contribution c_1/r corresponds to an irrotational vortex flow, while the contribution $c_2 r$ represents a solid body rotation. Consequently, the second contribution does not lead to any shear stress $\sigma_{r\theta}$. Moreover, the viscous term

$$\frac{2}{Re_D} \left(\frac{\partial^2 u_\theta}{\partial r^2} + \frac{1}{r}\frac{\partial u_\theta}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u_\theta}{\partial \theta^2} - \frac{u_\theta}{r^2} + \frac{2}{r^2}\frac{\partial u_r}{\partial \theta} \right) \tag{6.78}$$

in the θ-momentum equation (i.e., the divergence of the stress) vanishes for both contributions:

$$\frac{\partial^2 u_{\theta,1}}{\partial r^2} + \frac{1}{r}\frac{\partial u_{\theta,1}}{\partial r} - \frac{u_{\theta,1}}{r^2} = 0\,. \tag{6.79}$$

In other words, the assumed velocity profile in the bulk layer does not lead to any viscous contribution in the θ-momentum equation. If one additionally neglects the viscous free-surface layer, this is consistent with the assumption made to obtain Eq. (6.71).

To justify the choice $c_1 = 1$, one can consider the inviscid limit, in which one could expect that only the irrotational c_1/r contribution remains. In this case, the velocity should be maximum at the cylinder/sphere surface, i.e., $r = 1$, such that $u_{\theta,1}(r = 1, t) = u_w(t)$, which yields $c_1 = 1$. This argument also explains why c_1 is independent of δ. On the other hand, c_2 is linked to the rotational contribution of the velocity profile, which should directly be impacted by viscous effects. As such, it is not surprising that c_2 depends on δ.

6.4.1.3 Viscous boundary layer

In the viscous boundary layer, a velocity profile of the form

$$u_{\theta,1}(r, t) = u_{w,1}(t) \left[1 - \left(\frac{\delta - (r-1)}{\delta} \right)^3 \right] \qquad \text{for } 1 \leq r \leq 1 + \delta \tag{6.80}$$

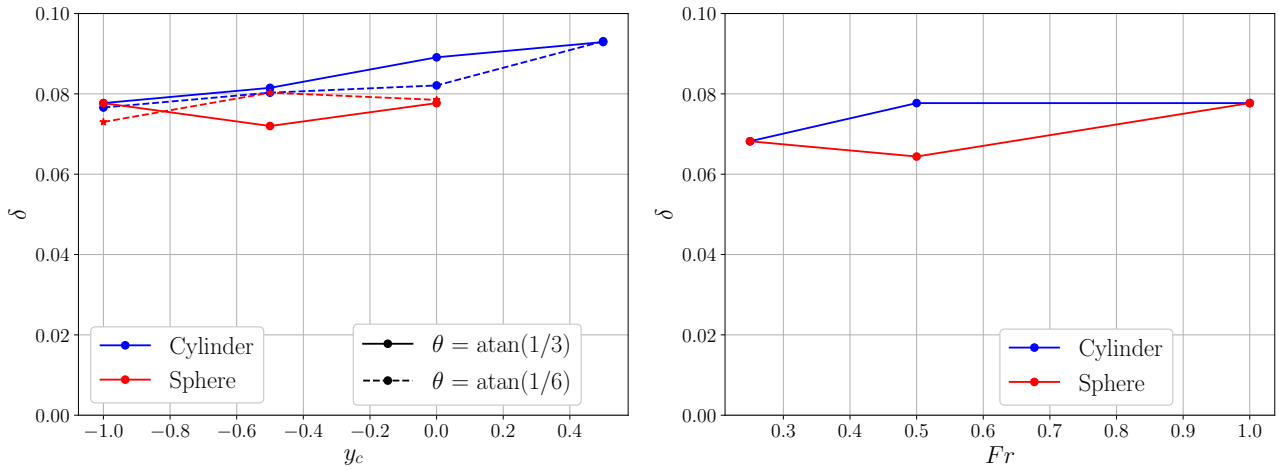
is assumed. The fit is illustrated by the dotted black line in Figs. 6.6 and 6.8, in which $\delta = 0.079$ and 0.07 for the cylinder and the sphere, respectively.

As stated above, the boundary layer thickness δ is approximately independent of the time t (or cylinder position y_c) and Froude number Fr during interface crossing. This is further illustrated by Fig. 6.9, which shows the variations of δ with y_c at two different angles θ and with Fr. Although δ is not perfectly constant, its variations remain small.

On the other hand, the boundary layer thickness should strongly depend on the Reynolds number Re_D. The velocity profiles shown above were all obtained at $Re_D = 1000$. In order to account for this dependency, it is assumed that the boundary layer remains laminar so that its thickness scales like

$$\delta \simeq \frac{k}{\sqrt{Re_D}} . \tag{6.81}$$

Using data for the case $Re_D = 1000$, one obtains the values $k = 2.5$ and $k = 2.21$ for the cylinder and the sphere, respectively.



(a) **(b)**

Figure 6.9: Boundary layer thickness **(a)** as a function of the cylinder/sphere position y_c at two different angles θ, $Fr = 1$ and $Re_D = 1000$, and **(b)** as a function of the Froude number at $y_c = -1$, $\theta = \text{atan}(1/3)$ and $Re_D = 1000$.

6.4.2 Closure using the assumed velocity profile

Using the assumed velocity profiles given by Eq. (6.77) above the boundary layer and Eq. (6.80) in the boundary layer, one can obtain the radial derivative of the azimuthal velocity at the free surface, Eq. (6.72), and the relationship between $u_{t,1}$ and $\overline{u_{\theta,1}}$. In particular, from Eq. (6.77), the tangential velocity at the free surface is given by

$$u_{t,1}(t) = u_{w,1}(t)\left(\frac{1}{1+h_0} + \frac{\delta}{(1+\delta)^2}(1+h_0)\right) , \tag{6.82}$$

which can be inverted to obtain

$$u_{w,1}(t) = \frac{u_{t,1}(t)(1+h_0)}{1+\delta\left(\frac{1+h_0}{1+\delta}\right)^2} . \tag{6.83}$$

On the other hand, the average velocity $\overline{u_{\theta,1}}$ is defined as

$$\overline{u_{\theta,1}} = \frac{u_{w,1}(t)}{h_0}\left[\int_1^{1+\delta}\left(1-\left(\frac{\delta-(r-1)}{\delta}\right)^3\right)\mathrm{d}r + \int_{1+\delta}^{1+h_0}\left(\frac{1}{r}+\frac{\delta}{(1+\delta)^2}r\right)\mathrm{d}r\right] \tag{6.84}$$

for the cylinder and

$$\overline{u_{\theta,1}} = \frac{2u_{w,1}(t)}{h_0(h_0+2)}\left(\int_1^{1+\delta}\left(1-\left(\frac{\delta-(r-1)}{\delta}\right)^3\right)r\mathrm{d}r + \int_{1+\delta}^{1+h_0}\left(\frac{1}{r}+\frac{\delta}{(1+\delta)^2}r\right)r\mathrm{d}r\right) \tag{6.85}$$

for the sphere, where the blue and red terms correspond to the boundary and to the bulk layers, respectively. After integration,

$$\overline{u_{\theta,1}} = \frac{u_{w,1}}{h_0}\left[\frac{3}{4}\delta + \ln\left(\frac{1+h_0}{1+\delta}\right) + \frac{\delta(h_0-\delta)(2+h_0+\delta)}{2(1+\delta)^2}\right] \quad \text{in 2D}, \tag{6.86}$$

$$\overline{u_{\theta,1}} = \frac{2u_{w,1}}{h_0(h_0+2)}\left[\frac{3}{4}\delta + \frac{9}{20}\delta^2 + h_0 - \delta + \frac{\delta}{3(1+\delta)^2}\left((1+h_0)^3 - (1+\delta)^3\right)\right] \quad \text{in 3D}, \tag{6.87}$$

and replacing $u_{w,1}(t)$ by $u_{t,1}(t)$ using Eq. (6.83), one finally obtains

$$\overline{u_{\theta,1}} = \frac{u_{t,1}(t)(1+h_0)}{h_0\left(1+\delta\left(\frac{1+h_0}{1+\delta}\right)^2\right)}\left[\frac{3}{4}\delta + \ln\left(\frac{1+h_0}{1+\delta}\right) + \frac{\delta(h_0-\delta)(2+h_0+\delta)}{2(1+\delta)^2}\right], \tag{6.88}$$

$$\overline{u_{\theta,1}} = \frac{2u_{t,1}(t)(1+h_0)}{h_0(h_0+2)\left(1+\delta\left(\frac{1+h_0}{1+\delta}\right)^2\right)}\left[-\frac{1}{4}\delta + \frac{9}{20}\delta^2 + h_0 + \frac{\delta}{3(1+\delta)^2}\left((1+h_0)^3 - (1+\delta)^3\right)\right], \tag{6.89}$$

for the cylinder and the sphere, respectively. In these two expressions, δ depends on Re_D according to Eq. (6.81). In the inviscid limit ($Re_D \to \infty$), the boundary layer thickness δ vanishes and Eqs. (6.88) and (6.89) become

$$\overline{u_{\theta,1}}(t) = \frac{u_{t,1}(1+h_0)\ln(1+h_0)}{h_0}, \tag{6.90}$$

$$\overline{u_{\theta,1}}(t) = \frac{2u_{t,1}(1+h_0)}{h_0+2}. \tag{6.91}$$

Finally, the velocity gradient at the free surface to close Eq. (6.71) can be determined using Eq. (6.77):

$$\left.\frac{\partial u_{\theta,1}}{\partial r}\right|_{r=1+h_0} = u_{w,1}\left(-\frac{1}{1+h_0^2} + \frac{\delta}{(1+\delta)^2}\right), \tag{6.92}$$

which, after introducing Eq. (6.83), yields

$$\left.\frac{\partial u_{\theta,1}}{\partial r}\right|_{r=1+h_0} = \frac{u_{t,1}}{(1+h_0)}\left(\frac{\delta\left(\frac{1+h_0}{1+\delta}\right)^2 - 1}{\delta\left(\frac{1+h_0}{1+\delta}\right)^2 + 1}\right). \tag{6.93}$$

Finally, Eq. (6.71) can be rewritten in the closed form

$$\frac{\partial u_{t,1}}{\partial t} = -\frac{u_{t,1}^2}{1+h_0} - \frac{\partial h_0}{\partial t}\left(\frac{u_{t,1}}{1+h_0}\right)\left(\frac{2\delta\left(\frac{1+h_0}{1+\delta}\right)^2}{1+\delta\left(\frac{1+h_0}{1+\delta}\right)^2}\right) + \frac{1}{Fr}. \tag{6.94}$$

It is interesting to note that, in the inviscid limit ($\delta = 0$), the second term on the right-hand-side vanishes, leading to the following equation:

$$\frac{\partial u_{t,1}}{\partial t} = -\frac{u_{t,1}^2}{1+h_0} + \frac{1}{Fr}. \tag{6.95}$$

This provides another justification for replacing the zero-shear-stress condition at the free-surface by the assumed velocity profile derivative. For sufficiently high Reynolds numbers, Eq. (6.95) should thus be a good approximation to Eq. (6.94).

6.4.3 Summary of the final model equations

A model to describe the evolution of the thin film of fluid above a cylinder or an axisymmetric sphere has been developed. A closure term for the pressure has been avoided by using the θ-momentum equation evaluated at the free surface, where the pressure gradient vanishes. In addition, a velocity profile in the radial direction has been assumed based on actual simulation profiles. Two separate r-dependencies have been assumed in the boundary layer and in the bulk, while the viscous free-surface layer has been neglected. Leveraging these velocity profiles, the relation between the average fluid velocity in the film and the velocity at the free surface can be deduced, which enables to obtain the former indirectly from the dynamical evolution of the latter.

The only required calibration parameter in this model is the parameter k that relates the boundary layer thickness δ to the Reynolds number Re_D, and that is obtained by fitting some of the velocity profiles from PFEM simulations (see Fig.6.6b). Knowing the average velocity in the film, the evolution of the film thickness is therefore known through mass conservation, as derived in Section 6.3.2. Through these steps Eqs. (6.63) to (6.66) can be replaced by the following equations (based on mass conservation, momentum conservation along θ, geometrical ansatz and assumed velocity profile):

2D cylinder

$$\frac{\partial f_0}{\partial t} = \overline{u_\theta}_1(t) \tag{6.96}$$

3D axisymmetric sphere

$$\frac{\partial f_0}{\partial t} = 2\overline{u_\theta}_1(t) \tag{6.96}$$

$$\frac{\partial u_{t,1}}{\partial t} = -\frac{u_{t,1}^2}{1+h_0} - \frac{\partial h_0}{\partial t}\left(\frac{u_{t,1}}{1+h_0}\right)\left(\frac{2\delta\left(\frac{1+h_0}{1+\delta}\right)^2}{1+\delta\left(\frac{1+h_0}{1+\delta}\right)^2}\right) + \frac{1}{Fr} \tag{6.97}$$

$$\mathrm{e}^{-f_0} = h_0\mathrm{e}^{h_0} \tag{6.98}$$

$$\mathrm{e}^{-f_0} = \frac{h_0}{h_0+2}\mathrm{e}^{2h_0} \tag{6.98}$$

$$\frac{\partial h_0}{\partial t} = -\left(\frac{h_0}{h_0+1}\right)\frac{\partial f_0}{\partial t} \tag{6.99}$$

$$\frac{\partial h_0}{\partial t} = -\left(\frac{h_0(h_0+2)}{2(h_0+1)^2}\right)\frac{\partial f_0}{\partial t} \tag{6.99}$$

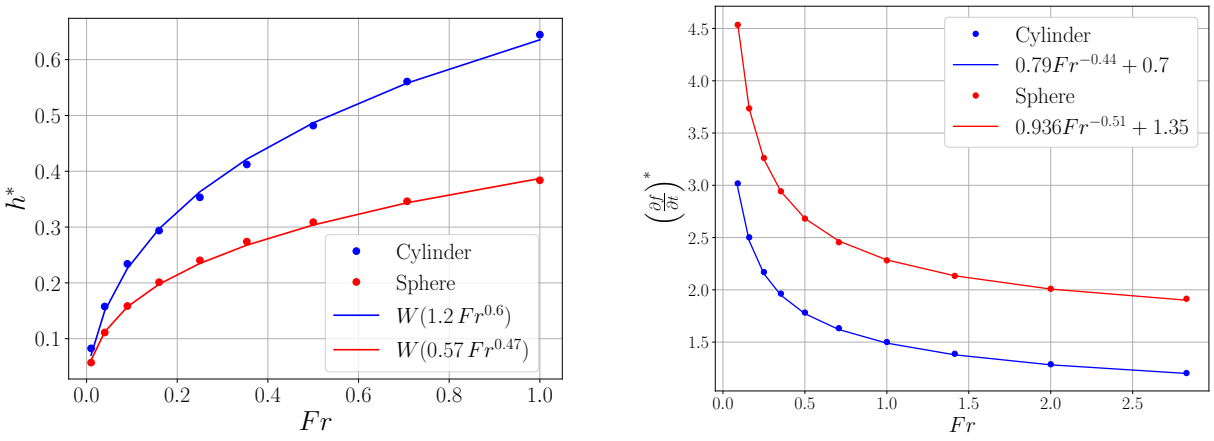
$$\overline{u_\theta}_{,1}(u_{t,1}) \quad \text{from Eq. (6.88)} \tag{6.100}$$

$$\overline{u_\theta}_{,1}(u_{t,1}) \quad \text{from Eq. (6.89)} \tag{6.100}$$

6.4.4 Initial conditions

To integrate the above two ordinary differential equations, Eqs. (6.96) and (6.97), two initial conditions for h_0 and $\partial_t h_0$, or equivalently for f_0 and $\partial_t f_0$, are required. A natural moment at which these initial conditions should be imposed is the time when the top cylinder reaches the initial free surface, i.e., at $y_c = -1$. This initial time is also convenient because an empirical relationship for $h_0^*(Fr)$ is already known, as illustrated again in Fig. 6.10a. Similarly, it would be useful to have another empirical expression that relates $(\partial_t f_0)^*$ to the Froude number. It turns out that a power law provides a very good fit of the simulation results, as shown in Fig. 6.10b.

The two expressions for $h_0^*(Fr)$ and $(\partial_t f_0)^*(Fr)$ could therefore be used as initial conditions for the model given by Eqs. (6.96) to (6.100). However, it is important to observe that these expressions do not include a dependence on the Reynolds number. As h_0 is expected to depend, albeit weakly, on Re_D, the model is hereafter evaluated using as initial conditions the exact values for h_0^* and $(\partial_t f_0)^*$ directly taken from the simulations, rather than the corresponding approximations from empirical correlations. This ensures a slightly better model accuracy at other Reynolds numbers.



(a) **(b)**

Figure 6.10: **(a)** Film thickness and **(b)** average azimuthal velocity at the interface crossing ($y_c = -1$) as a function of the Froude number that can be used as initial conditions for the model ($Re_D = 1000$). The symbols are simulation data while the continuous lines represent empirical fits.

6.4.5 Numerical resolution scheme

To obtain the solution $h_0(t)$, the two coupled ordinary differential equations, Eqs. (6.96) and (6.97), are integrated numerically from the initial conditions using a predictor/corrector scheme and a uniform time step size Δt.

First, $(u_t)^{n+1}$ is updated using a first order explicit scheme,

$$(u_{t,1})^{n+1} = (u_{t,1})^n + \left(\frac{\partial u_{t,1}}{\partial t}\right)^n \Delta t\,, \tag{6.101}$$

where $(\partial_t u_{t,1})^n$ is given by Eq. (6.97) evaluated at time $t_n = n\Delta t$. Note that, when needed, $(h_0)^n$ and $(\partial_t h_0)^n$ can be obtained from $(f_0)^n$ and $(\partial f_0)^n$ using Eq. (6.98) and Eq. (6.99), respectively. Following the same approach, an estimate of f_0 at time $n+1$ (predictor) is calculated as

$$(\tilde{f}_0)^{n+1} = (f_0)^n + \left(\frac{\partial f_0}{\partial t}\right)^n \Delta t\,, \tag{6.102}$$

where $(\partial_t f_0)^n$ is obtained from the solution at the current time step n through Eqs. (6.96) and (6.100). This predictor value, $(\tilde{f}_0)^{n+1}$, and $(u_t)^{n+1}$ can then be used to calculate an improved estimate $(\partial_t \tilde{f}_0)^{n+1}$ (corrector). Finally, the value of $(f_0)^{n+1}$ is computed using an average of the predictor and corrector:

$$(f_0)^{n+1} = (f_0)^n + \frac{1}{2}\left(\left(\frac{\partial f_0}{\partial t}\right)^n + \left(\frac{\partial \tilde{f}_0}{\partial t}\right)^{n+1}\right)\Delta t\,. \tag{6.103}$$

At this stage, the process starts over at Eq. (6.101) for the next time step.

In the following, the system of equations has been integrated numerically using a time step size $\Delta t = 0.005$. The total simulation time, on the other hand, has been chosen based on the fact that the velocity profile is formally not valid for $h_0 < \delta$. In practice, the integration has been stopped at a value $h_0 = 0.5\delta$ as the fit remained accurate up to this value. In the viscous thin film limit, i.e., for smaller h_0, another model with a different assumed velocity profile should be used. This is briefly considered in Section 6.5.5.

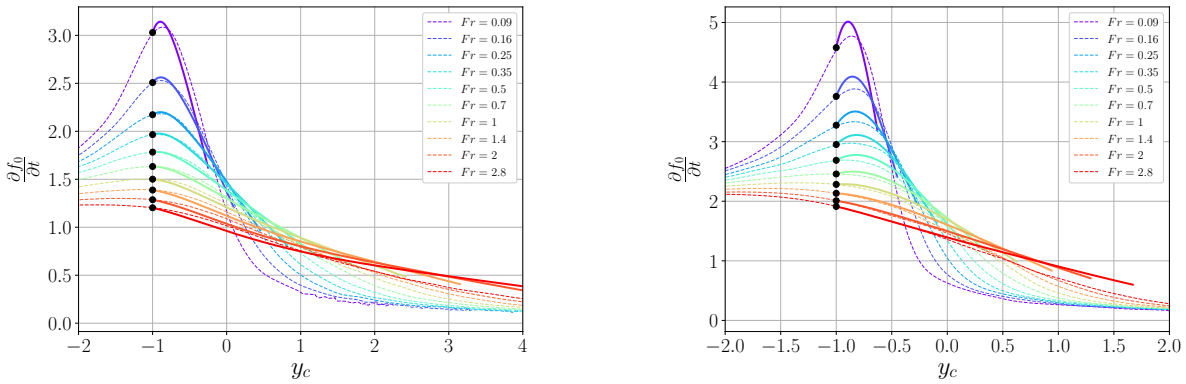
6.5 Results

The model accuracy is now assessed by considering different cases. First, viscous calculations are performed at $Re_D = 1000$, i.e., the Reynolds number at which the parameter k for the boundary layer

thickness has been calibrated. This is then extended to other Reynolds numbers. The inviscid limit is also briefly considered. In a subsequent section, the model is leveraged to obtain the pressure at the cylinder/sphere apex. Finally, the viscous thin film limit (i.e., when $h \ll 1$), for which a different assumed velocity profile is proposed, is investigated.

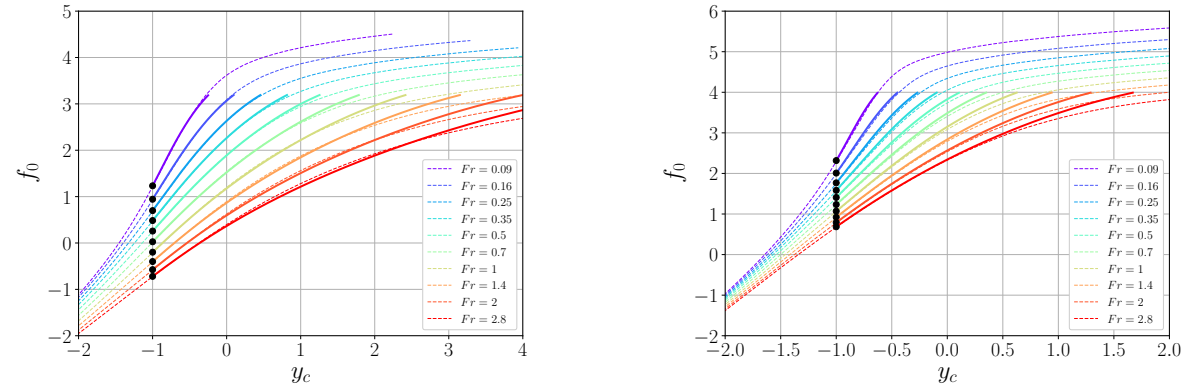
6.5.1 Model validation at $Re_D = 1000$

Using the initial conditions given in Fig. 6.10, the model is integrated, for the cylinder and the sphere, at $Re_D = 1000$ and several Froude numbers, ranging from $Fr = 0.09$ to $Fr = 2.8$ (corresponding to case F1 in Table 5.4). The results of the model (continuous lines) are shown in Figs. 6.11 to 6.13 for $\partial_t f_0$, f_0 and h_0 and are compared to the numerical results from PFEM simulations (dashed lines). The agreement between model and PFEM simulations is very encouraging, especially considering all the assumptions that have been made in the model development. As expected, small discrepancies are nevertheless visible as y_c increases. Among other things, this might be due to the assumed constant boundary layer thickness, while δ can be expected to depend weakly on Froude number and time. Neglecting the viscous free-surface layer might also contribute to the discrepancy.



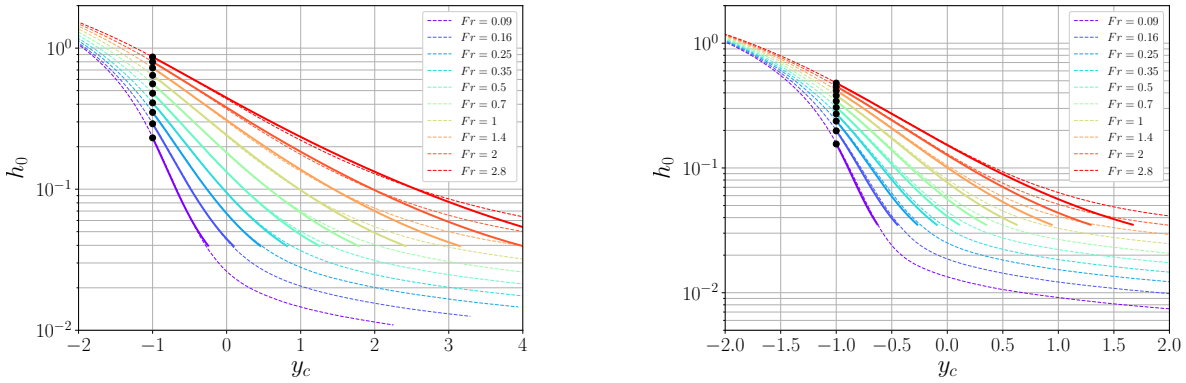
(a) **(b)**

Figure 6.11: Average azimuthal velocity $\partial_t f_0 = \overline{u_\theta}/\theta$ for the cylinder **(a)** and the sphere (twice the velocity in this case) **(b)** at different Froude numbers and $Re_D = 1000$; PFEM simulations (dashed lines) and model (continuous lines). The black dots represent the initial conditions for the model.



(a) **(b)**

Figure 6.12: Argument of the exponential representing the film thickness in the case of the cylinder **(a)** and the sphere **(b)** at different Froude numbers and $Re_D = 1000$; PFEM simulations (dashed lines) and model (continuous lines). The black dots represent the initial conditions for the model.



(a) **(b)**

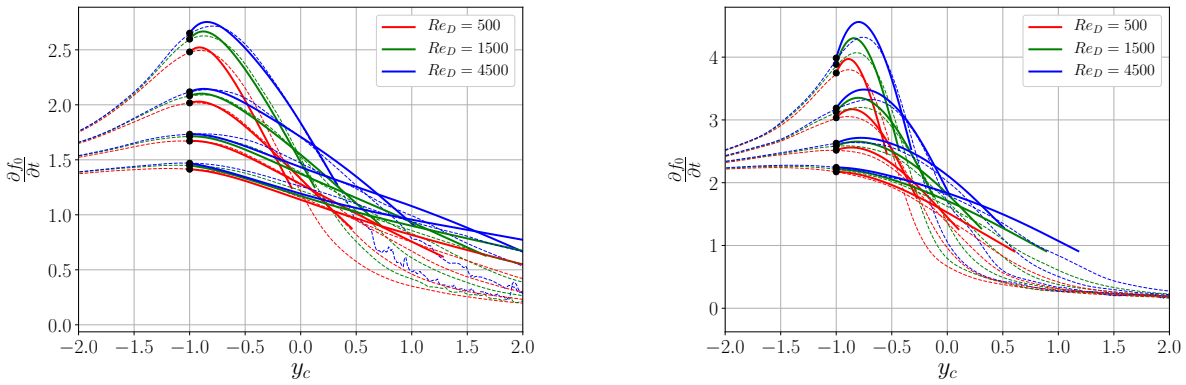
Figure 6.13: Film thickness at the apex of the cylinder **(a)** and the sphere **(b)** at different Froude numbers and $Re_D = 1000$; PFEM simulations (dashed lines) and model (continuous lines). The black dots represent the initial conditions for the model.

6.5.2 Model validation at other Reynolds numbers

To assess the validity of the model for the boundary layer thickness, Eq. (6.81), three other Reynolds numbers are now considered. The parameters of the corresponding simulations are listed in Table 6.1. The comparison between model and PFEM simulations is shown for $\partial_t f_0$, f_0 and h_0 in Figs. 6.14 to 6.16, for the 2D cylinder and the 3D axisymmetric sphere. Here again, the predictions of the model for the range of Froude and Reynolds numbers considered are quite good, providing further support to the model.

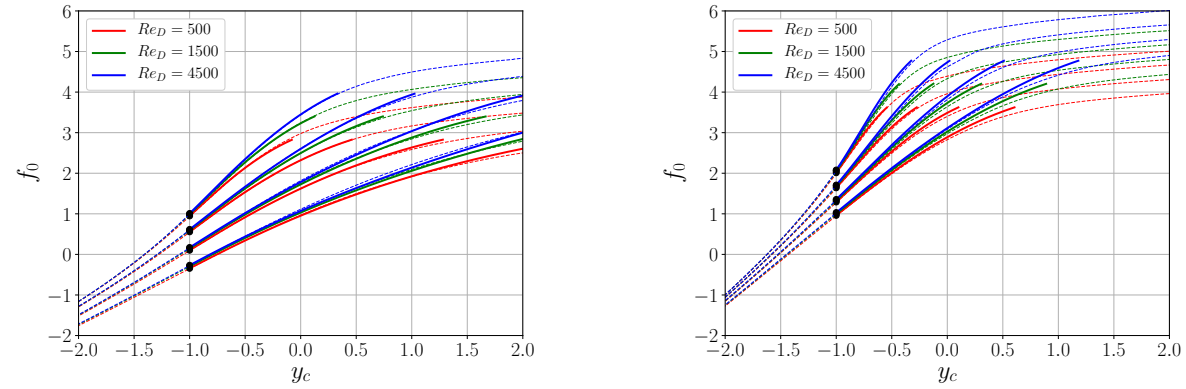
		F1
Cylinder diameter	D	2
Cylinder velocity	U	1
Froude number	Fr	0.15 - 0.3 - 0.6 - 1.2
Reynolds number	Re_D	500 - 1500 - 4500
Tank half width	W	21
Tank height	H	28
Initial depth of the cylinder	d	7
Minimal mesh size	L^*_{min}	0.003
Maximal mesh size	L^*_{max}	1.2
Time step size	Δt	0.0033

Table 6.1: Simulation parameters for the cylinder/sphere rising toward a free surface at constant velocity, used for the validation of the film model. All quantities are non-dimensionalized using the cylinder radius a and velocity U, except for the Reynolds number which uses the cylinder diameter D instead of a, to be consistent with the literature.



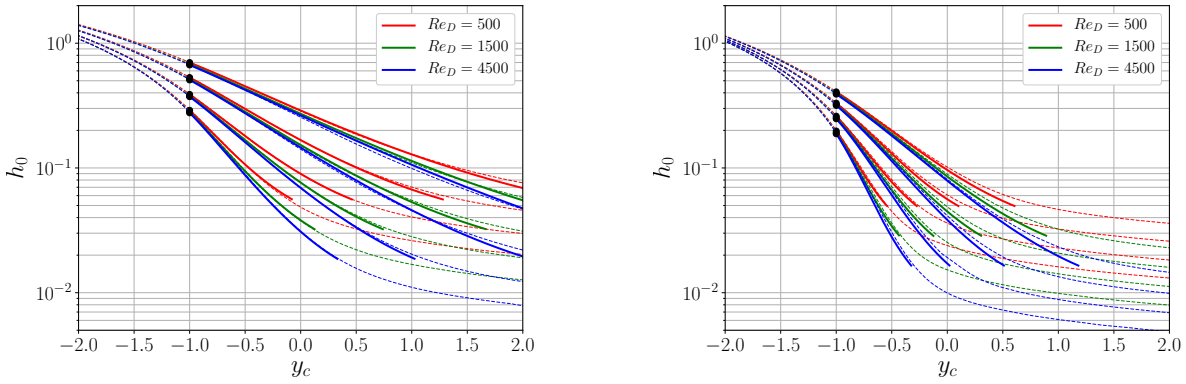
(a) **(b)**

Figure 6.14: Average azimuthal velocity $\partial_t f_0 = \overline{u_\theta}/\theta$ for the cylinder **(a)** and the sphere (twice the velocity in this case) **(b)** at different Froude numbers (Fr $\in$ [0.15-0.3-0.6-1.2], top to bottom curve triplets) and Reynolds numbers of 500 (red), 1500 (green) and 4500 (blue); PFEM simulations (dashed lines) and model (continuous lines). The black dots represent the initial conditions for the model.



(a) **(b)**

Figure 6.15: Argument of the exponential representing the film thickness in the case of the cylinder **(a)** and the sphere **(b)** at different Froude numbers (Fr $\in$ [0.15-0.3-0.6-1.2], top to bottom curve triplets) and Reynolds numbers of 500 (red), 1500 (green) and 4500 (blue); PFEM simulations (dashed lines) and model (continuous lines). The black dots represent the initial conditions for the model.

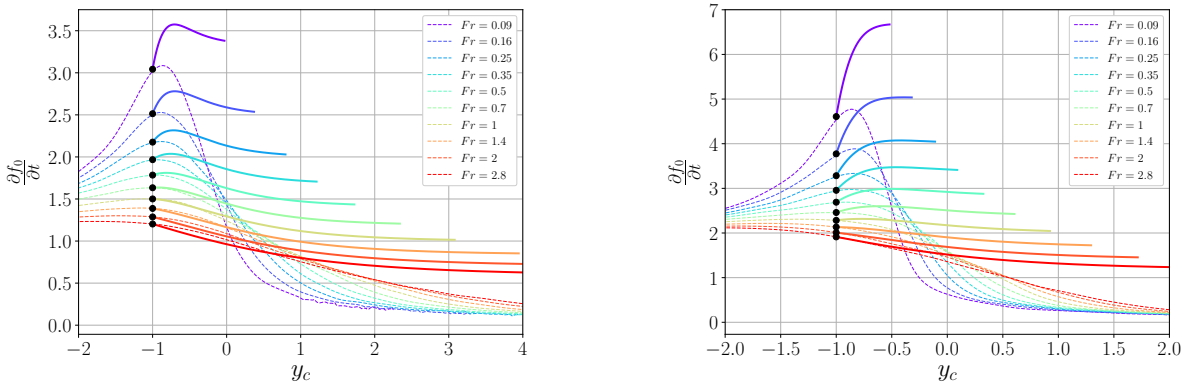


(a) **(b)**

Figure 6.16: Film thickness at the apex of the cylinder **(a)** and the sphere **(b)** at different Froude numbers (Fr $\in$ [0.15-0.3-0.6-1.2], top to bottom curve triplets) and Reynolds numbers of 500 (red), 1500 (green) and 4500 (blue); PFEM simulations (dashed lines) and model (continuous lines). The black dots represent the initial conditions for the model.

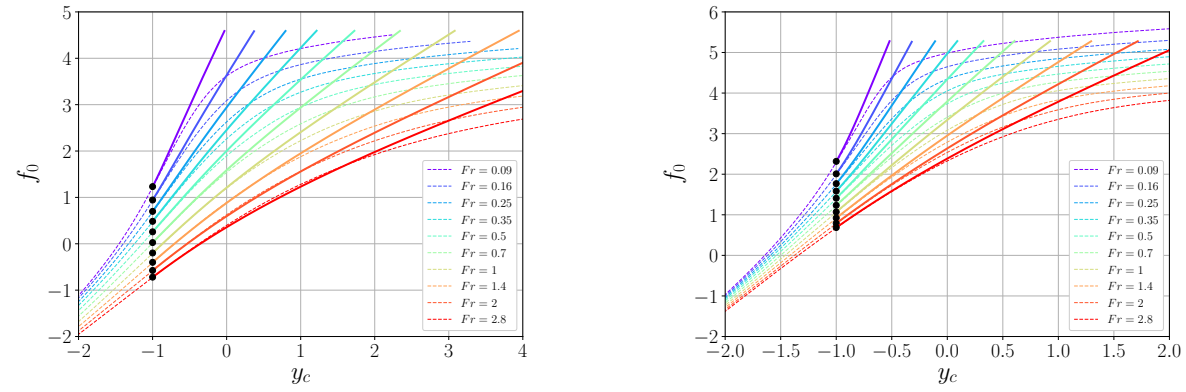
6.5.3 Inviscid limit

Although the PFEM solver cannot deal with free-slip boundary conditions on curved surfaces, it is interesting to analyze the model predictions in the inviscid limit, i.e., when $\delta = 0$. The assumed velocity profile then reduces over the entire film height to the bulk layer profile, Eq. (6.77), with only the irrotational contribution. Results are again shown for $\partial_t f_0$, f_0 and h_0 in Figs. 6.17, 6.18 and 6.19. As expected, large differences are observed between the model assuming an inviscid flow and the viscous PFEM calculations. In particular, owing to the absence of friction at the cylinder/sphere surface, the mean azimuthal velocity, $\overline{u_\theta} = \partial_t f_0$ becomes much larger than in the viscous case. This translates then into a faster decrease of the film thickness h_0. One can also observe that the curvature change in the h_0 curves, visible in Fig. 6.13, is not present in the inviscid case of Fig. 6.19. One has thus not only a quantitative but also a qualitative change of the solution.



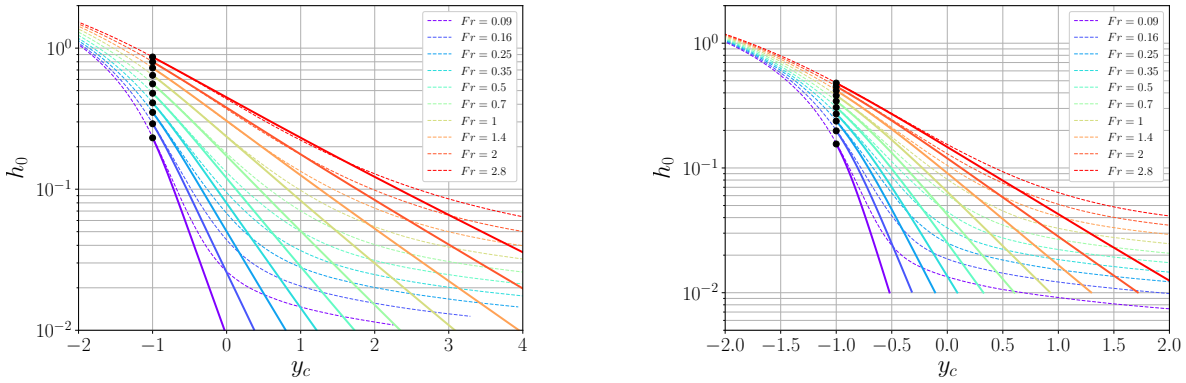
(a) **(b)**

Figure 6.17: Average azimuthal velocity $\partial_t f_0 = \overline{u_\theta}/\theta$ for the cylinder **(a)** and the sphere (twice the velocity in this case) **(b)** at different Froude numbers; PFEM simulations at $Re_D = 1000$ (dashed lines) and model assuming an inviscid flow (continuous lines). The black dots represent the initial conditions for the model.



(a) **(b)**

Figure 6.18: Argument of the exponential representing the film thickness in the case of the cylinder **(a)** and the sphere **(b)** at different Froude numbers; PFEM simulations at $Re_D = 1000$ (dashed lines) and model assuming an inviscid flow (continuous lines). The black dots represent the initial conditions for the model.



(a) **(b)**

Figure 6.19: Film thickness at the apex of the cylinder **(a)** and the sphere **(b)** at different Froude numbers; PFEM simulations at $Re_D = 1000$ (dashed lines) and model assuming an inviscid flow (continuous lines). The black dots represent the initial conditions for the model.

6.5.4 Pressure at the apex of the body

The purpose of the film model is to predict the film thickness h during the interface crossing of the cylinder/sphere. Nonetheless, the model can also be leveraged to predict the pressure or, more precisely, its zeroth-order approximation in θ, $p_{w,0}$, at the cylinder/sphere apex. This is demonstrated here for the case of the cylinder.

To achieve this, the radial momentum equation is integrated from $r = 1$ to $r = 1 + h \approx 1 + h_0$ while neglecting the viscous effects in the radial direction:

$$\begin{aligned} -[p]_1^{1+h_0} = p_{w,0} + \mathcal{O}(\theta^2) &= \int_1^{1+h_0} \frac{\partial u_{r,0}}{\partial t} \mathrm{d}r + \int_1^{1+h_0} u_{r,0} \frac{\partial u_{r,0}}{\partial r} \mathrm{d}r + \int_1^{1+h_0} \frac{\mathrm{d}r}{Fr} + \mathcal{O}(\theta^2) \\ &= \int_1^{1+h_0} \frac{\partial u_{r,0}}{\partial t} \mathrm{d}r + \left[\frac{u_{r,0}^2}{2} \right]_1^{1+h_0} + \frac{h_0}{Fr} + \mathcal{O}(\theta^2) \\ &= \int_1^{1+h_0} \frac{\partial u_{r,0}}{\partial t} \mathrm{d}r + \frac{1}{2} \left(\frac{\partial h_0}{\partial t} \right)^2 + \frac{h_0}{Fr} + \mathcal{O}(\theta^2) \,, \end{aligned} \tag{6.104}$$

which yields, at zeroth order in θ,

$$p_{w,0} = \int_1^{1+h_0} \frac{\partial u_{r,0}}{\partial t} \mathrm{d}r + \frac{1}{2} \left(\frac{\partial h_0}{\partial t} \right)^2 + \frac{h_0}{Fr} \,. \tag{6.105}$$

The radial velocity, i.e., $u_{r,0}$, can be obtained from mass conservation (for the cylinder):

$$u_{r,0} = \frac{1}{r} \left(- \int u_{\theta,1} \mathrm{d}r + c_1 \right) \tag{6.106}$$

where c_1 is an integration constant to satisfy the impermeability condition

$$u_{r,0}|_{r=1} = 0 \,. \tag{6.107}$$

Replacing $u_{\theta,1}$ by Eqs. (6.77) and (6.80) in Eq. (6.106) then yields

$$\frac{u_{r,0}(r,t)}{u_{w,1}(t)} = \begin{cases} 1 + \frac{1}{4\delta^3 r}(\delta + 1 - r)^4 + \frac{c_1}{r} & \text{for } 1 \le r \le 1 + \delta \,, \\ \frac{\ln(r)}{r} + \frac{\delta}{2(1+\delta)^2} r + \frac{c_2}{r} & \text{for } 1 + \delta \le r \le 1 + h_0 \,, \end{cases} \tag{6.108}$$

where the integration constant c_2 can be determined from the continuity of $u_{r,0}$ in $r = 1 + \delta$. Because the impermeability condition and the continuity condition must be satisfied at all times t, we have

$$c_1 = -1 - \frac{\delta}{4} \,, \tag{6.109}$$

$$c_2 = \frac{\delta}{4} - \ln(1 + \delta) \,. \tag{6.110}$$

As the time dependence of $u_{r,0}(r,t)$ is solely contained in $u_{w,1}(t)$, its time derivative $\partial_t u_{r,0}$ is given by $\partial_t u_{w,1}$ multiplied by the right-hand-side of Eq. (6.108). We thus have for the first term on the right-hand-side of Eq. (6.105)

$$\int_1^{1+h_0} \frac{\partial u_{r,0}}{\partial t} \mathrm{d}r = \frac{\partial u_{w,1}}{\partial t} \left[\int_1^{1+\delta} \left(1 + \frac{1}{4\delta^3 r}(\delta + 1 - r)^4 - \frac{1+\delta/4}{r} \right) \mathrm{d}r \right. \\ \left. + \int_{1+\delta}^{1+h_0} \left(\frac{\ln(r)}{r} + \frac{\delta}{2(1+\delta)^2} r + \frac{\delta}{4r} - \frac{\ln(1+\delta)}{r} \right) \mathrm{d}r \right]. \tag{6.111}$$

After integration in r, one obtains

$$\int_1^{1+h_0} \frac{\partial u_{r,0}}{\partial t} \mathrm{d}r = \frac{\partial u_{w,1}}{\partial t} \left[\frac{\ln(1+\delta)}{4\delta^3}(6\delta^2 + 4\delta + 1) + \frac{1}{48\delta^2}(23\delta^3 - 52\delta^2 - 42\delta - 12) \right. \\ \left. + \frac{1}{2}\left(\ln \frac{1+h_0}{1+\delta} \right)^2 + \frac{\delta}{4} \ln \frac{1+h_0}{1+\delta} + \frac{\delta}{4}\left(\frac{(1+h_0)^2}{(1+\delta)^2} - 1 \right) \right]. \tag{6.112}$$

Introducing Eq. (6.112) into Eq. (6.104) finally yields

$$p_{w,0} = \frac{\partial u_{w,1}}{\partial t} \left[\frac{\ln(1+\delta)}{4\delta^3}(6\delta^2 + 4\delta + 1) + \frac{1}{48\delta^2}(23\delta^3 - 52\delta^2 - 42\delta - 12) \right. \\ \left. + \frac{1}{2}\left(\ln \frac{1+h_0}{1+\delta} \right)^2 + \frac{\delta}{4} \ln \frac{1+h_0}{1+\delta} + \frac{\delta}{4}\left(\frac{(1+h_0)^2}{(1+\delta)^2} - 1 \right) \right] \\ + \frac{1}{2}\left(\frac{\partial h_0}{\partial t} \right)^2 + \frac{h_0}{Fr}, \tag{6.113}$$

with $\partial_t u_{w,1}$ the time derivative of $u_{w,1}$ given by Eq. (6.83), i.e.,

$$\frac{\partial u_{w,1}}{\partial t} = \frac{1}{1 + \delta\left(\frac{1+h_0}{1+\delta}\right)^2} \left[(1+h_0)\frac{\partial u_{t,1}}{\partial t} + \frac{(1+\delta)^2 - \delta(1+h_0)^2}{(1+\delta)^2 + \delta(1+h_0)^2} u_{t,1} \frac{\partial h_0}{\partial t} \right]. \tag{6.114}$$

In turn, $u_{t,1}$, $\partial_t u_{t,1}$, h_0 and $\partial_t h_0$ are obtained from the numerical solution of the system of Eqs. (6.96)-(6.100).

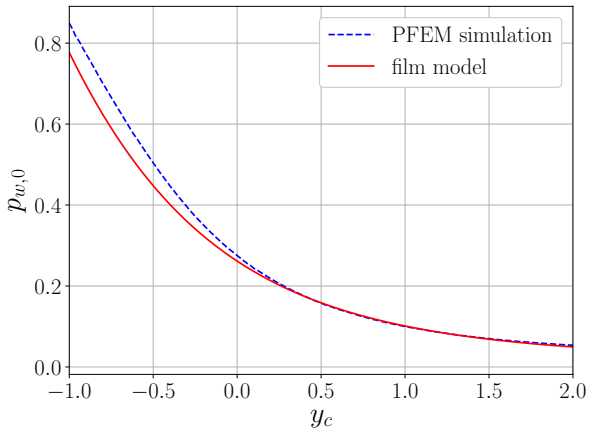


Figure 6.20: Pressure at the cylinder apex as a function of time / position for $Fr = 1$ and $Re_D = 1000$: zeroth-order approximation, Eq. (6.113), of the film model (continuous red line) and pressure from PFEM simulations (dashed blue line).

The film model approximation $p_{w,0}(t)$, Eq. (6.113), for the cylinder at $Fr = 1$ and $Re_D = 1000$ is compared to PFEM simulation results in Fig. 6.20. It can be seen that the model provides a good

qualitative approximation of the pressure at the cylinder apex. Quantitatively, the relative error is less than 10%. This demonstrates that the film model proposed here can not only provide a good approximation for the film thickness, but also for the pressure at the apex of the cylinder. Finally, note that the mathematical developments have been restricted to the cylinder case here, but a similar approach could also be used for the axisymmetric sphere.

6.5.5 Thin film limit

The film model is now extended to the viscous thin film regime, $h \ll 1$. In this limit, curvature effects can be neglected and viscous effects dominate so that the velocity profile corresponds to a semi-parabola. This is illustrated in Fig. 6.21 for the case of the cylinder, where the normalized velocity profile u_θ/u_t is represented as a function of the scaled radius

$$\tilde{r}(t) = 1 + \frac{r-1}{h_0(t)}\,. \tag{6.115}$$

One can observe that, as the film becomes thinner (i.e., y_c, and thus t, increase), the normalized velocity profile tends towards the parabolic profile (dotted black line in Fig. 6.21)

$$\frac{u_\theta(\tilde{r},t)}{u_t(t)} = 1 - (2-\tilde{r})^2 \qquad \text{for } 1 \leq \tilde{r} \leq 2\,. \tag{6.116}$$

The same steps as in Section 6.4.1 can be followed, using the assumed velocity profile for the thin film limit given by Eq. (6.116) this time. First, the average velocity can be related to the velocity at the free surface:

$$\overline{u_{\theta,1}} = \frac{2}{3}u_{t,1}\,. \tag{6.117}$$

Then, one can neglect the two terms multiplied by $\partial_t h_0$ in Eq. (6.71) as they are of first order in h_0 and thus much smaller than the other terms. However, unlike previously, in the thin film limit the viscous terms dominate. This means that the θ-momentum equation evaluated at the free surface, Eq. (6.71), should now include viscous terms $\mathcal{V}_t$:

$$\frac{\partial \overline{u_{\theta,1}}}{\partial t} = -\frac{3}{2}\frac{\overline{u_{\theta,1}}^2}{1+h_0} + \frac{2}{3Fr} + \frac{2}{3}\mathcal{V}_t\,, \tag{6.118}$$

where Eq. (6.117) has been used to replace $u_{t,1}$ by $\overline{u_{\theta,1}}$. For the cylinder, using the same assumptions as in Section 6.4.1.2, $\mathcal{V}_t$ is given at first order by

$$\mathcal{V}_t = \frac{2}{Re_D}\left[\frac{\partial^2 u_{\theta,1}}{\partial r^2} + \frac{1}{r}\frac{\partial u_{\theta,1}}{\partial r} - \frac{u_{\theta,1}}{r^2}\right]_{r=1+h_0}\,. \tag{6.119}$$

Considering $r \simeq 1$ (thin film assumption), $\partial_r = \frac{1}{h_0}\partial_{\tilde{r}}$, and introducing the velocity profile given by Eq. (6.116), one finds

$$\mathcal{V}_t = \frac{-2u_{t,1}}{Re_D}\left(1 + \frac{2}{h_0^2}\right) \simeq \frac{-4u_{t,1}}{Re_D h_0^2}\,, \tag{6.120}$$

where the last approximation is made considering $h_0 \ll 1$. The mathematical development being much more complicated for the case of the axisymmetric sphere, the same form of $\mathcal{V}_t$ is used as an ad-hoc approximation. Finally, the free-surface equation in the thin film limit becomes

$$\frac{\partial \overline{u_{\theta,1}}}{\partial t} = -\frac{3}{2}\overline{u_{\theta,1}}^2 + \frac{2}{3Fr} - \frac{4\overline{u_{\theta,1}}}{Re_D h_0^2}\,. \tag{6.121}$$

Using this equation and the relation between $\overline{u_{\theta,1}}$ and $\partial_t f_0$, Eq. (6.96), enables to get the evolution of $\overline{u_{\theta,1}}$ and f_0, and therefore h_0 through Eq. (6.98). The numerical integration scheme is analogous to the one presented in Section 6.4.5, except that one has directly an equation for $\overline{u_{\theta,1}}$ instead of only indirectly through $u_{t,1}$. The thick film model, Eqs. (6.96) to (6.100), is first integrated until the film

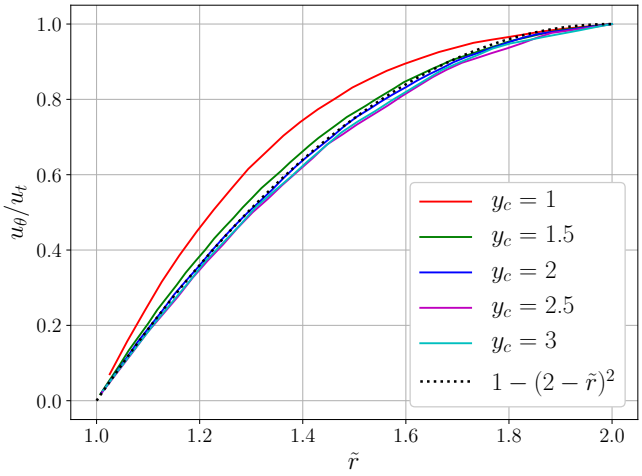
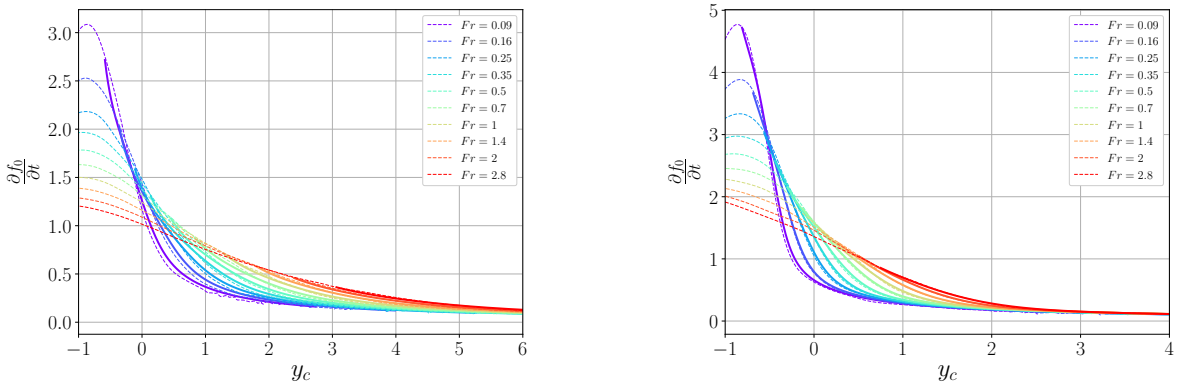


Figure 6.21: Normalized velocity as a function of the scaled radius defined in Eq. (6.115) in the thin film limit for the cylinder at $Fr = 1$ and $Re_D = 1000$; PFEM simulations at several positions y_c (continuous colored lines) and parabolic fitting profile, Eq. (6.116), assumed in the model (dotted black line).

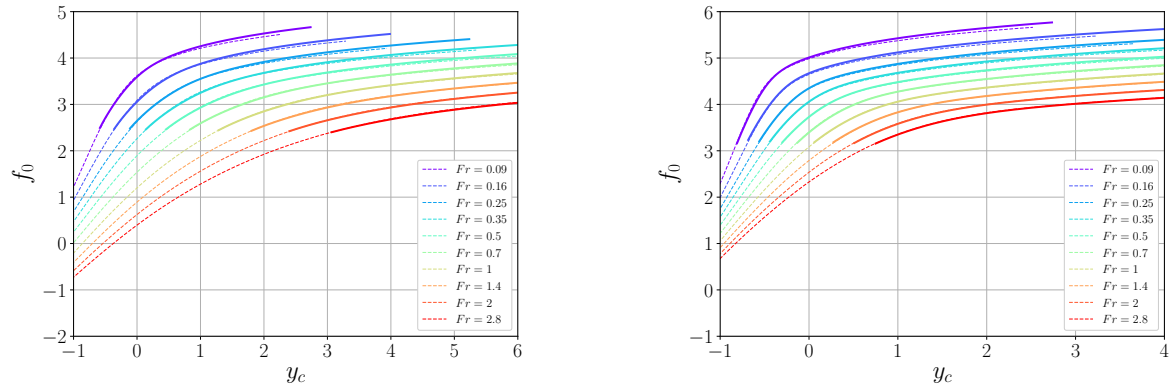
thickness h_0 reaches the boundary layer thickness δ. The corresponding time t (or position y_c) is then used as initial time at which the initial conditions for $\partial_t f_0$ and f_0 (and thus h_0) are imposed for the thin film model. These initial conditions could be obtained from the thick film model, but they are here taken from the PFEM simulations. The rational is to eliminate the error associated with the thick film model and thus to isolate the error only associated with the thin film model.

The thin film model predictions are shown in Figs. 6.22 to 6.24 and compared to the PFEM simulation results for several Froude numbers and $Re_D = 1000$. The model is able to capture the general trend very well despite a slight underestimation of the film thickness as time increases, which is more visible at small Froude number. Nonetheless, it should be emphasized that, here again, the dependency on the non-dimensional numbers has been derived a priori, and that no calibration has been required.



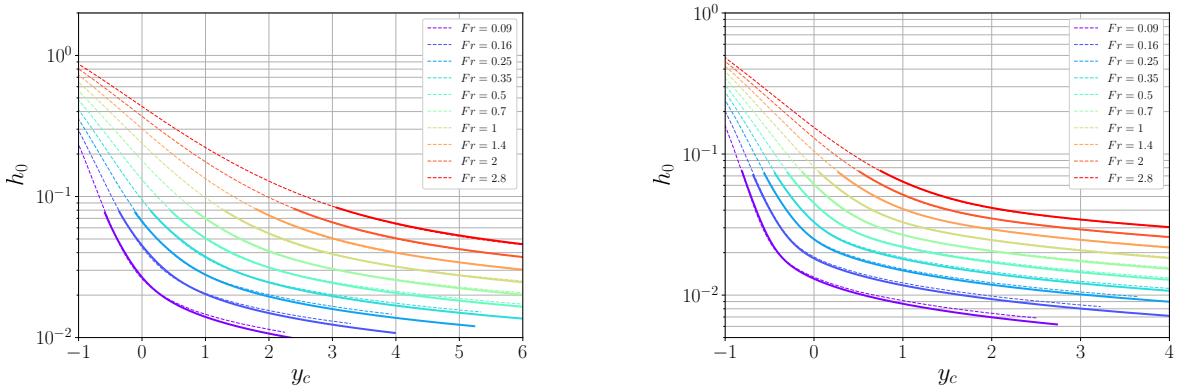
(a) **(b)**

Figure 6.22: Average azimuthal velocity $\partial_t f_0 = \overline{u_\theta}/\theta$ for the cylinder **(a)** and the sphere (twice the velocity in this case) **(b)** in the thin film limit at different Froude numbers and $Re_D = 1000$; PFEM simulations (dashed lines) and thin film model (continuous lines).



(a) **(b)**

Figure 6.23: Argument of the exponential representing the film thickness in the case of the cylinder **(a)** and the sphere **(b)** in the thin film limit at different Froude numbers and $Re_D = 1000$; PFEM simulations (dashed lines) and thin film model (continuous lines).



(a) **(b)**

Figure 6.24: Film thickness at the apex of the cylinder **(a)** and the sphere **(b)** in the thin film limit at different Froude numbers and $Re_D = 1000$; PFEM simulations (dashed lines) and thin film model (continuous lines).

6.6 Conclusion

In this chapter, a film model has been proposed to predict the film thickness above a rising cylinder and a rising sphere during the interface crossing. The model leverages the relation between the average velocity in the film, $\overline{u_\theta}$, and the film thickness h through the use of the mass conservation equation and a special Ansatz where h is expressed as a special function: the Lambert $\mathcal{W}$ function in the case of a 2D cylinder and a similar function in the case of an axissymetric sphere (see Eqs. (6.30) to (6.33)). In both cases, these special functions involve a decreasing exponential whose argument f is a function of t and θ. In the limit where $\partial_\theta h \ll 1$, the main assumption of the model and a feature of the present flow in agreement with the literature and PFEM simulations, $\partial_t f$ represents the average azimuthal velocity inside the film multiplied by the number of directions through which the fluid can flow, $\mathcal{D}$-1, $\mathcal{D}$ being the number of space dimensions (see Eq. (6.63)). Through a small angle approximation, the dynamics inside the film could therefore be described through a linear differential equation for f (mass conservation) and the θ-momentum equation for the free-surface tangential velocity u_t, the latter being related to $\partial_t f = (\mathcal{D}-1)\overline{u_\theta}$ through an assumed velocity profile. Inside the boundary layer, the profile is cubic in r, whereas in the bulk it has the form $u_\theta(r) = \frac{1}{r} + \frac{\delta}{(1+\delta)^2} r$, which has the implication that the viscous terms vanish from the θ-momentum equation. The dependency on the Reynolds number is given through the boundary layer thickness $\delta(Re_D) = kRe_D^{1/2}$ where the parameter $k = 2.5$ for the cylinder and $k = 2.21$ for the sphere has been obtained from PFEM simulations. It should be emphasized that

k is the only calibration parameter of the model. The proposed approach leads to a set of two ordinary differential equations, where the dependency on the Froude number stems from the initial conditions $h^*(Fr)$ and $(\partial_t f)^*(Fr)$, and from the gravity force term $\frac{1}{Fr}$ in the momentum equation.

The model has then been extended to the thin film limit, $h < \delta \ll 1$, where viscous terms dominate. In particular, the assumed velocity profile of the thick film model has been modified to a new assumed velocity profile, parabolic in r. In this case, the model integrate the dependency on both the Froude and Reynolds numbers without any calibration parameter.

It has also been shown for the cylinder case that the film model can be leveraged to provide an approximation of the pressure at the cylinder/sphere apex. This is achieved by considering the integrated r-momentum equation and the assumed velocity profile.

Finally, the predictions of both the thick and thin film models have been validated for different Froude and Reynolds numbers through comparison with PFEM simulations. Overall, a good agreement is observed. In particular, the model is able to reproduce very well the qualitative behavior of the solution and the quantitative discrepancies, inherent to the various simplifications made in the model development, remain limited. It has also been shown that, in the inviscid limit, the film thinning is both quantitatively and qualitatively different.

The main source of errors in the model include

- the uniform thickness assumption and corresponding small angle approximation,
- the assumed velocity profile and neglecting the viscous free-surface layer,
- the assumption that the boundary layer thickness does not depend on time and Froude number, and the corresponding approximation of δ.

Including terms of higher order than the leading order term in the small angle expansion could help address the first limitation. This however makes the mathematical development of the model significantly more complex. On the other hand, the motivation behind the assumed velocity profile approach was to avoid modeling the pressure. Taking higher order terms in θ into account could provide a way to model the pressure, and thus to by-pass the need for an assumed velocity profile, addressing the second issue. To address the third limitation, a better model for the boundary layer thickness that includes the dependence on time and Froude number should be developed. This might allow taking into account the influence of the initial release depth. These improvements are however left for future work.

At last, it should be mentioned that the present film model does not include the effect of surface tension on the film dynamics. From dimensional analysis, one expects surface tension to have a significant impact only during the late phase of the process, when small scale variations of the free surface appear. However, it has been found that surface tension tends to decrease the film thickness at small Froude numbers (Vincent et al. private communication). The probable explanation is the presence of a large free-surface curvature responsible for a depression on the body sides. This suggests that a strong interaction between the film and the bath is at play, a phenomenon that cannot be taken into account by a simple film model.

Chapter 7

Conclusions and Perspectives

This work aimed to study flows around bodies crossing a physical interface. To achieve this goal, the advantages of a Lagrangian formalism have been leveraged through the use of the Particule Finite Element Method (PFEM). As it has been shown, the state of the art in the application of this method is subjected to some limitations, such as the mass conservation linked to the remeshing and boundary recognition steps. It has so far also lacked a clear and well-defined mesh adaptation algorithm. The objective of this work was therefore twofold. On the one hand, it consisted in the development and implementation of a mesh adaptation technique for the PFEM so as to improve the mass conservation and efficiency of the method. This contribution included the validation of the overall algorithm for several two-dimensional cases as well as some suggestions and theoretical demonstrations of its possible extension to three dimensions. On the other hand, this work also consisted in the application of the newly developed algorithm for the study of the interface crossing of a body pulled out of a bath at constant velocity. In particular, the second part of this work has been dedicated to the in-depth analysis of a two-dimensional cylinder, and to a lesser extent, of an axisymmetric sphere, crossing a free surface.

7.1 Adequacy of the PFEM to simulate flows around bodies approaching an interface

The main challenges in the flows of interest are, as shown in Section 1.3, the presence of a deforming interface and its interaction with the moving body. In this context, unlike Eulerian or Lagrangian-Eulerian approaches, the PFEM does not require a particular boundary tracking algorithm as the equations directly dictates the interface displacement, which confers the PFEM an advantage over other methods. Nevertheless, the drawback is that the mesh nodes are subjected to large displacements so that the computational domain needs to be periodically remeshed to avoid too large mesh distortion that would otherwise lower the stability and accuracy of the time-integration. To deal with this aspect without compromising (too much) the code efficiency, triangular and linear finite elements are typically used, such that fast Delaunay triangulation can be performed for remeshing the domain. Afterwards, the boundaries are automatically identified through the boundary recognition algorithm, i.e., the α-shape technique, by identifying badly-shaped triangles adjacent to the physical boundaries. This last feature of the method is precisely the main source of the mass conservation error linked to remeshing.

7.1.1 Limitations of the PFEM

Mass conservation represents the main challenge in the PFEM. Corresponding errors can be divided into two types. The first type of error is not specific to the PFEM and is referred to as the "numerical mass conservation error". It is the error associated with the time integration. In this thesis, the error has been demonstrated to be directly related to the invariants of the velocity gradient tensor, each order of Δt being proportional to one of the invariant (see Eqs. (3.67) and (3.68) in Section 3.3). In particular, the first order in Δt is proportional to the residual of the incompressibility constraint, which is larger in the presence of stabilizing terms. This intuitively explains why such stabilizing terms have some negative impact on the numerical mass conservation. The second type of error is very specific to the

use of the α-shape technique for the boundary recognition, and is referred to as the "remeshing mass conservation error". As it has been shown, the α-shape technique leads to adding elements in merging fluid regions and deleting elements where the free surface gets stretched, which respectively translates into mass creation and destruction. This work has focused on improving only this second type of error, as it is more often the major source of mass conservation errors in the PFEM.

Another limitation of the PFEM is that it does not provide a standard framework to include mesh adaptation, i.e., local dynamic refinement or coarsening of the mesh.

7.1.2 Improvement of the PFEM

In order to incorporate mesh adaptation capabilities in the PFEM while decreasing the remeshing mass conservation error, boundary nodes tagging has been combined with a prescribed local mesh size. In this framework, the creation/destruction of mesh elements has been limited as much as possible and is only performed when the mesh resolution approaches the nearby/minimal target size.

7.1.2.1 Target mesh size or size field

The first step of the method consists in defining the size field[1] (or target mesh size). Two main approaches have been proposed. The first one is geometry-based and consists in interpolating the prescribed size field between bounds based on distances from regions of interest (see Section 3.2.1.1). In the second approach, the size field is prescribed based on the flow solution itself. In this thesis, the norm of the velocity gradient tensor and the free-surface curvature have been used (see Section 3.2.1.2), but other quantities could also be considered.

7.1.2.2 Improved boundary recognition algorithm

The second step of the method aims at improving the mass conservation properties of the boundary recognition algorithm. In the traditional PFEM, the information about the boundary location is not conserved during the remeshing step. In the new method, boundary nodes are tagged. Moreover, several criteria based on the number of boundary nodes belonging to a given element have been added to the classical α-shape technique so as to ensure that elements are added/removed only when it is absolutely necessary, e.g., to simulate merging or separation processes. Moreover, when performed, the creation/destruction of mesh elements at the boundary is accomplished at a high mesh resolution owing to the mesh refinement/coarsening technique.

This overall procedure, as discussed in the validation chapter, decreases mass destruction more strongly than the mass creation but the resultant mass variations remain below the current state of the art for the PFEM[2].

7.1.2.3 Node insertion/removal algorithms

Based on the size field and the location of the boundary nodes, the mesh refinement/coarsening to approximately achieve the target mesh size is performed by adding/removing nodes in specific locations. The addition of nodes is based on a cell subdivision while the removal of nodes on a cell collapse. These approaches respectively refine/coarsen the mesh according to how the element areas compare with the prescribed size field (see Section 3.2.2), using specific thresholds. In particular, the choice of these thresholds ensures that refinement and coarsening regions do not overlap, and that a compromise is achieved between the coarsening-induced numerical diffusion and the level of element distortion.

Finally, this addition/removal of nodes also includes specific treatments near the boundaries so as to ensure good element quality there, which is beneficial for mass conservation.

[1] This terminology is used here as it better matches the existing literature on the subject.

[2] Note that this claim does not include the most recent developments, such as, for instance, Leyssens et al. [133] and Fernandez et al. [134].

7.1.3 Validation

The mesh adaptation algorithm has been validated through several two-dimensional test cases. The first two do not include a free surface and therefore only evaluate the performance of the mesh refinement/coarsening algorithm, while the last three also assess the improvement brought by the new boundary recognition algorithm.

Flow around a cylinder

The first case was the flow around a cylinder with different variants described in Section 4.1: a static cylinder ($Re_D \in \{40, 80, 120, 160, 200, 9500\}$), a rotating cylinder ($Re_D = 200$) and a transversally oscillating cylinder ($Re_D = 200$). For each case, results from the literature have been successfully reproduced despite the use of a limited number of mesh elements thanks to the mesh refinement algorithm, demonstrating its interest for reducing the computational cost. This is specifically the case for external flows around bodies because the major part of the fluid domain may be discretized by coarse elements as they are located far away from boundary layers and wake of these bodies.

Lid-driven cavity flow

The second case is the lid-driven cavity flow ($Re \in 100, 400$). For this test case, the interest of the normalized version of the solution-based mesh refinement (see Eqs. (3.23) and (3.24)) is demonstrated by its ability to capture small flow features, such as the vortices located at the cavity corners, unlike the case with the "non-normalized" version. In particular, the position and intensities of the vortices are in agreement with the results of the literature. Nevertheless, the gain in terms of solution accuracy over CPU time ratio is not very significant in this case, as refinement is required in many regions of the flow to capture all the shearing layers.

Forced sloshing

For the third case, the forced sloshing of a liquid in a reservoir oscillating harmonically has been studied. The efficiency of the mesh refinement based on the free-surface curvature is demonstrated by its ability to correctly capture the free surface deformation with a limited number of elements. The time evolution of the free surface deformation and the pressure at the wall have been compared with both experimental results and other PFEM simulations with a uniform mesh. In particular, it has been shown that the new mesh adaptation method leads to results (free-surface deformation and impact at the wall) closer to the experiment than those obtained with a uniform mesh, while requiring much fewer mesh elements. In addition, mass conservation also turns out to be better, even if some creation of mass still persists because of the successive wetting of the lateral walls. Despite the undeniable improvement brought by the new algorithm, this remaining mass creation caused by the wetting of the walls indicates that there is still room for further improvement, which is left for future work.

Drop falling into a liquid bath

The fourth case, the drop falling into a liquid bath, solely addresses the correct evolution of the free surface and the mass conservation properties of the new algorithm. It also illustrates the effect of the collision anticipation algorithm described in Section 3.2.3.2, in which the merging of two approaching free surfaces is anticipated. Around the expected merging region, the smaller mesh size is propagated so as to limit the mass creation occurring during the merging process. In addition, the combination of different geometrical and solution-based refinement criteria enables refining the region of high free-surface curvature and high velocity gradients. The algorithm results in better mass conservation properties than with a uniform mesh, while requiring fewer mesh elements. Unfortunately, for that case again, the wetting of the lateral walls still results in some increase of the total mass.

Cylinder rising toward and crossing the interface

The last validation case is the rising cylinder toward the free surface, i.e., the main case of interest for the second part of the present work. In this test case, the new mesh adaptation method enables to gradually refine the fluid region in between the free surface and the approaching body. Combined with the new boundary recognition algorithm, it results in very good mass conservation up to liquid films of thickness comparable to the minimal mesh size, thanks to the nodal insertion in the middle of free-surface edges. On the contrary, the classical α-shape technique induces significant mass destruction due to deleted obtuse triangles following the stretching of the free surface. The improvement achieved by the new algorithm regarding the free-surface stretching is therefore clearly demonstrated. In particular, the good behavior of the new algorithm regarding surface stretching makes it well suited for the study of the free-surface deformation and the flow inside the film above the body's apex.

7.1.4 Concept of nodal density and extension to 3D

The new mesh adaptation algorithm is limited to two-dimensional meshes, which are generally easier to tackle than three-dimensional meshes. Nevertheless, based on theoretical considerations, the extension of this mesh adaptation algorithm to three dimensions has been discussed. The general idea relies on the claim that a quantitative measure of the nodal density (i.e., the number of nodes per unit of area/volume) at the discrete level can be performed by interpreting internal angles (solid angles in 3D) at cell vertices as nodal fractions. The total nodal fraction per element is then simply the sum of the internal angles (solid angles) divided by 2π (4π) and the nodal density is this total nodal fraction divided by the element area (volume). Using this novel concept of nodal density and fundamental results of discrete geometry (e.g., Euler invariant, Gram-Euler theorem), several relations for 2D and 3D triangular meshes have been derived. The application of this concept to mesh refinement/coarsening has then been discussed and preliminary 3D algorithms have been proposed. The extension of the new boundary recognition algorithm to 3D then follows in a natural way. Finally, the problematic case of slivers has also been briefly discussed.

7.1.5 Perspectives for further improvement of the PFEM

While the proposed mesh adaptation algorithm successfully enables the reduction of remeshing mass conservation errors while increasing the accuracy over CPU time ratio, it is still subjected to some limitations, such that further improvement of the proposed approach or development of alternative approaches are possible. The first one consists in (1) improving the code efficiency. In addition, other improvements also include the possibility to (2) avoid complete re-triangulation of the entire domain by performing local mesh modifications, (3) use anisotropic mesh refinement, (4) improve boundary conditions on fixed boundaries, (5) use another boundary-recognition technique, (6) extend the proposed algorithm to three dimensions.

7.1.5.1 Improvement of the code efficiency

The code has been developed as a "proof of concept" and is thus not optimal in many regards, for technical and historical, rather than theoretical, reasons. In future developments a special attention should be put on the following aspects to improve the code efficiency:

- The parallelization of relevant parts of the code (excluding remeshing step). While some parts of the code have been parallelized (e.g., the building of the elemental matrices), this is not the case for all parts. Increasing parallelization will help decrease the computational cost of large simulations.

- A better data storage management. So far, some intermediate data generated by the remeshing are stored on disk, which should be avoided by keeping all the information in memory during code execution. This could be done by interfacing the remeshing library (Triangle) with the solver

instead of executing the library from the command line, unavoidably leading to some data written on disk.[3]

Note that other ways of improving the solver efficiency could be considered, but would affect the overall algorithm. For instance, parallelizing the remeshing step requires dividing the mesh into sub-meshes with interface boundaries between them, such that the overall remeshing output would be affected by the process and would not be equivalent to a remeshing performed in series.

7.1.5.2 Avoiding unnecessary re-triangulation

In this work, each time the mesh became too distorted, it was fully discarded and replaced by a a completely new mesh, considering the triangulation algorithm as a black box, as it is the case in the classical PFEM approach. Although it simplifies the implementation of the mesh refinement/coarsening algorithm as the triangulation data structure is automatically updated after each re-triangulation without any specific local treatment, it might not be efficient when the mesh in significant parts of the computational domain remains of good quality. Another alternate approach could avoid such complete re-triangulation, and therefore reduce the computational time associated with the remeshing step of the PFEM.

For instance, as already discussed in Section 3.1.2, another way to combine the α-shape technique with the use of non-uniform meshes is to use regular triangulation together with the weighted α-shape technique. The interest of the method relies in the possibility of removing nodes within the mesh refinement algorithm, such that the node removal process could potentially be automatized. In other words, the triangulation is performed at the same time as the node removal, and the remeshing after the node removal is therefore unnecessary, reducing the total mesh coarsening and remeshing CPU time. On the other hand, the local mesh modifications required to avoid remeshing after node insertions is usually much easier to tackle than after node removal. More specifically, the elements impacted by the new nodes are easily identified such that the required reconnections are easier to perform.

The main challenge of the overall approach relies in the development of a fully integrated remeshing algorithm that is as efficient as the one provided in existing libraries. Within this formulation, not only the fluid elements would remain in memory, but also the empty elements, such that the entire convex hull of the cloud of nodes would be known at any time.

A related method is currently developed by Leyssens et al. [133], where the node addition is performed as part of the remeshing using Delaunay refinement. It consists in adding nodes at the center of the circumscribed circles of the triangle that need to be refined (see Ruppert's and Chew's algorithms for some examples [135]),or on the middle of the boundary edges in case the center of the circumscribed circle is located outside the fluid domain.[4] With this method, when the mesh has been cleaned using Delaunay refinement, the time integration is performed and nodes are displaced, the α-shape criterion for boundary recognition being only applied subsequently. If mesh element quality is preserved after the nodal displacement has occurred, the α-shape technique should behave well and should not produce artifacts as empty holes in the fluid domain. In practice, it implies the use of a small time step and/or mesh adaptation at each time step to ensure that the mesh quality is preserved and to prevent the occurrence of artifacts. By opposition, in the present work, this requirement is not absolutely necessary for the method to operate, although it remains a good practice.

7.1.5.3 Anisotropic mesh refinement

Another possibility of improvement concerns the metric-based mesh adaptation (see Section 3.1.1.3). In the present work, the metric consisted in the single parameter L^*, the target mesh size, which can be imposed based on both geometrical and physics-based criteria. In the general metric-based mesh adaptation formalism, the mesh elements not only satisfy a global scaling L^*, but can have different size in different directions, which is determined by the local metric tensor. While this may be straightforward in the case of structured meshes, the use of metrics for the PFEM mesh adaptation can be challenging for several reasons. (1) The criterion for the addition and removal of nodes should be based on different edge

[3]This had been planned but was later abandoned due to other conflicting work packages.

[4]In some way, it is similar to the boundary edge splitting proposed in the present work in Section3.2.3.1.

length thresholds depending on the edge direction with respect to the eigenvectors of the metric matrix, rather than using the area of the elements. (2) The Delaunay triangulation should be performed on an isotropic space (e.g., if an anisotropic metric is used, on the abstract space (u, v) rather than on the real physical space (x, y)), which may be challenging from a computational point of view. (3) The solution-based criterion to define the metric should be adapted in order to reduce the computational errors. This could be done using the Hessian matrix associated with the different variables (see Appendix B to see how to compute it). The challenge would then be to define the metric using a clever combination of the different Hessian matrices associated with the variables. (4) The node displacement, due to the Lagrangian nature of the PFEM, can lead to the rotation of some mesh elements, such that the size of these elements can become too large or too small in a specific direction, just because of this rotation relatively to the metric. In the worst case, this could induce cycles of node insertions and deletions, which would in turn produce significant numerical diffusion. Therefore, another challenge would be to take this effect into account to properly adapt the metric so as to avoid these node addition and removal cycles.

In summary, even if the use of anisotropic metric has, in theory, the potential to further increase the accuracy over CPU time ratio, it faces many challenges in the context of the PFEM. In particular, it is not clear whether the small potential improvement is worth the effort in regard to all the challenges to overcome, which is the main reason why it has not been considered in the present work. This question is therefore left open for future research on the PFEM.

7.1.5.4 Improved boundary conditions on fixed boundaries

In the PFEM, the mass conservation is not only affected by the boundary recognition algorithm used to follow deforming interfaces, but also by the way the fixed boundary conditions are treated, in particular to describe wetting and de-wetting phenomena. There are therefore perspectives of further improvement in that regards, for both free-slip and no-slip boundary conditions.

Free-slip boundary condition

In the context of the PFEM, the free-slip boundary condition is useful when one can neglect viscous effects near walls, which would otherwise requires a small mesh size to correctly capture the boundary layer (e.g., fluid in large reservoirs, channels, etc...). By imposing a free-slip boundary condition, one can therefore use elements much larger than the boundary layer adjacent to solid walls, thus saving precious computational time for many practical applications in which the boundary layer does not significantly impact the physics of interest (e.g., in the imposed sloshing or the droplet fall test cases presented in Chapter 4).

In the original PFEM solver used here, the free-slip boundary condition was first only possible on horizontal and vertical surfaces, and simply consisted in removing one of the two velocity degrees of freedom. To impose it on an inclined surface, one first has to rotate the degrees of freedom using local rotation matrices, and then simply remove the degree of freedom of the displacement along the normal direction to the surface (similarly to horizontal and vertical surfaces). Such an approach has been re-implemented in this work for the sloshing test case, based on the work of Cerquaglia [1, 24]. However, the results have not been shown as the method failed at the second splash on the left wall (see Section 4.3). The problem with the free-slip boundary condition is that the walls are fictitious and are thus not represented by nodes, unlike for the no-slip boundary condition. To solve this issue, the walls where free slip occurs should be represented as lines in memory, and the fluid crossing these lines should somehow be reattached to them, the free-slip condition (removal of one degree of freedom) being imposed after the reattachment is complete. Such an approach has for instance been implemented by Leyssens [133], which improved the mass conservation properties of the PFEM. Note that Cerquaglia et al. [24] used another approach based on contact elements, where the free-slip nodes slide along the solid surfaces described by nodes, but it is not a computationally accurate approach. Specifically, the contact elements constitute some non-physical empty space between the fluid and the sliding wall. As a consequence, the sliding motion of the fluid particles on top of these elements does not always occur at the same distance from the boundary, and in case of curved boundaries, does not rigorously follow the boundary shape. In that respect, another possible improvement of the free-slip boundary condition is

precisely its extension to curved surfaces. Such surfaces are challenging within a Lagrangian framework, as the surface nodes are moving tangentially to the surface. One cannot simply remove the normal degrees of freedom because the tangential motion of particles would lead to the deterioration of the curved surfaces. Instead, the normal velocity could be constrained based on the tangential velocity. For instance, it can be shown that, for a surface with a local radius of curvature r, the normal velocity u_n can be related to the the tangential component u_t and the time step Δt through the relation

$$u_n = \frac{u_t^2 \, \Delta t}{r} \,, \tag{7.1}$$

such that the particle remains on the same arc of circle or radius r. The idea would then be to replace the momentum equation along the normal direction by the constraint, Eq. (7.1), and then to apply some small correction to take into account the variability of r along the surface. This would require to have the exact analytical shape of the surface in memory, with all the complications it may imply. Note that while this suggestion seems feasible in theory, it might not work in practice[5].

No-slip boundary condition

The Dirichlet (no-slip) boundary condition is also challenging in the context of the PFEM. This is particularly the case when fluid is attaching or detaching from dry walls. To describe the wetting of a wall, the method described in this work consists in creating some elements adjacent to the wall to make the junction with the fluid, generating undesired mass conservation error. This issue can be dealt with similar developments to those of Leyssens et al. [133], reattaching the fluid nodes on the wall, the latter having its own mesh-independent geometrical description. However, another problem subsists in the case of the no-slip boundary condition: the de-wetting of the surface is no longer possible as one prevents the attached node to detach from the surface. Recently, solutions have been proposed by Rizzieri et al. [136] to address this issue. It consists in allowing the non-sliding nodes to detach from the boundary when the average velocity of adjacent nodes projected into the normal direction reach a given threshold. To conclude, the overall promising effect of the approaches of Leyssens et al. [133] and Rizzieri et al. [136] is to decrease the mass conservation error in the presence of wetting/drying boundaries, when free-slip and no-slip conditions are respectively imposed on those.

7.1.5.5 Alternative boundary recognition algorithms

One of the major issue faced in this work was the tendency of the α-shape algorithm, a purely geometrical criterion, to cause mass creation and destruction close to boundaries. While it has been possible to improve the α-shape technique based on additional criteria involving boundary node tagging, alternative approaches could be considered. In particular, the drawback of the α-shape technique in the PFEM lies in its purely geometrical nature, so that more physical approaches may be better suited for the boundary recognition. A recent example of improvement in the context of the PFEM is the use of the level set function as a replacement of the $\alpha-$shape technique [134]. The method shows a significant improvement in mass conservation, in particular for the drop falling into a bath (see also Section 4.4), and seems to be a promising step in the right direction.

7.1.5.6 Extension to three dimensions

The main avenue for further improvement concerns the extension of the ideas presented in the first part of Chapter 3 to three-dimensional tetrahedral meshes. One important aspect in three dimensions is that the remeshing takes significantly more CPU time, such that re-triangulating the whole cloud of nodes each time the mesh is locally too distorted should be avoided. Instead, local mesh modifications should be preferred, as already discussed for two-dimensional triangular meshes in Section 7.1.5.2. For the rest, the extension to 3D would consist in implementing the ideas developed in Section 3.4, taking into account the goal of avoiding total re-triangulation of the domain. The reader is referred to Section 3.4.4 for the boundary recognition algorithm and the addition/removal of nodes. Additional aspects

[5]For instance, in the case of inclined flat boundaries, one could keep the original horizontal and vertical components u_x and u_y and constrain u_y based on u_x, or conversely. However, the experience shows that it does not work in practice, as "no control is given on what is happening in the rotated frame of reference" [24].

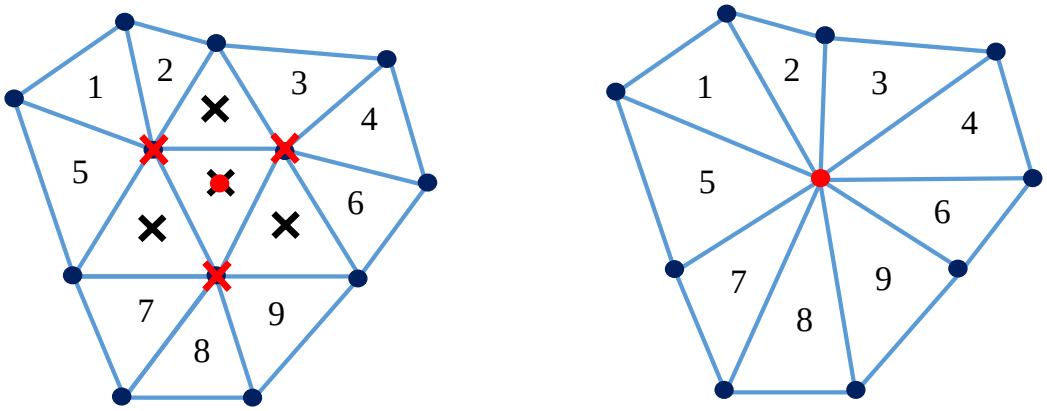


Figure 7.1: Illustration of a variant of the node removal algorithm using the cell-collapse approach, in which node removal is done directly, instead of tagging the elements and removing them within another loop. In this illustrative example, the central element has been detected to be too small (too high local nodal density), and is deleted (smaller black cross) and replaced by a new node (red dot). Its three associated nodes are also deleted (red crosses). All the adjacent elements sharing an edge are also removed (black crosses), while those only sharing one node (elements from 1 to 9) replace this node by the new one (red dot). The new mesh is shown on the right.

should furthermore be considered to enable local remeshing (as already mentioned in the perspectives of Section 7.1.5.2, not only the fluid elements but all elements should be kept in memory).

On the one hand, when a node is added, on a boundary face or on an edge, as suggested by the algorithm described in Section 3.4.4, the data structure should be modified locally to incorporate the new elements, faces and edges associated with this node insertion, and to delete those that are obsolete. This generally leads to a local non-Delaunay triangulation, such that a few more operations are necessary to restore the Delaunay triangulation, if desired[6].

On the other hand, node removal is a little bit more complex because the removal of tetrahedral cells and associated nodes directly affects the adjacent cells. For instance, problems arise when nodes should be removed in two or more regions close to each other, as conflicts between these regions may exist. To reduce this problem, the modification of the data structure should be local and immediate, as illustrated in Fig. 7.1 for the 2D case. In this respect, the cell-collapse approach proposed in this work has an advantage in terms of simplicity for the node removal algorithm: one simply has to replace the nodes of the collapsed element by the node in its middle, which is the single modification to perform for the adjacent elements (elements 1 to 9 in Fig. 7.1). The elements having previously two nodes or more among the deleted nodes are simply deleted (black crosses). This should not lead to any problem in terms of inverted elements in 3D, as the algorithm is very similar to that for two-dimensional meshes. The only drawback is the introduction of sharper angles at the inserted node, which can be mitigated by taking into account that the other nodes of Fig. 7.1 may be removed based on the coarsening criterion applied to other elements adjacent to those shown in the figure, which would expectedly increase the angles at the red central node. Again, in case the resulting mesh is not Delaunay, a few more operations can be performed locally to make it Delaunay, if it is desired.

Note that another advantage of the proposed cell-collapse method is that the nodal insertion successfully increases the area of all neighbour elements (from 1 to 9 in Fig. 7.1), thus limiting the potential diffusion by subsequent node removal

7.2 Flows around symmetrical bodies rising toward a free surface

The second part of the present work consisted in a specific application: the flow around a cylinder and an axisymmetric sphere rising towards and crossing a free surface at constant velocity. For this problem, both the free-surface elevation and the vertical forces on the body were investigated, as it is classically the case in the literature (see Section 5.1). Simulations have been used to analyze the influence of the

[6]The Delaunay triangulation in 3D has not the optimal properties of the Delaunay triangulation in 2D, such that it is not detrimental to just leave the mesh like this, and solve the problem on a non-Delaunay mesh.

Reynolds and Froude numbers. Comparison with experimental measurements have also been proposed. Finally, a reduced-order model has been developed to predict the film thickness above the body's apex (Chapter 6).

7.2.1 Free-surface elevation

First, it has been shown that the surface elevation $y_s(t)$ (i.e., the distance of the highest point of the free surface with respect to the initial free-surface level) mainly depends on the inertia-to-gravity balance, represented by the Froude number,

$$Fr = \frac{U^2}{g\,a}\,, \tag{7.2}$$

where U is the rising velocity of the body, g the acceleration of gravity, and a a characteristic size of the body (here the radius of the cylinder/sphere). In practice, this has been shown in Chapters 5 and 6 by considering the variation of the distance between the cylinder apex and the top of the free surface, $h(t)$. In particular, a correlation involving the Lambert $\mathcal{W}$ function has been fitted for $h^*(Fr)$, h^* being defined as $h(t)$ at the time when $h(t)$ and $y_s(t)$ coincide (i.e., the cylinder apex is at the level of the initial free surface), for small values of the Froude number ($Fr \leq 1$). Although the Froude number is the most important parameter in this case, the Reynolds number Re_D and the release depth d also have an indirect impact on the surface elevation through the wake size j^*. In particular, the surface elevation is larger for a larger wake size $j^*(Re_D, d)$, i.e., for larger initial release depth d and smaller Re_D. By opposition, the wake is almost not affected by the presence of the free surface, except during and after the interface crossing ($y_c > -1$).

To predict the variation of $h(t)$ for different Froude and Reynolds number, radially averaged models have been proposed in Chapter 6. For both the cylinder and the axisymmetric sphere, it has been shown that the function $h(t)$ could be expressed with the help of some special function $\mathcal{S}$ applied to an exponential:

$$h(t) = \mathcal{S}(\exp(-f(t)))\,. \tag{7.3}$$

In particular, the function $\mathcal{S}$ is the Lambert $\mathcal{W}$ function for the case of the cylinder, and another "$\mathcal{W}$-looking" function in the case of the sphere. This formalism enabled to greatly simplify the mass conservation equation, under the assumption that $\partial_\theta h \ll 1$, which is known in the literature to hold for the cylinder, but which is more generally a good approximation for a smooth body surface symmetrical around its apex[7]. The simplification yielded an equation relating the argument of the exponential, $f(t)$, and the cross section average (radial average in 2D) of the azimuthal velocity u_θ, in particular in the small angle limit from the cylinder apex:

$$\frac{\partial f}{\partial t} = (\mathcal{D}-1)\frac{\partial \overline{u_\theta}}{\partial \theta}\,, \tag{7.4}$$

$\mathcal{D}$ being the dimension of the problem.

7.2.2 Kinematics of the film above the body's apex

For the film regime where $h(t) < 1$, two similar kinematic models have been proposed in Chapter 6 for the cylinder and the sphere. These models leverage the momentum equation along θ evaluated at the free surface, where the free-surface velocity u_t is related to the average velocity $\overline{u_\theta}$ through an assumed velocity profile. In particular, this velocity profile can be characterized as a cubic function in the boundary layer of thickness δ, followed by a velocity profile of the form $1/r + \delta r/(1+\delta)^2$. In this film model δ is the only quantity that requires calibration and has been assumed to have a $Re_D^{-1/2}$-dependency and no dependency on the Froude number, as well as a negligible variation in time. Note that questioning these assumptions and proposing a better closure model for δ could be a possible improvement of the present model. This approach enables to neglect the influence of the pressure as this pressure is constant at the free surface, which greatly simplifies the equations.

[7] Which is the case for both the cylinder and the sphere.

It has been shown that the evolution of $h(t)$ above the cylinder/sphere and the corresponding average velocity $\overline{u_\theta}$ are well captured for Froude numbers ranging from 0.25 to 2 and Reynolds numbers ranging from 500 to 4500, which demonstrates the interest of the proposed film model.

Additionally, when the film reaches a thickness smaller than the boundary layer,

$$h \simeq l_\nu < \delta(Re) \tag{7.5}$$

i.e., when it is of the order of the viscous-to-gravity length scale, the model becomes less accurate as it neglects the dynamics of the boundary layer, the latter becoming determinant in the dynamics of the film. For that regime, the film model has been modified, assuming a parabolic velocity profile over the entire film thickness (with the maximum of the parabola at the free-surface). This has been shown to correctly predict the film thickness in the thin film limit, for both the cylinder and the sphere.

7.2.3 In-depth dynamics

For the dynamics occurring "in depth", i.e., when the cylinder apex is below the initial free surface ($y_c < -1$), the main challenge was to understand the effects of (1) the free surface at constant pressure and (2) the gravity field. In particular, the impact of these two effects has been analyzed by comparison with the flow around the same body, moving at the same speed in an infinite fluid domain. It has been shown that these two effects have different relative contributions depending on the Froude number.

For small Froude number ($Fr < 1$), the effect of gravity dominates, and the bulge forming when the body approaches the free surface tends to accelerate the flow perpendicularly to the motion of the body around its apex (larger values of $\partial_t f$ in Chapter 6), as illustrated in Fig. 7.2a. The weight of the bulge also tends to increase the drag and the skin friction coefficient (as shown in Chapter 5), in comparison to the impulsively started cylinder in an infinite domain at the same Reynolds number. The added drag C_g coming from the free-surface deformation subjected to gravity has been called the "gravity drag", and has been approximately related to the free-surface elevation y_s.

For large Froude number ($Fr \gg 1$), the effect of gravity is small and the main effect stems from the free surface at constant pressure, which results in an easier displacement of the free-surface material line in comparison to the same material line within an infinite fluid domain, as illustrated in Fig. 7.3. This results in a deceleration of the flow perpendicularly to the motion of the body around its apex, as well as a diminution of the drag and skin friction coefficient, as illustrated in Fig. 7.2b. The drag deficiency or negative drag C_{fs} coming from the free surface at constant atmospheric pressure has been called the "free-surface drag", and has also been related to the free-surface elevation y_s. Note that both the effects of C_g and C_{fs} are to be taken into account simultaneously for an arbitrary Froude number, but this has not been investigated in the present work. It could however be considered for future work, as discussed later in Section 7.2.6.1.

7.2.4 Comparison with experiments

Finally, some comparisons with experiments using oil have been performed at Reynolds numbers below 1200 for the cylinder case. Both the experimental film thickness $h(t)$ and drag force have been compared to those predicted by the simulations. Despite some discrepancies arising from the 3D nature of the flow in the experiments (finite aspect ratio of the cylinder), a good match has been obtained for $h(t)$. For the drag, several observations have been made. When the cylinder was below the free surface ($y_c < -1$), good match between experiments and simulations has been observed, except during the acceleration phase, or when the wake became unstable. In particular, the acceleration delayed the maximum drag in comparison to that obtained very soon after the impulsive start in the simulations (see Fig. 5.34). For the experiments, the drag reaches its maximum very soon after the end of the acceleration phase, and its maximum is located very close to the corresponding value obtained in the simulations for the same cylinder position y_c.

During the interface crossing and after, a local maximum of the drag was observed when the wake thickness below the cylinder reached a value around 1.2 radii, except for the smaller velocities where it was slightly smaller. It has been shown that this maximum drag was due to a suction effect resulting from a decreasing pressure below the cylinder. This effect is maximal at a time t^* when the product of

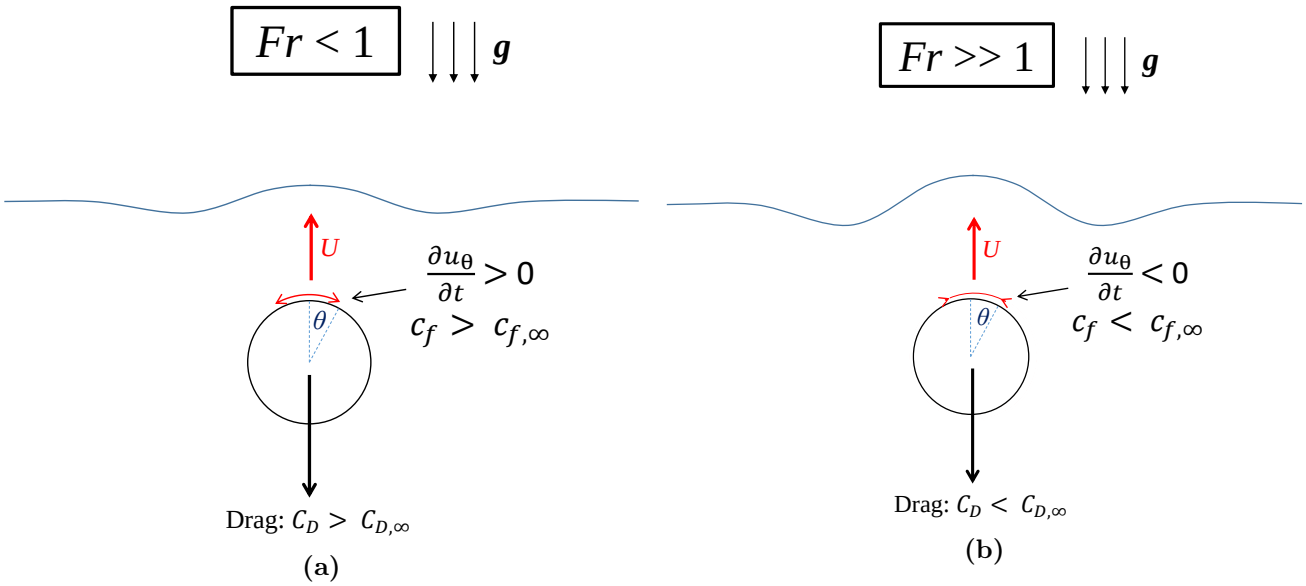


Figure 7.2: Schematic comparison of the flow dynamics around a rising cylinder between two regimes where the Froude number is either smaller or much larger than 1. For small Froude number (**a**), the gravity effects are large. As the cylinder/sphere approaches the free surface, it results in an increase of the azimuthal velocity u_θ, as well as of the drag and skin friction coefficients, compared to the flow around the same body (same speed and same Reynolds number Re_D) in an infinite fluid. For Froude number much larger than 1 (**b**), the gravity effects are small and the main effect is the easier pushing of the flow above the cylinder, due to the free surface at constant pressure, compared to the case in an infinite fluid. Consequently, as the cylinder approaches the free surface, there is a progressive decrease of the azimuthal velocity u_θ, as well as of the drag and skin friction coefficients, compared to flow around the same body in an infinite fluid.

low pressure and wake width Δ is maximal. This also explains why the maximum observed drag was proportional to this wake thickness at that time t^*, and why the drag progressively decreases as the wake thickness decreases afterwards. In addition, the weight W of the annulus of fluid around the cylinder at the time t^* is also proportional to this wake thickness at that time, which explains the correlation between this weight and the maximum drag, where an average ratio of 1.4 has been found between both. Finally, if one divides the forces by W (given by Eq. (5.17)), and use a common "gravitational" time $t_g = \sqrt{\frac{g}{a}}\, t$ for each curve, all the curves collapse, as shown in Fig. 5.41, indicating that the drainage dynamics in the film is driven by gravity.

7.2.5 Case of a highly viscous fluid

The case of the rising cylinder has also been studied in the very small Reynolds number regime ($Re_D < 1$), for a constant value of the Froude number, $Fr = 1$, where, by opposition to the previous cases, it has a strong impact on the free-surface deformation. A power law has been obtained for the wake width $w(t)$, which can be derived from a very simple stretching model. For the different Reynolds numbers, the wake reaches, after the interface crossing ($y_c > 1$), a characteristic length L_w, which is defined as the distance travelled by the cylinder, starting from a release depth of one diameter (i.e., with a thickness of one radius of fluid initially above the cylinder) until the wake thickness reaches a size of one diameter. Here again, L_w is related to Re_D through a quasi-power law, obtained empirically.

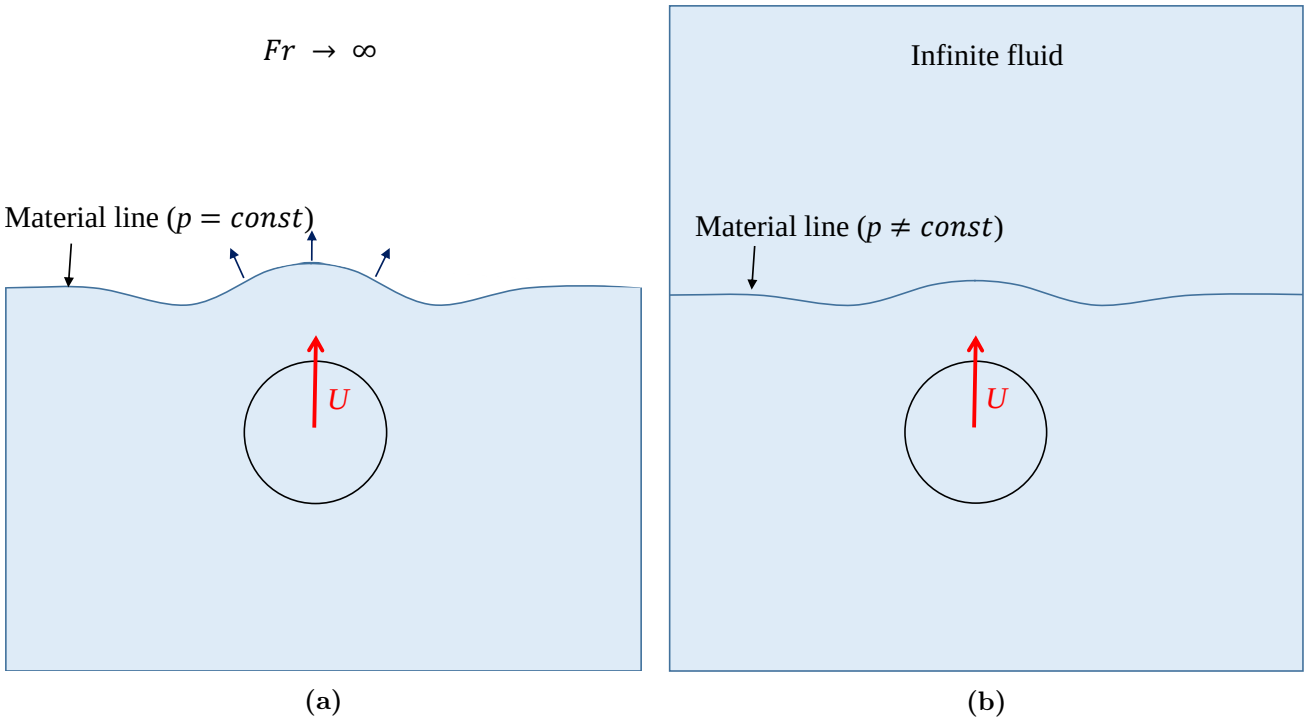


Figure 7.3: Schematic illustration of the easier displacement of a material line at constant atmospheric pressure (i.e., the free surface), when a moving body approaches it (**a**), in comparison to the displacement of the "same" material line in an infinite fluid domain (**b**).

7.2.6 Perspectives for future studies

While significant efforts have been made to cover in depth the subject of interface crossing, there is still much room for improvement regarding this topic. In particular, one can cite: (1) the drag model during the rise of the cylinder below the free-surface, (2) the wake dynamics after the interface crossing, (3) the film dynamics above the apex of the body. Perspectives regarding these three flow features are given in the following paragraphs.

7.2.6.1 Drag and skin friction dependency on Fr

Considering the total drag C_D on the cylinder as well as the skin friction c_f along its surface, increased values of these quantities have been observed at small Froude number, with respect to those obtained in an infinite domain for the same Reynolds number, i.e., with more or less the same wake dynamics. By opposition, lower values have been obtained at large Froude number. The increases and decreases of the total drag, respectively at small and high Froude number, have been roughly correlated with the free-surface elevation y_s, but the limited sampling among the possible values of the Froude number prevented the development of a solid model valid for a wide range of Fr.

Therefore, a natural extension of this work would be to investigate in more depth the link between the variation of the drag excess C_g or drag deficiency C_{fs}, and other quantities ($y_s(t)$, $h(t)$, etc.) with respect to the flow parameters (Froude number Fr, release depth d,...) and variables (y_s,$h(t)$,...). Concretely, this would require additional simulations to further explore the parameter space.

7.2.6.2 Wake dynamics above the free surface

At the end of Section 5.2.2.2 it is referred to Appendix G, where it is theoretically shown that the amount of entrained liquid below the cylinder and the characteristic elevation of this entrained liquid above the initial free surface result from a momentum-to-gravity balance. First, the total vertical fluid momentum stored in the wake depends on the history of the cylinder motion occuring below the free surface. It therefore depends on both the release depth and the Reynolds number, while being almost independent of the Froude number. Then, during and after the interface crossing, the gravity starts to have an influence, progressively decreasing the vertical momentum of the entrained liquid, this decrease

becoming gradually stronger as more fluid is going above the initial free surface. As a consequence, the characteristic elevation of the wake has been shown to result from a balance between the stored momentum and the gravity magnitude. This is reflected by an elevation length scale that depends on both the Froude number Fr and the characteristic wake length L_c.

Verifying this scaling is however challenging. On the one hand, the unstable nature of the wake dynamics results in a chaotic behavior of the wake. This means that different simulations of the same problem lead to different wake dynamics above the free surface. As a consequence, any analysis of this problem should involve statistical tools, such as averages over different simulations of the same problem, which makes this investigation very expensive in terms of CPU time. On the other hand, one needs to define a practical way to measure the "characteristic wake elevation", which is not obvious a priori.

Unfortunately, because of these difficulties, the characteristic wake elevation has not been measured in simulations. Moreover, not enough simulations could be performed in order to compute such statistical averages. However, these steps represent a natural possible extension of this work.

7.2.6.3 Dynamics of the film above the body's apex

The film model of Chapter 6 relies on the θ-momentum equation evaluated at the free surface. In contrast, an alternative approach could consist in modelling the dynamics inside the film. While the parameter α of Eq. (6.64) can be closed using the assumed velocity profile for $u_\theta(r)$, the term $\overline{\chi}$ requires the closure of the pressure term, which involves the radial momentum equation. In particular, the pressure terms of order[8] $2n$ in θ in the r-momentum equation translate into terms of order $2n-1$ in the θ-momentum equation. To address this, one can leave the dependency on θ unknown and solve the system of differential equations for the variables t and θ, which requires boundary conditions and some a priori knowledge of the film-pool interactions.

Another possible approach is to close the system of equations by truncating the dependency on θ up to a given order, and balance terms of equal order in θ in both the momentum (i.e., the one remaining after eliminating the pressure) and mass conservation equations. This leads to a total of $2n+2$ equations associated with the couple of unknowns $(\, u_{r,2i}(t), u_{\theta,2i+1}(t)\,)$ for $i = 0, ..., n$. With this approach the difficulty lies in the increasing number of equations up to the order considered and in the closure hypotheses required for the truncation at that order, the former being related to some assumptions on the flow.

Alternatively, in the context of the proposed model that relies on the free-surface momentum equation, a better closure model for the parameter δ could be proposed. This could be done for instance by introducing the assumed velocity profile for the boundary layer into the Navier-Stokes equations and integrating them over the boundary layer thickness. However, unfortunately here again the difficulty stems from closing the pressure term that depends on terms of higher order in θ, as discussed above.

At last, the proposed film model, which does not rely on a thin film approximation, and its general framework could be extended to address other, more complex surface geometries.

7.3 Final words

In this work, the state of the art associated with the PFEM has been pushed forward in two specific directions. First, the α-shape technique has been modified by adding criteria based on node tagging, thus improving mass conservation properties, but without completely removing the fundamental limitation of the method in this regard. Although other efforts have contributed to improving the α-shape technique (for instance Leyssens et al. [133] with Delaunay refinement), alternative approaches to this purely geometric-based α-shape technique, such as for instance the PFEM combined with a level set method [134], could also be considered.

Secondly, novel remeshing techniques have been proposed, which rely on different existing concepts found in the literature, such as size field (or target mesh size), cell-collapse and edge bisection. The novelty of the approach lies in the area-based criterion for node addition and removal, and the related concepts of nodal density and nodal fraction. In particular, these concepts have been used to develop

[8] The radial momentum equation is even in θ by symmetry.

a theoretical extension of the proposed mesh adaptation techniques to three dimensions. Irrespective of the number of space dimensions, the goal was to be able to enforce a distribution of nodes with a density as close as possible to that prescribed by the size field. This can only be achieved by clearly defining the concept of node density. To the author's best knowledge, the question of measuring the local density of nodes in a mathematically rigorous way was yet unexplored, and the definition of the nodal density given in Chapter 3 represents an important contribution in that regard.

In the second part of this work, the understanding of the flow dynamics, and in particular the film dynamics, above a cylinder/sphere exiting a bath has been improved. Specifically, the physics underlying the surge of the free surface and the associated drag exerted on the cylinder/sphere are now better understood. In addition, the film drainage near the cylinder/sphere apex has been described through the use of a new film model. The originality of the model is that it does not rely on a thin film approximation, and therefore offers perspectives for generalization. In particular, one can wonder whether the vanishing contribution of the stress in the momentum equation for the bulk layer also applies to other cases involving large films with internal boundary layers and, if it is the case, under what conditions. The vanishing stress would then provide a tool to compute velocity profiles in the film, which in turn would allow reducing the equations to one single space dimension.

Finally, several avenues have been proposed for further improving the PFEM, and we hope that the method will continue to develop in the near future. On the other hand, the new contributions towards better understanding and modeling the physics of interface crossing open the way to further improvements in the field. For example, the film model and empirical relations proposed in this work could potentially find direct application in coating and related fields or, at least, offer a strong basis for the development of new models. Nonetheless, much remains to do.

Appendix A

Analytical solutions for the solution-based mesh refinement

The solution-based mesh refinement being function of the norm of the velocity gradient matrix, it is shown how it behaves with the spatial coordinates for different analytical solution. The following 2D cases in polar coordinates are considered:

The first case is a general solution such that

$$\mathbf{u} - \mathbf{U}_\infty = (a\cos\theta, b\sin\theta) r^\xi \,, \tag{A.1}$$

a, b and ξ being some parameters. Note the potential flow around a cylinder, of velocity U and radius R, enters in that category with the choice $a = -UR^2$, $b = UR^2$ and $\xi = -2$. The second case is that of a vortex, such that

$$\mathbf{u} - \mathbf{U}_\infty = (0, a\, r^\xi) \,. \tag{A.2}$$

Note that Euler vortex of circulation Γ enters in that category with the choice $a = \frac{\gamma}{2\pi}$ and $\xi = -1$. Let's first look at the first case. By mass conservation, one has that

$$\frac{\partial r u_r}{\partial r} + \frac{\partial u_\theta}{\partial \theta} = 0 \tag{A.3}$$

$$\iff \tag{A.4}$$

$$\frac{\partial (r^{\xi+1})}{\partial r} a\, \cos\theta + \frac{\partial (\sin\theta)}{\partial \theta} b r^\xi = 0 \tag{A.5}$$

$$\iff \tag{A.6}$$

$$(\xi+1)a = -b \,, \tag{A.7}$$

which enables to rewrite b as $-(\xi+1)a$. From this velocity field, the velocity gradient matrix can be computed, and has the following form

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_r}{\partial r} & \frac{1}{r}\frac{\partial u_r}{\partial \theta} - \frac{\partial u_r}{\partial r} \\ \frac{\partial u_\theta}{\partial r} & \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r} \end{pmatrix} \,, \tag{A.8}$$

and whose the norm is

$$||\nabla \mathbf{u}|| = \sqrt{\left(\frac{\partial u_r}{\partial r}\right)^2 + \left(\frac{1}{r}\frac{\partial u_r}{\partial \theta} - \frac{\partial u_r}{\partial r}\right)^2 + \left(\frac{\partial u_\theta}{\partial r}\right)^2 + \left(\frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{u_r}{r}\right)^2} \,. \tag{A.9}$$

In our first case (Eq. (A.1)), one gets

$$\nabla \mathbf{u} = \begin{pmatrix} a\xi\, r^{\xi-1}\cos\theta & a\xi\, r^{\xi-1}\sin\theta \\ -a(\xi+1)\xi\, r^{\xi-1}\sin\theta & -a\xi\, r^{\xi-1}\cos\theta \end{pmatrix} \tag{A.10}$$

and

$$||\nabla \mathbf{u}|| = |a\xi|\, r^{\xi-1}\sqrt{2 + (2\xi+\xi^2)\sin^2\theta} \,. \tag{A.11}$$

On the other hand, for the second case (Eq. (A.2)) one gets

$$\nabla \mathbf{u} = \begin{pmatrix} 0 & -ar^{\xi-1} \\ a\xi r^{\xi-1} & 0 \end{pmatrix} \tag{A.12}$$

and

$$||\nabla \mathbf{u}|| = |a| r^{\xi-1} \sqrt{1+\xi^2}\,. \tag{A.13}$$

This demonstrate that, for these two cases, the norm of the velocity gradient is one order less in term of the radius with respect to the initial velocity field (removing the constant term U_∞ that does not play any role in the computation of the velocity gradient). A good choice of the exponent parameter β of the solution-based imposition of L^* (Eq. (3.23)) can therefore be made by using our knowledge on ξ. If this scaling of the flow is not a priori known, another possibility is to use the rescaled version of the solution-based criterion, which uses the normalized velocity gradient defined by Eq. (3.24). In these two cases, the relative velocity gradient simply scale as r^{-1} such that an appropriate choice of β is 1. Lets quickly demonstrate that the scaling is indeed of order -1 with respect to r:

For the first case (Eq. (A.1)) the norm of the velocity field (in the moving frame of reference to cancel U_∞) is given by

$$||\mathbf{u} - U_\infty|| = \sqrt{a^2 r^{2\xi} \cos^2\theta + (1+\xi)^2 a^2 r^{2\xi} \sin^2} = |a|\, r^\xi \sqrt{1 + (2\xi+\xi^2)\sin^2\theta}\,, \tag{A.14}$$

and the normalized norm of the velocity gradient is then given by

$$\frac{||\nabla \mathbf{u}||}{||\mathbf{u} - U_\infty||} = |\xi|\, r^{-1} \sqrt{\frac{2 + (2\xi+\xi^2)\sin^2\theta}{1 + (2\xi+\xi^2)\sin^2\theta}}\,, \tag{A.15}$$

which is well of order r^{-1} as expected [1]. Now if the second case is taken back (Eq. (A.2)), one simply has

$$\frac{||\nabla \mathbf{u}||}{||\mathbf{u} - U_\infty||} = \frac{|a| r^{\xi-1}\sqrt{1+\xi^2}}{|a| r^\xi} = \sqrt{1+\xi^2}\, r^{-1}\,, \tag{A.16}$$

which is again well of order r^{-1} as expected.

This finishes to demonstrate how relevant is the solution-based criterion for the definition of L^*. Even if particular mass-conservative forms have been taken[2] to justify it, it is reasonable to assume that the norms of the velocity and its gradient around a body are scaling, in a first approximation, as some power of the distance to that body, which is an assumption that helps for choosing good parameters for the solution-based criterion.

[1] Note that the dependency on θ also disappears for $\xi = -2$, which corresponds to the case of a potential flow around a cylinder.

[2] It has not been checked whether or not such a flow satisfies the momentum equation: there is no guarantee that only a pressure field can balance it. However, an arbitrary external force can.

Appendix B

Hessian matrix and higher order derivatives computation with a linear finite element formulation

This section provides the way to compute the Hessian matrix with linear finite element in the context of anisotropic mesh refinement. This is not directly used in the present work but it is an interesting perspective for alternative ways of doing solution-based mesh refinement. Note that this section also describes how higher order derivatives can be approximated.

second order derivative at the element level

As a reminder the Hessian matrix related to one scalar field η (which can be any component of the velocity field or the pressure) is given by (in index notation):

$$H_{i,j} = \frac{\partial^2 \eta}{\partial x_i \, x_j} \, . \tag{B.1}$$

In particular, one needs a discrete expression associated to each element, such that the average of that matrix can be taken at the element level:

$$\overline{\frac{\partial^2 \eta}{\partial x_i \, x_j}} = \overline{H_{i,j}} = \frac{\int_{V_e} H_{i,j} dV}{V_e} \, , \tag{B.2}$$

where V_e is the element volume (area). To compute this quantity, Gauss Theorem can be used in the special case of a scalar ϕ multiplied by the external normal vector, integrated on the surface (in index notations):

$$\int_{S_e} \phi \, n_i dS = \int_{V_e} \frac{\partial \phi}{\partial x_i} dV \, . \tag{B.3}$$

Replacing ϕ by $\frac{\partial \eta}{\partial x_j}$ into this equation yields to:

$$\int_{S_e} \frac{\partial \eta}{\partial x_j} \, n_i dS = \int_{V_e} \frac{\partial^2 \eta}{\partial x_j \partial x_i} dV = \overline{H_{j,i}} V_e \, , \tag{B.4}$$

which gives each component of our Hessian matrix multiplied by the volume V_e. As linear finite elements are used, this should gives $0 = 0$ if only the gradients inside this element are taken into account. In fact these gradients are discontinuous from one element to another, and some averaging between these discontinuous values can be done at the edge level in order to avoid a trivial results. The simplest solution would be to take the arithmetic average of the two adjacent gradients, which should work fine for uniform meshes made of equilateral triangles. However, better approximations should be possible to get considering weighted averages based on the relatives size of adjacent triangles. Note that this is for instance the case for the computation of second derivative at the node level, rather than at the element level.

Second order derivative at the node level

Computing the second order derivatives is in fact what is done when the discrete matrices are built from the finite element discretization. Suppose that a node is surrounded by N_e elements e_I. The second order derivative at this node is given by the following weighted average

$$\widetilde{\frac{\partial^2 \eta}{\partial x_j \partial x_i}} = \frac{\sum_{I=0}^{N} \int_{e_I} \frac{\partial^2 \eta}{\partial x_j \partial x_i} \hat{N}^I(x,y) de_I}{\sum_{I=0}^{N} \int_{e_I} \hat{N}^I(x,y) de_I} = \frac{\sum_{I=0}^{N} \int_{e_I} \frac{\partial^2 \eta}{\partial x_j \partial x_i} \hat{N}^I(x,y) de_I}{\frac{1}{3} \sum_{I=0}^{N} V_{e_I}}, \tag{B.5}$$

where the $\hat{N}^I(x,y)$ are the linear shape functions associated at each adjacent elements that are equal to 1 at the node, and where the "wide tilde" notation has been used to make a difference with the average at the node level from that at the element level (horizontal line). The numerator of this expression is of course formally 0, but using the same trick as before, Gauss theorem can be used to transfer one derivative from η to the shape function. The surface terms coming from this operation should formally not cancel out in between element, as the first order derivative is discontinuous here. However, it is assumed continuous as before, in order to get a non trivial expression of the second derivative. This finally gives

$$\widetilde{\frac{\partial^2 \eta}{\partial x_j \partial x_i}} = -\frac{\sum_{I=0}^{N} \int_{e_I} \frac{\partial \eta}{\partial x_j} \frac{\partial \hat{N}^I(x,y)}{\partial x_i} de_I}{\frac{1}{3} \sum_{I=0}^{N} V_{e_I}}, \tag{B.6}$$

that can be computed since only first order derivatives of the shape functions appear.

Approximation of higher order derivative at the node level

Now, using a similar reasoning, the way to compute estimations of the higher order derivative is shown. Assuming that it is required to compute the third-order derivative of the variable η:

$$\frac{\partial^3 \eta}{\partial x_i \partial x_j \partial x_k}. \tag{B.7}$$

It immediately follows that, using a similar reasoning than above,

$$\widetilde{\frac{\partial^3 \eta}{\partial x_i \partial x_i}} = -\frac{\sum_{I=0}^{N} \int_{e_I} \frac{\partial^2 \eta}{\partial x_i \partial x_j} \frac{\partial \hat{N}^I(x,y)}{\partial x_k} de_I}{\frac{1}{3} \sum_{I=0}^{N} V_{e_I}}, \tag{B.8}$$

where a second order derivative of η appears at the numerator. It can be approximated using the average at the element level (components of the Hessian matrix at the element level),

$$\overline{\frac{\partial^2 \eta}{\partial x_i \, x_j}}, \tag{B.9}$$

which requires quantities of the elements adjacent to these elements adjacent to the node. Finally, one has

$$\widetilde{\frac{\partial^3 \eta}{\partial x_i \partial x_i}} = -\frac{\sum_{I=0}^{N} \int_{e_I} \overline{\frac{\partial^2 \eta}{\partial x_i \partial x_j}} \frac{\partial \hat{N}^I(x,y)}{\partial x_k} de_I}{\frac{1}{3} \sum_{I=0}^{N} V_{e_I}}. \tag{B.10}$$

As one can see, higher order derivatives can be approximated within a linear finite element formulation, by considering quantities associated to element further and further away, as it is the case for all type of space discretisation. However, in our context, because of the mesh non-uniformity, the accuracy on those approximation becomes poorer and poorer, such that in most cases, only the second order derivative at the node level are considered in the discretized equation. Nevertheless, approximation of higher order derivatives could be considered as a criterion to set the target mesh size, estimate computational errors, etc..., because there the accuracy is not so crucial than for solving the numerical equations.

Finally, to come back to our original concern, i.e., the use of the Hessian matrix to define a mesh refinement metric, there is another difficulty, assuming that the Hessian matrix has been correctly computed.

The difficulty is that, since there is one Hessian matrix for each of the variables (components of the velocity and pressure field), a clever way to combine them should be found to get the anisotropic metric that would lead to a local mesh size minimizing the overall local error, but this question would lead too far.

Appendix C

Equivalent system of partial differential equations

We give a more detailed analysis of the influence of the time discretization on numerical errors, compared to that what is done in Section 3.3. The present analysis is based on the Equivalent system of partial differential equations of the semi-discrete backward Euler scheme (continuous in space and discontinuous in time), to evaluate, at the more fundamental level, the impact of this scheme for the Lagrangian form of the Navier-Stokes equations.

The incompressible Navier-Stokes equations are expressed using a Lagrangian formalism, which can be written as follows,

$$\frac{D\mathbf{u}}{Dt} = -\frac{1}{\rho}\nabla p + \nu\Delta\,\mathbf{u} + \mathbf{f} = \mathbf{a}\,, \tag{C.1}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{C.2}$$

$$\frac{D\mathbf{x}}{Dt} = \mathbf{u}\,, \tag{C.3}$$

where $\mathbf{x}, \mathbf{u}, \mathbf{a}$ are respectively the position, the velocity and the acceleration vectors, ρ is the fluid density, ∇p is the pressure gradient, $\Delta\,\mathbf{u}$ is the Laplacian of the velocity field, and $\mathbf{f}$ is the vector of body and surface forces.

The Lagrangian formalism, as a reminder, implies that there is no convective terms, because the time derivatives $\frac{D\cdot}{Dt}$ are material derivatives, which measure the time variation of quantities attached to moving fluid particles. Within this formalism, the position of the moving particles should also be updated which is represented by the last equation.

The equation C.1 is the incompressibility constraint. As it does not include any time derivative, it makes the use of explicit scheme complicated even if some solution exists as the fractional step methods. Therefore, in this thesis, an implicit backward Euler scheme was used, such that the system of PDE from Eqs. (C.2) to (C.3) is time-discretized using the following set of equations[1],

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t} = -\frac{1}{\rho}\left(\nabla p\right)^{n+1} + \nu(\Delta\,\mathbf{u})^{n+1} + \mathbf{f}^{n+1} = \mathbf{a}^{n+1}\,, \tag{C.4}$$

$$\left(\nabla\cdot\mathbf{u}\right)^{n+1} = 0 \tag{C.5}$$

$$\frac{\mathbf{x}^{n+1}-\mathbf{x}^{n}}{\Delta t} = \mathbf{u}^{n+1}\,, \tag{C.6}$$

where Δt is the time step, and the subscripts n and $n+1$ respectively represent the successive times t and $t+\Delta t$. Note that the acceleration as been rewritten in a compact form using $\mathbf{a}$.

To evaluate the error introduced by this time discretization, the equivalent system of partial differential equations must be derived. To do so, each term is approximated using a Taylor expansion around a time in between t and $t+\Delta t$, at a time $t+\epsilon\,\Delta t$, with $0<\epsilon<1$. Note that the subscript used for that time is simply $n+\epsilon$.

[1] In this analysis, only the time discretization is considered, assuming that the space discretization is ideal,i.e., that it does not introduce errors or stability problems. This can be done considering the great stability properties of that time scheme

Equivalent differential equations

The Taylor development of each terms of Eqs. (C.4) to (C.6) are detailed around any time $t + \epsilon\Delta t$, between t and $t + \Delta t$. As one can see, both of the equations to update the velocity and the position of the fluid particles have a form

$$\frac{\boldsymbol{\xi}^{n+1} - \boldsymbol{\xi}^n}{\Delta t} = \left.\frac{D\boldsymbol{\xi}^*}{Dt}\right|^{n+1} , \tag{C.7}$$

where $\boldsymbol{\xi}$ are either the positions or the velocities[2]. Since they have the same form, the development into Taylor series is similar for each case and leads to similar expression of the truncation error.

The mass conservation equation is slightly different. As seen in Section 3.3, with a backward Euler time integration, the variation of the area of an element is larger at the beginning of a time step, and 0 at the end of the time step (see Eq. (3.63)). This result can in fact be formally written using the Equivalent differential equation of the discrete divergence-free equation. However, for the EDE, we want to have an error representative of the overall mass conservation error occurring during a time step. Before going there, a small mathematical artifice is introduced, which is justified at posteriori. It consists in introducing a scalar function ϕ^* whose the particulate time derivative corresponds to the divergence of the velocity field:

$$\frac{D\phi^*}{Dt} = \nabla \cdot \mathbf{u}^* . \tag{C.8}$$

From there, the implicit backward Euler scheme would writes

$$\frac{\phi^{*,n+1} - \phi^{*,n}}{\Delta t} = \left.\frac{D\phi}{Dt}\right|^{n+1} , \tag{C.9}$$

which has the same form than Eq. (C.7) but where the $*$ subscript has been placed on the LHS. Afterwards, all the quantities need to be expressed around $t + \epsilon\Delta t$ using Taylor expansion. One has

$$\boldsymbol{\xi}^{n+1} = \boldsymbol{\xi}^{n+\epsilon} + (1-\epsilon)\Delta t \left.\frac{D\boldsymbol{\xi}}{Dt}\right|^{n+\epsilon} + \frac{(1-\epsilon)^2}{2}\Delta t^2 \left.\frac{D^2\boldsymbol{\xi}}{Dt^2}\right|^{n+\epsilon} + \mathcal{O}(\Delta t^3) , \tag{C.10}$$

$$\boldsymbol{\xi}^{n} = \boldsymbol{\xi}^{n+\epsilon} - \epsilon\Delta t \left.\frac{D\boldsymbol{\xi}}{Dt}\right|^{n+\epsilon} + \frac{(\epsilon)^2}{2}\Delta t^2 \left.\frac{D^2\boldsymbol{\xi}}{Dt^2}\right|^{n+\epsilon} + \mathcal{O}(\Delta t^3) , \tag{C.11}$$

$$\left.\frac{D\boldsymbol{\xi}^*}{Dt}\right|^{n+1} = \left.\frac{D\boldsymbol{\xi}^*}{Dt}\right|^{n+\epsilon} + (1-\epsilon)\Delta t \left.\frac{D^2\boldsymbol{\xi}^*}{Dt^2}\right|^{n+\epsilon} + \mathcal{O}(\Delta t^2) , \tag{C.12}$$

$$\frac{\boldsymbol{\xi}^{n+1} - \boldsymbol{\xi}^n}{\Delta t} = \left.\frac{D\boldsymbol{\xi}}{Dt}\right|^{n+\epsilon} + (\frac{1}{2} - \epsilon)\Delta t \left.\frac{D^2\boldsymbol{\xi}}{Dt^2}\right|^{n+\epsilon} + \mathcal{O}(\Delta t^2) , \tag{C.13}$$

where Eq. (C.13) is obtained by subtracting C.10 from C.11 and dividing by Δt. Finally, replacing Eqs. (C.13) and (C.12) into Eq. (C.7) yields

$$\left.\frac{D\boldsymbol{\xi}}{Dt}\right|^{n+\epsilon} = \left.\frac{D\boldsymbol{\xi}^*}{Dt}\right|^{n+\epsilon} + \frac{1}{2}\Delta t \left.\frac{D^2\boldsymbol{\xi}^*}{Dt^2}\right|^{n+\epsilon} + \mathcal{O}(\Delta t^2) , \tag{C.14}$$

which leads to Eqs. (C.25) and (C.27) respectively for the velocity and the position updates. The mass conservation equation is obtained by using the scalar function ϕ instead of $\boldsymbol{\xi}$ inverting the position of the $*$ symbol:

$$\left.\frac{D\phi^*}{Dt}\right|^{n+\epsilon} = \left.\frac{D\phi}{Dt}\right|^{n+\epsilon} + \frac{1}{2}\Delta t \left.\frac{D^2\phi}{Dt^2}\right|^{n+\epsilon} + \mathcal{O}(\Delta t^2) . \tag{C.15}$$

Furthermore, by noting that $\left.\frac{D\phi^*}{Dt}\right|^{n+\epsilon} = (\nabla \cdot \mathbf{u}^*)|^{n+\epsilon} = 0$ by definition of an incompressible flow, one finally gets

[2] the $*$ subscript on the RHS is used to make the difference between the real values obtained from the scheme from the ideal ones, i.e., the one that would be obtained if no error was introduced during a time step (they differ only by a term proportional to Δt)

$$\frac{D\boldsymbol{\phi}}{Dt} = -\frac{\Delta t}{2}\frac{D^2\boldsymbol{\phi}}{Dt^2} + \mathcal{O}(\Delta t^2)$$
$$\Longleftrightarrow$$
$$\nabla \cdot \mathbf{u} = -\frac{\Delta t}{2}\frac{D(\nabla \cdot \mathbf{u})}{Dt} + \mathcal{O}(\Delta t^2)\,, \tag{C.16}$$

where the $n+\epsilon$ subscript has been removed has the expression is valid for all ϵ (i.e., it does not appear in the leading term of the truncation error). From there, we can already see that the continuity equation is second order accurate in time. Indeed, if one assumes a local solution of the differential of the form [3]

$$\nabla \cdot \mathbf{u}(t) = C^{n+1}(t_{n+1} - t)^\gamma = C^{n+1}(-t)^\gamma\,, \tag{C.17}$$

one solution is given by choosing $\gamma = 2$ at $t = -\Delta t$. On the other, the additional stabilization terms of the PSPG method consist in some projected residual of the momentum conservation equation, which is well known to be only first order accurate in time with a backward Euler time integration scheme. Therefore, this additional term immediately implies that the continuity equation also becomes first order accurate in time, making the stabilization the main responsible for numerical mass conservation errors.

detailed derivation of the equivalent mass conservation equation

The remaining difficulty is to evaluate the term $\frac{D(\nabla\cdot\mathbf{u})}{Dt}$ to relate it with other physical quantities. Unlike the partial derivative, the particulate derivative defined as

$$\frac{D\cdot}{Dt} = \frac{\partial\cdot}{\partial\, t} + u_i\frac{\partial\cdot}{\partial x_i}\,, \tag{C.18}$$

where the index notation with Einstein summation convention is used, does not commute with space derivative (the indices corresponds to the different space directions). For instance, the divergence of the acceleration $\mathbf{a}$ is given by

$$\begin{aligned}
\nabla\cdot\,\mathbf{a} = \nabla\cdot\left(\frac{D\mathbf{u}}{Dt}\right) &= \frac{\partial}{\partial x_j}\left(\frac{\partial u_j}{\partial t} + u_i\frac{\partial u_j}{\partial x_i}\right) \\
&= \left(\frac{\partial^2 u_j}{\partial t\partial x_j} + u_i\frac{\partial^2 u_j}{\partial x_i\partial x_j}\right) + \frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} && \text{(C.19)}\\
&= \left(\frac{\partial\cdot}{\partial\, t} + u_i\frac{\partial\cdot}{\partial x_i}\right)\frac{\partial u_j}{\partial x_j} + \frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} && \text{(C.20)}\\
&= \frac{D}{Dt}(\nabla\cdot\mathbf{u}) + \nabla\mathbf{u} : \nabla\mathbf{u}^T\,, && \text{(C.21)}
\end{aligned}$$

which can be rewritten

$$\frac{D}{Dt}(\nabla\cdot\mathbf{u}) = \nabla\cdot\,\mathbf{a} - \nabla\mathbf{u} : \nabla\mathbf{u}^T\,. \tag{C.22}$$

Finally, this can be inserted in Eq. (C.16) to get Eq. (C.29), rewritten here as

$$\nabla\cdot\mathbf{u} = -\left(\nabla\cdot\,\mathbf{a} - \nabla\mathbf{u} : \nabla\mathbf{u}^T\right)\frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,. \tag{C.23}$$

Remark

For the velocity and the position, since the first order term of the truncation error does not depend on ϵ, it can be removed from Eqs. (C.25) and (C.27) to get Eq. (C.28) and (C.30), as already mentioned in the corresponding section. The goal of the use of the mathematical artifice for the mass conservation

[3] $t_{n+1} = 0$

equation is precisely to remove the dependence of the truncation error on ϵ. Indeed, ones could simply have used a Taylor development for the constraint $(\nabla \cdot \mathbf{u})^{n+1}$ to get

$$(\nabla \cdot \mathbf{u})^{n+1} = (\nabla \cdot \mathbf{u})^{n+\epsilon} + \left((\nabla \cdot \mathbf{a})^{n+\epsilon} - \left(\nabla \mathbf{u} : \nabla \mathbf{u}^T\right)^{n+\epsilon} \right) (1-\epsilon)\Delta t = 0\,, \tag{C.24}$$

which does depend on ϵ. This makes sense because, as the constraint is at $n+1$, the error on mass conservation is smaller when ϵ is close to 1. However, using the ϕ function, the value obtained as a coefficient of Δt was $\frac{1}{2}$ which is the average of all possible value of ϵ between 0 and 1, describing some average error on mass conservation from the time step, which is what is needed to get a correct equivalent differential equation.

Final form of the equivalent system of differential equations

The analysis performed in the previous paragraphs finally leads to the following system of equations,

$$\left.\frac{D\mathbf{u}}{Dt}\right|^{n+\epsilon} = \mathbf{a}^{n+\epsilon} + \left.\frac{D^2\mathbf{u}}{Dt^2}\right|^{n+\epsilon} \frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,, \tag{C.25}$$

$$(\nabla \cdot \mathbf{u})|^{n+\epsilon} = -\left((\nabla \cdot \mathbf{a})^{n+\epsilon} - \left(\nabla \mathbf{u} : \nabla \mathbf{u}^T\right)^{n+\epsilon} \right) \frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,, \tag{C.26}$$

$$\left.\frac{D\mathbf{x}}{Dt}\right|^{n+\epsilon} = \mathbf{u} + \mathbf{a}^{n+\epsilon}\frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,. \tag{C.27}$$

Remarkable enough, the first order term (in Δt) does not depend on ϵ which makes the relation valid for all ϵ such that it can just be dropped from the equations, which gives

$$\frac{D\mathbf{u}}{Dt} = \mathbf{a} + \frac{D^2\mathbf{u}}{Dt^2}\frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,, \tag{C.28}$$

$$\nabla \cdot \mathbf{u} = -\left(\nabla \cdot \mathbf{a} - \nabla \mathbf{u} : \nabla \mathbf{u}^T \right) \frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,, \tag{C.29}$$

$$\frac{D\mathbf{x}}{Dt} = \mathbf{u} + \mathbf{a}\frac{\Delta t}{2} + \mathcal{O}(\Delta t^2)\,. \tag{C.30}$$

From that equivalent system of equations, a few observations can be made. The first equation dictates the variation of the velocity of a fluid particle. The correct solution should be given by the acceleration at that time, i.e., the sum of the forces divided by ρ. Instead, the truncation error term includes the second derivative of the velocity vector. In other words, the error at each time step is proportional the the time variation of the acceleration of that particle. The main famous example of such an error is the braking of a particle in circular motion, which explains why the vortices artificially and slowly decay over time.

The second equation shows that the velocity field is not exactly divergence free. Even if one has the discrete constraint $(\nabla \cdot \mathbf{u})^{n+1} = 0$, this equation reflects the fact that the divergence of the velocity field is not 0 at each time, but proportional[4] to Δt^2, and to the difference between the divergence of the acceleration field and the inner product of the velocity tensor with its own symmetric tensor, which is directly related to the pressure field though the Poisson equation[5]:

$$-\frac{1}{\rho}\boldsymbol{\Delta}\, p + \nabla \cdot \mathbf{f}_{\text{body}} = \nabla \mathbf{u} : \nabla \mathbf{u}^T = \nabla \cdot \mathbf{a} + \mathcal{O}(\Delta t)\,, \tag{C.31}$$

where a potential divergent body force field f_{body} has been included (which is rather rare in practice as body forces mostly consist in gravity, which is divergent-free).

[4] The term into brackets is the divergence of the residual of the momentum conservation equation, which is known to be proportional to Δt, bringing a total dependency on Δt^2

[5] In fact, the term into brackets corresponds to the divergence of the residual of the momentum conservation equation.

Appendix D

Surface tension and convergence of the non linear algorithm

Convergence criterion

In this work, the solution of one implicit time step is obtained by using a Picard algorithm. Let's denote by the subscript $n+1, k$, the k^{th} iteration to find the value of our variables $\mathbf{q}^{n+1}$ (velocity and pressure in our case). The Picard algorithm consists in solving the following system of equations[1] many times to obtain $\mathbf{q}^{n+1,k+1}$ from $\mathbf{q}^{n+1,k}$ until convergence:

$$\mathbf{M}^{n+1,k}(\mathbf{q}^{n+1,k})\mathbf{q}^{n+1,k+1} + \mathbf{K}_{\text{int}}^{n+1,k}(\mathbf{q}^{n+1,k})\mathbf{q}^{n+1,k+1} = \mathbf{f}_{\text{ext}}{}^{n+1,k}(\mathbf{q}^{n+1,k})\,. \tag{D.1}$$

In this equation, $\mathbf{M}$ is the mass matrix, $\mathbf{K}_{\text{int}}$ is the matrix of internal forces (viscous stresses and pressure gradients) and $\mathbf{f}_{\text{ext}}$ is the vector of external forces (body forces, surface traction, surface tension, etc...). It can be noted in a compact form, rewriting the matrix×vector products as force vectors[2]:

$$\mathbf{f}_{\text{ine}}(k, k+1) + \mathbf{f}_{\text{int}}(k, k+1) = \mathbf{f}_{\text{ext}}(k, k+1)\,. \tag{D.2}$$

After each iteration, the residual vector $\mathbf{R}^{k+1}$ is computed as

$$\mathbf{R}^{k+1} = \mathbf{f}_{\text{ine}}(k, k+1) + \mathbf{f}_{\text{int}}(k, k+1) - \mathbf{f}_{\text{ext}}(k, k+1)\,. \tag{D.3}$$

This should be close to a null vector at convergence. From this vector, a scalar value of the residual is computed:

$$\text{res}^{k+1} = \frac{||\mathbf{R}^{k+1}||}{\text{NormFactor}}\,, \tag{D.4}$$

where NormFactor is either a user defined value (a good practice is to use NormFactor $= \rho\, g$ when gravity is involved) or based on a global scaling computed from the forces:

$$\text{NormFactor} = ||f_{\text{ine}}|| + ||f_{\text{int}}|| + ||f_{\text{ext}}||\,. \tag{D.5}$$

From there, the user has to define a tolerance, tol below which the solution is considered to be converged, such that the next time iteration can start. In other words, the nonlinear iterations stop when

$$\text{res}^{k+1} < \text{tol}\,. \tag{D.6}$$

Note that in this work, most simulations have used that criterion with NormFactor $= \rho\, g$ and tol $= 10^{-6}$.

[1] The incompressibility constrains and the stabilization terms are not shown here as there are not involved in the computation of the residual. Note that a similar criterion exists, which uses the residual on the stabilization terms instead of the momentum conservation.

[2] the subscripts have been removed as it involves both contribution at k and $k+1$

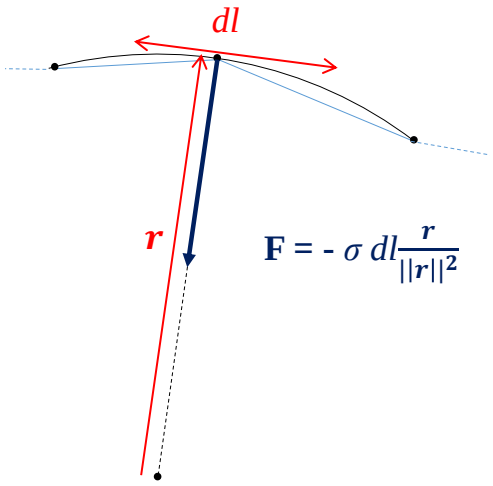


Figure D.1: Illustration of the computation of the surface tension force, which depends on positions of the nodes at $n+1$ and therefore need non linear iteration to converge.

Surface tension

The surface tension developed by Marco Lucio Cerquaglia in his code is computed as illustrated in Fig. D.1. It is based on the radius of curvature of the surface, which is computed as the radius of the circle circumscribed to the triangle made by a node on the free surface and its two neighbours on that free surface. It is given by the following formula,

$$\mathbf{F} = -\sigma \, dl \frac{\mathbf{r}}{||\mathbf{r}||^2} \, , \tag{D.7}$$

where σ is the surface tension at the interface, $\mathbf{r}$ is a vector joining the center of the circle to the node on which the surface, and dl is the sum of the projections of the two half free-surface edges on the direction perpendicular to $\mathbf{r}$, as illustrated in Fig. D.1.

Without this force, the right hand side of Eq. (D.1) would normally remain constant over the non linear iterations, which is why the Picard algorithm is also referred as the fixed-point algorithm. The fact that this is not the case with surface tension renders the convergence less easier. Moreover, this difficulty of convergence is further aggravated by the fact that the surface tension force depends on both the amplitude and the direction of $\mathbf{r}$, which is function of the nodal positions in an highly non linear way.

Therefore, when using surface tension, the PFEM user should either use a smaller time step to limit surface deformation and makes the convergence easier, either use another nonlinear algorithm which is less affected by this issue. For instance, another PhD student working on PFEM, Simon Février, pointed out that this problem was lowered using a Newton-Raphson algorithm. However, in this work, the Picard algorithm was used as it has been shown to be efficient in most applications, while remaining quite simple, provided that an adapted time step is chosen[3].

[3]Note that an adaptive time step routine has been added to Cerquaglia's implementation to reduce the time step when convergence issues where encountered

Appendix E

The Viscous-gravity length scale

Here, the goal is to derive the length scale l_ν at which the gravity and the viscous forces balance, such that the other sources, inertial terms and pressure terms, are only considered in the next order in $h(\theta, t)$. The local Reynolds number Re_ν depends on both that length scale l_ν and a characteristic velocity U_c:

$$Re_\nu = \frac{U_c\, l_\nu}{\nu}\,. \tag{E.1}$$

To get an estimate of that characteristic velocity, the following fact is leveraged: the viscous term, related to the velocity gradients and scaling as $\nu\frac{U_c}{l_\nu}$, should balance with the gravity term scaling as $\mathrm{g}l_\nu$. In other words, the shear force related to the velocity gradient in the film of thickness l_ν should balance the gravity force $\mathrm{g}l_\nu$, which writes

$$\nu\frac{U_c}{l_\nu} = g\, l_\nu\,. \tag{E.2}$$

Isolating U_c finally gives

$$U_c = \frac{g\, l_\nu^2}{\nu}\,. \tag{E.3}$$

This can only make sense if the viscosity effects are important, which can be written as $Re_\nu = 1$, and inserted in Eq. (E.1), together with the expression of U_c, given by Eq. (E.3):

$$1 = \frac{g\, l_\nu^3}{\nu^2}\,. \tag{E.4}$$

Finally, l_ν can be isolated to get

$$l_\nu = \left(\frac{\nu^2}{g}\right)^{\frac{1}{3}}\,. \tag{E.5}$$

Of course, a normalized form can be obtained by dividing both sides of this equation by the radius a, where the Reynolds number $Re_D = 2Re_a = \frac{2\,U\,a}{\nu}$ and the Froude number $Fr = Fr_a = \frac{U^2}{g\,a}$ appear, which gives

$$\frac{l_\nu}{a} = \left(\frac{4Fr}{Re_D^2}\right)^{\frac{1}{3}}\,. \tag{E.6}$$

Note that in the body of the thesis, $\frac{l_\nu}{a}$ was noted as l_ν as each quantities were non-dimensionalized in the section where l_ν has been introduced.

Appendix F

Link between vorticity and skin friction coefficient at the cylinder surface

In this section, the link between the voriticity at the cylinder surface and the skin friction coefficient is derived. The skin friction is defined by Eq. (5.5),

$$c_f = \frac{\mu \frac{\partial u_t}{\partial n}}{\frac{1}{2}\rho U^2}\,, \tag{F.1}$$

which can be non-dimensionalized using $u_t = U\hat{u}_t$ and $n = a\hat{n}$ where $\frac{\partial \hat{u}_t}{\partial \hat{n}}$ is the non-dimensional derivative of the tangential velocity along the normal direction. With a few manipulation, the friction coefficient can be rewritten

$$c_f = \frac{4}{Re_D}\frac{\partial \hat{u}_t}{\partial \hat{n}}\,, \tag{F.2}$$

where the hats are dropped from the notation from this point, remembering that each variable is non-dimensional. To get $\frac{\partial u_t}{\partial n}$ from the velocity gradient matrix $\nabla\,\mathbf{u}$, this last is projected on the normal vector $\mathbf{n}$, the resulting vector being in turn projected on the tangent vector $\mathbf{t}$,

$$\frac{\partial u_t}{\partial n} = \mathbf{t}\cdot(\nabla\,\mathbf{u})\;\mathbf{n}\,, \tag{F.3}$$

where, at a position θ on the cylinder surface, $\mathbf{n}$,$\mathbf{t}$ and $\nabla\,\mathbf{u}$ are given by

$$\mathbf{n} = (\cos\theta, \sin\theta)\,, \tag{F.4}$$

$$\mathbf{t} = (\sin\theta, -\cos\theta)\,, \tag{F.5}$$

$$\nabla\,\mathbf{u} = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} \end{pmatrix}\,. \tag{F.6}$$

This can be inserted into Eq. (F.3) to get (developing the matricial products):

$$\frac{\partial u_t}{\partial n} = \left(\frac{\partial u_x}{\partial x} - \frac{\partial u_y}{\partial y}\right)\sin\theta\cos\theta + \frac{\partial u_x}{\partial y}\sin^2\theta - \frac{\partial u_y}{\partial x}\cos^2\theta\,. \tag{F.7}$$

Then, using the following trigonometric relations,

$$\sin\theta\cos\theta = \frac{\sin 2\theta}{2}\,, \tag{F.8}$$

$$\sin^2\theta = \frac{1-\cos 2\theta}{2}\,, \tag{F.9}$$

$$\cos^2\theta = \frac{1+\cos 2\theta}{2}\,. \tag{F.10}$$

The following can finally be written,

$$\frac{\partial u_t}{\partial n} = \frac{1}{2}\left[\left(\frac{\partial u_x}{\partial x} - \frac{\partial u_y}{\partial y}\right)\sin 2\theta + \left(\frac{\partial u_x}{\partial y} - \frac{\partial u_y}{\partial x}\right) - \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right)\cos 2\theta\right]\,, \tag{F.11}$$

which makes the vorticity $\omega = \frac{\partial u_x}{\partial y} - \frac{\partial u_y}{\partial x}$ appearing. The vorticity is finally related to the skin friction coefficient, replacing $\frac{\partial u_t}{\partial n}$ by its expression in Eq. (F.2):

$$\frac{2}{Re_D}\left[\omega + \sin 2\theta \left(\frac{\partial u_x}{\partial x} - \frac{\partial u_y}{\partial y}\right) - \cos 2\theta \left(\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x}\right)\right] . \tag{F.12}$$

As a conclusion, excepted the pre-factor $\frac{2}{Re_D}$, the skin friction and the vorticity have the same scaling, and qualitative observations and discussions on one of these quantities is generally valid for the other quantity. For example, there are both increasing as the rising cylinder approaches the free surface.

Appendix G

Behavior of the wake during and after the interface crossing

The expected scaling of the wake elevation above the initial free surface is derived theoretically by considering the gravity-to-inertia balance. Because, at moderate and large Reynolds numbers, the wake shows many random features, a numerical analysis would require a significant number of simulations. Wake properties are thus investigated numerically only at very small Reynolds numbers, for which the wake remains symmetric, in Section 5.2.4. As it has been shown, before the interface crossing the state of the wake does not depend on the free-surface dynamics, and the separation region is well defined. Therefore, the characteristic thickness of the wake can be approximated as the cylinder radius a, while its characteristic length L_c depends on the Reynolds number and the release depth (for instance, the separation bubble length is a good choice). The total momentum in the wake just before the interface crossing (defined as $t^* = 0$), p_w, should thus be approximately proportional to the density ρ, size of the wake aL_c and cylinder velocity U,

$$p_w(t^* = 0) = \lambda_1 \, \rho \, a \, L_c \, U \, , \tag{G.1}$$

where λ_1 is some constant. Neglecting any viscous effect and assuming that gravity is balanced by buoyancy for the immersed part of the wake ($y < 0$) but fully acts on its emerged portion ($y > 0$), as schematically shown in Fig. G.1, the change of momentum in the wake can be expressed as

$$\frac{dp_w}{dt^*} = -\lambda_2 \, \rho \, g \, a \, U \, t^* \, , \tag{G.2}$$

where λ_2 is another constant. Using the initial condition in Eq. (G.1), we have

$$p_w(t^*) = \lambda_1 \, \rho \, a \, L_c \, U - \frac{1}{2} \lambda_2 \, \rho \, g \, a \, U \, t^{*2} \, . \tag{G.3}$$

The maximum elevation of the wake's center of gravity would approximately be reached when the wake momentum vanishes. This occurs at a characteristic time

$$t_c^* = \sqrt{2 \frac{\lambda_1}{\lambda_2} \frac{L_c}{g}} = k \sqrt{\frac{L_c}{g}} \, , \tag{G.4}$$

where k is a constant encompassing the constants λ_1 and λ_2. The non-dimensional characteristic elevation of the wake's center of gravity is then given by

$$k' \frac{U \, t_c^*}{a} = k'' \sqrt{\frac{U^2 \, L_c}{g \, a^2}} = k'' \sqrt{Fr \, \hat{L}_c} \, , \tag{G.5}$$

with $k' < 1$ to take into account the progressive deceleration of the wake, $k'' = k' \, k$, and $\hat{L}_c = \frac{L_c}{a}$. This suggests that the larger the Froude number and the longer the wake, the higher the wake is entrained, as one can logically expect.

The present analysis is very simplified and provides only a rough scaling. To get a better idea about how the fluid is entrained, other aspects should be investigated, such as the wake-tank interactions, the effect of vortices, but also the lowering of the free-surface level, etc...

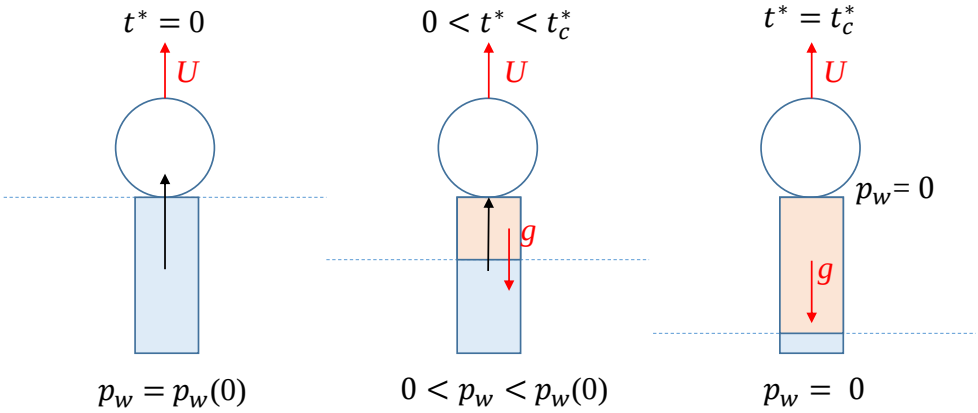


Figure G.1: Simple model illustrating the effect of gravity on the entrained wake (schematically represented by the rectangle below the cylinder). Left: the lower part of the cylinder (i.e., the top part of the wake) reaches the initial free-surface level ($t^* = 0$). Middle: the wake is subjected to a resultant gravity force proportional to the mass of the emerged part of the wake (light orange); gravity is balanced by buoyancy in its immersed part (light blue). Right: the wake has reached its maximum elevation ($t^* = t_c^*$) as its vertical momentum p_w vanishes.

Appendix H

Velocity profile close to the boundary layer

The shape of the velocity profile above the cylinder/sphere near the free-surface suggests a local modification of the form

$$u_{\theta,1}(r) = u_{w,1}\left(\frac{1}{r} + \delta r + c_3(t)\left(r - (1 + h - \delta_{FS})\right)^2\right) \; \forall r > 1 + h - \delta_{FS}\,, \tag{H.1}$$

where δ_{FS} is some free-surface layer thickness, which presumably scales as the boundary layer thickness δ. As the film thins, this local profile goes down such that the profile shape is fundamentally, but locally unsteady. Considering this aspect of the flow would have required much efforts, however, the value of $c_3(t)$ can be correlated with that of δ_{FS} considering the no-shear condition at the free-surface. Indeed, according to Eq. (6.75) one has

$$\left.\frac{\partial u_{\theta,1}}{\partial r}\right|_{r=1+h_0} \simeq \left.\frac{u_{\theta,1}}{r}\right|_{r=1+h_0} = \frac{u_{\theta,1}|_{r=1+h_0}}{1+h_0}\,. \tag{H.2}$$

Evaluating both sides using Eq. (H.1) yields

$$u_{w,1}\left(\frac{-1}{(1+h_0)^2} + \delta + 2c_3\delta_{FS}\right) = \frac{u_{w,1}}{1+h_0}\left(\frac{1}{1+h_0} + \delta(1+h_0) + c_3\delta_F^2\right)\,, \tag{H.3}$$

which implies

$$c_3(t) = \frac{2}{(1+h_0)\delta_{FS}\left(2(1+h_0) - \delta_{FS}\right)} \simeq \frac{1}{(1+h_0)^2\delta_{FS}} \tag{H.4}$$

where the last equality is obtained by making the approximation $\delta_{FS} \ll 1$. For Fig. 6.7, a value $\delta_{FS} = 0.1$ has been used, which results in a value of $c_3 \simeq 3.02$. To account for this new velocity profile in the model, additional modifications should be performed. Not only the relation between $\overline{u_{\theta,1}}$ and $u_{t,1}$ is affected but also the free-surface momentum equation. Indeed, neglecting viscous terms is no more valid with this new approach, since the gradient tensor does not vanish anymore due to the different velocity profile. Note that it may be neglected as well but should clearly be justified. Anyway, this is left for future work. Finally, the reader may agree that, even if these extensions have not been considered in the present model, some insights on how to solve the problem have been given by providing the scaling of $c_3(\delta_{FS}, t)$, and that good approximation for δ_{FS} should not be too hard to obtain.

Bibliography

1. Cerquaglia, M. L. *Development of a fully-partitioned PFEM-FEM approach for fluid-structure interaction problems characterized by free surfaces, large solid deformations, and strong added-mass effects* PhD thesis (University of Liège, 2019).
2. Chang, B. *et al.* Jumping dynamics of aquatic animals. *Journal of the Royal Society Interface* (**16**.152) (2019).
3. Craik, A. D. The origins of water wave theory. *Annual Review of Fluid Mechanics* (**36**), 1–28 (2004).
4. Méhauté, B. *An Introduction to Hydrodynamics and Water Waves* (Springer Berlin, Heidelberg, 1976).
5. Osborne, A. R. Brief history and overview of nonlinear water waves. *International Geophysical series* (**97**), 3–31 (2010).
6. Yang, J. & Balaras, E. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *Journal of Computational Physics* (**215**.1), 12–40 (2006).
7. Hirt, C. & Nichols, B. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of computational physics.* (**39**.1) (1981).
8. Bures, L., Sato, Y. & Pautz, A. Piecewise linear interface-capturing volume-of-fluid method in axisymmetric cylindrical coordinates. *Journal of Computational Physics* (**436**), 110291 (2021).
9. Chakraborty, I., Biswas, G. & Ghoshdastidar, P. S. A coupled level-set and volume-of-fluid method for the buoyant rise of gas bubbles in liquids. *International Journal of Heat and Mass Transfer* (**58**.1-2), 240–259 (2013).
10. Mittal, R. & Iaccarino, G. Immersed boundary methods. *Annual Review of Fluid Mechanics* (**37**), 239–261 (2005).
11. Steger, J. L. & Benek, J. A. On the use of composite grid schemes in computational aerodynamics. *Computer methods in applied mechanics and engineering* (**64**.1), 301–320 (1987).
12. Donéa, J., Huerta, A., Ponthot, J.-P. & Rodrïguez-Ferran, A. *Arbitrary Lagrangian-Eulerian Methods* 2004.
13. Boman, R. & Ponthot, J. P. Efficient ALE mesh management for 3D quasi-Eulerian problems. *International Journal for Numerical Methods in Engineering* (**92**.10), 857–890 (2012).
14. Moes, N., Remacle, J.-F., Lambrechts, J., Le, B. & Chevaugeon, N. The eXtreme Mesh deformation approach (X-MESH) for the Stefan phase change model. *Journal of Computational Physics* (**477**), 111878 (2023).
15. Gingold, R. & Monaghan, J. Smoothed particle hydrodynamics, theory and application to non-spherical stars. *Royal Astronomy Society* (**181**.3), 375–389 (1977).
16. Lucy, L. A numerical approach to the testing of the fission hypothesis. *The astronomical journal* (**82**), 1013–1024 (1977).
17. Vacondio, R., Altomare, C., De Leffe, M., *et al.* Grand challenges for Smoothed Particle Hydrodynamics numerical schemes. *Computational Particle Mechanics* (**8**), 575–588 (2020).

18. Barcarolo, D., Le Touzé, D., Oger, G. & De Vuyst, F. Adaptive particle refinement and derefinement applied to the smoothed particle hydrodynamics method. *Journal of Computational Physics* (**273**), 640–657 (2014).

19. López, Y. R., Roose, D. & Morfa, C. R. Dynamic particle refinement in SPH: Application to free surface flow and non-cohesive soil simulations. *Computational Mechanics* (**51**) (2013).

20. Vacondio, R., Rogers, B., Stansby, P., Mignosa, P. & Feldman, J. Variable resolution for SPH: A dynamic particle coalescing and splitting scheme. *Computer Methods in Applied Mechanics and Engineering* (**256**), 132–148 (2013).

21. Vacondio, R., Rogers, B., Stansby, P. & Mignosa, P. Variable resolution for SPH in three dimensions: Towards optimal splitting and coalescing for dynamic adaptivity. *Computer Methods in Applied Mechanics and Engineering* (**300**), 442–460 (2016).

22. Cremonesi, M., Franci, A. & Idelsohn, S. A State of the Art Review of the Particle Finite Element Method (PFEM). *Archives of Computational Methods in Engineering* (**27**), 1709–1735 (2020).

23. Oñate, E. & Idelsohn, S. R. The Particle Finite Element Method - An overview. *International Journal of Computational Methods* (**1**.2), 267–307 (2004).

24. Cerquaglia, M. L., Deliége, G., Boman, R., Terrapon, V. & Ponthot, J.-P. Free-slip boundary conditions for simulating free-surface incompressible flows through the particle finite element method. *International Journal for Numerical Methods in Engineering* (**110**.10), 921–946 (2017).

25. Edelsbrunner, H. & Mücke, E. P. Three-Dimensional Alpha Shapes. *ACM Transactions on Graphics* (**13**.1), 43–72 (1994).

26. Fortin, M. An analysis of the convergence of mixed finite element methods. *RAIRO analyse numérique* (**11**.4), 341–354 (1977).

27. Donéa, J. & Huerta, A. *Finite Element Methods for Flow Problems* (John Wiley & Sons, 2003).

28. Leborgne, G. An optimally consistent stabilization of the inf-sup condition. *Numerische Mathematik* (**91**.1), 35–56 (2002).

29. Brezzi, F. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. en. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique* (**8**.2), 129–151 (1974).

30. Donéa, J., Giuliani, S., Morgan, K. & Quartapelle, L. The significance of chequerboarding in a Galerkin finite element solution of the Navier-Stokes equations. *International Journal for Numerical Methods in Engineering* (**17**), 790–795 (1981).

31. Sani, R. L., Gresho, P. M., Lee, R. L. & Griffiths, D. F. The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier-Stokes equations: Part 1. *International Journal for Numerical Methods in Fluids* (**1**.1), 171–204 (1981).

32. Sani, R. L., Gresho, P. M., Lee, R. L. & Griffiths, D. F. The cause and cure (!) of the spurious pressures generated by certain FEM solutions of the incompressible Navier-Stokes equations: Part 2. *International Journal for Numerical Methods in Fluids* (**1**.2), 171–204 (1981).

33. Hughes, T. J. R., Franca, L. P. & Balestra, M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics* (**54**.3), 85–99 (1986).

34. Oñate, E. Finite increment calculus (FIC): a framework for deriving enhanced computational methods in mechanics. *Advanced Modeling and Simulation in Engineering Sciences* (**3**.1) (2016).

35. Idelsohn, S. R. & Oñate, E. The challenge of mass conservation in the solution of free-surface flows with the fractional-step method: Problems and solutions. *International Journal for Numerical Methods in Biomedical Engineering* (**26**.10), 1313–1330 (2010).

36. Tezduyar, S., Mittal, S., Ray, E. & Shih, R. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity pressure elements. *Computer Methods in Applied Mechanics and Engineering* (**95**.2), 221–242 (1992).

37. Christon, M. A., Martinez, M. J. & Voth, T. E. Generalized Fourier analyses of the advection-diffusion equation - Part I: One-dimensional domains. *International Journal for Numerical Methods in Fluids* (**45**.8), 839–887 (2004).

38. Voth, T. E., Martinez, M. J. & Christon, M. A. Generalized fourier analyses of the advection-diffusion equation - Part II: Two-dimensional domains. *International Journal for Numerical Methods in Fluids* (**45**.8), 889–920 (2004).

39. Edelsbrunner, H., Kirkpatrick, D. G. & Seidel, R. On the Shape of a Set of Points in the Plane. *IEEE Transactions on information theory* (**29**.4), 551–559 (1983).

40. Cremonesi, M., Frangi, A. & Perego, U. A Lagrangian finite element approach for the analysis of fuid–structure interaction problems. *International Journal of Numerical Methods in Engineering* (**184**.5), 610–630 (2010).

41. Bernadini, F. & Bajaj, C. L. *Sampling and Reconstructing Manifolds Using Alpha-Shapes* tech. rep. 97-013 (Department of computer science technical reports, 1997).

42. Park, S., Lee, S. & Kim, J. *A Surface Reconstruction Algorithm Using Weighted Alpha Shapes* in *Fuzzy Systems and Knowledge Discovery, Second International Conference, FSKD* (2005), 1141–1150.

43. Edelsbrunner, H. & Shah, N. Incremental Topological Flipping Works for Regular Triangulations. *Algorithmica* (**15**)**,** 223–241 (1996).

44. Franci, A. & Cremonesi, M. On the effect of standard PFEM remeshing on volume conservation in free-surface fluid flow problems. *Computational Particle Mechanics* (**4**.3), 331–343 (2017).

45. Berger, M. & Colella, P. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics* (**82**)**,** 64–84 (1989).

46. Friedel, H., Grauer, R. & Marliani, C. Adaptive Mesh Refinement for Singular Current Sheets in Incompressible Magnetohydrodynamic Flows. *Journal of Computational Physics* (**134**.1), 190–198 (1997).

47. Bell, J., Berger, M., Saltzman, J. & Welcome, M. Three-Dimensional Adaptive Mesh Refinement for Hyperbolic Conservation Laws. *SIAM Journal on Scientific Computing* (**15**)**,** 127–138 (1994).

48. Vanella, M., Rabenold, P. & Balaras, E. A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid-structure interaction problems. *Journal of Computational Physics* (**229**.18), 6427–6449 (2010).

49. Bansch, E. An adaptive finite-element strategy for the three-dimensional time-dependent Navier-Stokes equations. *Journal of Computational and Applied Mathematics* (**36**)**,** 3–28 (1991).

50. De Sterck, H. *et al.* Efficiency-based h- and hp-refinement strategies for finite element methods. *Numerical Linear Algebra with Applications* (**15**.2-3), 1–25 (2008).

51. Duval, M., Lozinskic, A., Passieux, J. & Salaunb, M. Residual error based adaptive mesh refinement with the non-intrusive patch algorithm. *Computer Methods in Applied Mechanics and Engineering* (**329**)**,** 118–143 (2018).

52. Bristeau, M., Glowinski, R. & Periaux, J. Numerical methods for the Navier-Stokes equations. Applications to the simulation of compressible and incompressible viscous flows. *Computer Physics Reports* (**6**)**,** 73–187 (1987).

53. Bathe, K. & Zhang, H. A mesh adaptivity procedure for CFD and fluid-structure interactions. *Computer and Structures* (**87**.11-12), 604–617 (2009).

54. Lohner, R. & Morgan, K. *Improved adaptive refinement strategies for finite element aerodynamic computations* in *24th Aerospace Sciences Meeting* (1986).

55. Dannenhofer, F. & Baron, J. *Grid adaptation for the 2D Euler equations* in *23rd Aerospace Sciences Meeting* (1985).

56. Frey, P. J. & Alauzet, F. Anisotropic mesh adaptation for CFD computations. *Computer Methods in Applied Mechanics and Engineering* (**194**.48-49), 5068–5082 (2005).

57. Castro Díaz, M. J., Hecht, F., Mohammadi, B. & Pironneau, O. Anisotropic Unstructured Mesh Adaptation For Flow Simulations. *International Journal For Numerical Methods In Fluids* (**25**), 475–491 (1997).

58. Pino, S. Metric-based mesh adaptation for 2D Lagrangian compressible flows. *Journal of Computational Physics* (**230**.5), 1793–1821 (2011).

59. Delaunay, B. Sur la sphère vide, à la mémoire de Georges Voronoï. *Bulletin de l'Académie des sciences de l'URSS. Classe des sciences mathématiques et na* (**6**), 793–800 (1934).

60. Lee, D. & Schachter, B. Two Algorithms for Constructing a Delaunay Triangulation. *International Journal of Computer and Information Sciences* (**9**), 219–242 (1980).

61. Rajan, V. T. Optimality of the Delaunay Triangulation in $\mathbb{R}^d$. *Discrete Comput. Geom.* (**12**), 189–202 (1994).

62. Liu, Y., Pan, H., Snyder, J., Wang, W. & Guo, B. Computing self-supporting surfaces by regular triangulation. *ACM Transactions on Graphics* (**32**.4) (2013).

63. Falla, R., Bobach, B.-J., Boman, R., Ponthot, J.-P. & Terrapon, V. E. Mesh adaption for two-dimensional bounded and free-surface flows with the particle finite element method. *Computational Particle Mechanics* (**10**.5) (2023).

64. Hannukainen, A., Korotov, S. & Křížek, M. On global and local mesh refinements by a generalized conforming bisection algorithm. *Journal of Computational and Applied Mathematics* (**235**.2), 419–436 (2010).

65. Cerquaglia, M. L. *et al. On some drawbacks and possible improvements of a Lagrangian finite element approach for simulating incompressible flows* in *IV International Conference on Particle-based Methods-Fundamentals and Applications* (2015).

66. Thomas, P. D. & Lombard, C. K. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal* (**17**.10), 1030–1037.

67. Guillard, H. & Farhat, C. On the significance of the geometric conservation law for flow computations on moving meshes. *Comput. Methods Appl. Mech. Engrg.* (**190**), 1467–1482 (2000).

68. West, D. B. *et al. Introduction to graph theory* (Prentice hall Upper Saddle River, 1996).

69. Gram, J. P. Om rumvinklerne i et polyeder. *Tidsskrift for mathematik* (**4**), 161–163 (1874).

70. Sommerville, D. M. Y. & MacMahon, P. A. The relations connecting the angle-sums and volume of a polytope in space of n dimensions. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* (**115**.770), 103–119 (1927).

71. Perles, M. A. & Shephard, G. C. Angle sums of convex polytopes. *Mathematica Scandinavica* (**21**.2), 199–218 (1967).

72. Gardner, R. & Verghese, K. On the solid angle subtended by a circular disc. *Nuclear Instruments and Methods* (**93**.1), 163–167 (1971).

73. Meduri, S., Cremonesi, M. & Perego, U. An efficient runtime mesh smoothing technique for 3D explicit Lagrangian free-surface fluid flow simulations. *International Journal for Numerical Methods in Engineering* (**117**.4), 430–452 (2018).

74. Tournois, J., Srinivasan, R. & Alliez, P. *Perturbing slivers in 3D delaunay meshes* in *18th International Meshing Roundtable, IMR* (2009), 157–173.

75. Cerquaglia, M. L. *Development of a fully-partitioned PFEM-FEM approach for fluid-structure interaction problems characterized by free surfaces, large solid deformations, and strong added-mass effects* PhD thesis (university of Liege, 2019).

76. Henderson, R. D. Details of the drag curve near the onset of vortex shedding. *Physics of Fluids* (**7**.9), 2102–2104 (1995).

77. Sheard, G., Hourigan, K. & Thompson, M. Computations of the drag coefficients for low-Reynolds-number flow past rings. *Journal of Fluid Mechanics* (**526**), 257–275 (2005).

78. Wieselberger, C. Neuere Feststellungen über die Gestze des Flussigkeits - und Luftwiderstands. *Phys. Z.* (**22**.11), 321–328 (1921).

79. Norberg, C. Flow around a circular cylinder: aspects of fluctuating lift. *Journal of Fluids and Structures* (**15**), 459–469 (2001).

80. Henderson, R. D. *Adaptive Spectral Element Methods, parallel algorithms and simulations* PhD thesis (Princeton University, 1994).

81. Williamson, C. H. K. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics* (**28**.1), 477–539 (1996).

82. Zhang, H.-Q., Fey, U. & Noack, B. R. On the transition of the cylinder wake. *Physics of Fluids* (**7**.4), 779–794 (1995).

83. Taneda, S. Experimental investigation of the wake behind Cylinders and Plates at Low Reynolds Number. *Journal of the Physical Society of Japan* (**11**.3), 302–307 (1956).

84. Grove, A. S., Shair, F. H. & Peterson, E. An experimental investigation of the steady separated flow past a circular cylinder. *Journal of Fluid Mechanics* (**19**.1), 60–80 (1964).

85. Coutanceau, M. & Bouard, R. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. *Journal of Fluid Mechanics* (**79**.2), 231–256 (1977).

86. Acrivos, A., Leal, L. G., Snowden, D. D. & Pan, F. Further experiments on steady separated flows past bluff objects. *Journal of Fluid Mechanics* (**34**.1), 25–48 (1968).

87. Sen, S., Mittal, S. & Biswas, G. Steady separated flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics* (**620**), 89–119 (2009).

88. Keller, H. B. & Takami, H. in *Numerical Solutions of Nonlinear Differential Equations* 115 (1966).

89. Henderson, R. D. Non-linear dynamics and pattern formation in turbulent wake transition. *Journal of Fluid Mechanics* (**352**), 65–112 (1997).

90. Mittal, S. & Kumar, B. Flow past a rotating cylinder. *Journal of Fluid Mechanics* (**476**), 303–334 (2003).

91. Wang, S., Hu, J., Huang, C. & Yu, Y. Graphics processing unit-accelerated smoothed particle hydrodynamics—Finite difference method and the application for the flow around a cylinder with forced motions. *Physics of Fluids* (**33**.12), 127122 (2021).

92. Kumar, S., Navrose & Mittal, S. Lock-in in forced vibration of a circular cylinder. *Physics of Fluids* (**28**.11), 113605 (2016).

93. Koumoutsakos, P. & Leonard, A. High-Resolution simulations of the flow around an impulsively started cylinder using vortex methods. *Journal of Fluid Mechanics* (**296**), 1–38 (1995).

94. Ghia, U., Ghia, K. & Shin, C. T. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics* (**48**), 387–411 (1982).

95. Hou, S., Zou, Q., Chen, S., Doolen, G. & Cogley, A. Simulation of cavity flow by the lattice Boltzmann method. *Journal of Computational Physics* (**118**), 329–347 (1995).

96. Schreiber, R. & Keller, H. B. Driven Cavity Flows by Efficient Numerical Techniques. *Journal of Computational Physics* (**49**.2), 310–333 (1983).

97. Vanka, S. Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *Journal of Computational Physics* (**65**), 138–158 (1986).

98. Wright, N. & Gaskell, P. An Efficient Multigrid Approach To Solving Highly Recirculating Flows. *Computers and Fluids* (**24**.1), 63–79 (1995).

99. Delorme, L., Colagrossi, A., Souto-Iglesias, A., *et al.* A set of canonical problems in sloshing, Part 1: Pressure field in forced roll-comparison between experimental results and SPH. *Ocean Engineering* (**36**.2), 168–178 (2009).

100. Souto-Iglesias, A., Bulian, G. & Botia-Vera, E. A set of canonical problems in sloshing. Part 2: Influence of tank width on impact pressure statistics in regular forced angular motion. *Ocean Engineering* (**38**.16), 1823–1830 (2015).

101. Havelock, T. H. The method of images in some problems of surface waves. *Proceedings of the Royal Society of London, series A* (**115**.771), 268–280 (1927).

102. Havelock, T. H. The Vertical Force on a Cylinder Submerged in a Uniform Stream. *Proceedings of the Royal Society of London, series A* (**122**.790), 387–393 (1929).

103. Havelock, T. H. The Forces on a Circular Cylinder Submerged in a Uniform Stream. *Proceedings of the Royal Society of London, series A* (**157**.892), 526–534 (1936).

104. Dean, W. R. On the reflexion of surface waves by a submerged circular cylinder. *Mathematical Proceedings of the Cambridge Philosophical Society* (**44**.4), 483–491 (1948).

105. Ursell, F. Surface waves on deep water in the presence of a submerged circular cylinder. I. *Mathematical Proceedings of the Cambridge Philosophical Society* (**46**.1), 141–152 (1950).

106. Havelock, T. H. The wave resistance of a cylinder started from rest. *Quarterly journal of mechanics and applied mathematics* (**2**.3), 325–334 (1949).

107. Havelock, T. H. The resistance of a submerged cylinder in accelerated motion. *Quarterly journal of mechanics and applied mathematics* (**2**.4), 419–427 (1949).

108. Tuck, E. 0. The effect of non-linearity at the free surface on flow past a submerged cylinder. *J. Fluid Meeh* (**22**.2), 401–414 (1965).

109. Miloh, T. Free-surface flow due to impulsive motion of a submerged circular cylinder. *Journal of Fluid Mechanics* (**286**)**,** 67–101 (1995).

110. Miloh, T. Free-surface flow generated by a small submerged circular cylinder starting from rest. *Journal of Fluid Mechanics* (**286**)**,** 103–116 (1995).

111. Tesle, J. G. Inviscid flow about a cylinder rising to a free surface. *Journal of Fluid Mechanics* (**182**)**,** 149–168 (1987).

112. Greenhow, M. Water entry and exit of horizontal circular cylinders. *Philosophical Transactions of the Royal Society* (**355**)**,** 551–563 (1997).

113. Greenhow, M. & Lin, W.-M. *Nonlinear free surface effects: experiments and theory* tech. rep. (Massachusetts institute of technology, 1983).

114. Horowitz, M. & Williamson, C. H. Dynamics of a rising and falling cylinder. *Journal of Fluids and Structures* (**22**.6-7), 837–843 (2006).

115. Horowitz, M. & Williamson, C. H. Vortex-induced vibration of a rising and falling cylinder. *Journal of Fluid Mechanics* (**662**)**,** 352–383 (2010).

116. Truscott, T. T., Epps, B. P. & Techet, A. H. Unsteady forces on spheres during free-surface water entry. *Journal of Fluid Mechanics* (**704**)**,** 173–210 (2012).

117. Truscott, T. T., Epps, B. P. & Munns, R. H. Water exit dynamics of buoyant spheres. *Phys. Rev. Fluids* (**1**.7), 074501 (2016).

118. Nair, V. V. & Bhattacharyya, S. K. Water entry and exit of axisymmetric bodies by CFD approach. *Journal of Ocean Engineering and Science* (**3**.2), 156–174 (2018).

119. Kleefsman, K. & Veldman, A. E. An Improved Volume-of-Fluid (iVOF) Method for Wave Impact Type Problems. *In EPRINTS-BOOK-TITLE University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science* () (2004).

120. Ni, B. Y., Zhang, A. M. & Wu, G. X. Simulation of complete water exit of a fully-submerged body. *Journal of Fluids and Structures* (**58**)**,** 79–98 (2015).

121. Hao, H., Yu, J., Song, Y., Chen, F. & Liu, T. Deformation and regimes of liquid column during water exit of a partially submerged sphere using the front-tracking lattice Boltzmann method. *Journal of Fluids and Structures* (**99**) (2020).

122. Wu, Q. & Ni, B. Experimental study on large deformation of free surface during water exit of a sphere. *Ocean Engineering* (**140**), 369–376 (2017).

123. Vincent, L. Cylinder exit dynamics: inertial exit, and the effect of finite wake size. *Journal of fluid dynamics* () (Submitted).

124. Liju, P.-Y., Machane, R. & Cartellier, A. Surge effect during the water exit of an axisymmetric body traveling normal to a plane interface: experiments and BEM simulation. *Experiments in Fluids* (**31**), 241–248 (2001).

125. Dorbolo, S. Experimental investigation of exit dynamics of a circular cylinder from the water. *Journal of fluid dynamics* () (In prep.).

126. Greenspan, H. P. On the Motion of a small viscous droplet that wets a surface. *Journal of fluid mechanics* (**84**), 125–143 (1978).

127. Reynolds, O. On the Theory of Lubrication and its Application to Mr. Beauchamp Tower's Experiments, including an experimental determination of the viscosity of olive oil. *Philosophical Transactions of the Royal Society of London* (**177**), 157–235 (1886).

128. Ruyer-Quil, C., Kofman, N., Chasseur, D. & Mergui, S. Dynamics of falling liquid films. *European Physical Journal E* (**37**.4) (2014).

129. Bouard, R. & Coutanceau, M. The early stage of development of the wake behind an impulsively started cylinder for $40 < \mathrm{Re} < 10000$. *Journal of fluid mechanics* (**101**), 583–607 (1980).

130. Von Kármán, T. On Laminar and Turbulent Friction. *Journal of Applied Mathematics and Mechanics* (**1**.4), 1467–1482 (1921).

131. Chang, H.-C. Solitary Wave Formation and Dynamics on Falling Films. *Advances in Applied Mechanics* (**32**).

132. Craster, R. V. & Matar, O. K. Dynamics and stability of thin liquid films. *Reviews of Modern Physics* (**81**.3), 1131–1198 (2009).

133. Leyssens, T. & Remacle, J.-F. *A Mesh Adaptation algorithm for highly deforming domains in the Particle Finite Element Method* in *XI International Conference on Adaptive Modeling and Simulation (ADMOS)* (2023).

134. Fernandez Sanchez, E. *et al.* A particle finite element method based on Level–Set functions. *Journal of Computational Physics* (**487**), 112187 (2023).

135. Chew, L. P. *Guaranteed-quality mesh generation for curved surfaces* in *In Proceedings of the ninth annual symposium on Computational geometry* (1993), 274–280.

136. Rizzieri, G., Ferrara, L. & Cremonesi, M. Simulation of viscoelastic free-surface flows with the Particle Finite Element Method. *Computational Particle Mechanics* () (2024).