An immersed computational framework for multiphase fluid-structure interaction.

Yang, Liang

Prifysgol Abertawe
Swansea University

An Immersed Computational Framework for Multiphase Fluid-Structure Interaction

by

Liang Yang

Submitted to the College of Engineering
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

at

Swansea University

March 31, 2015

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

Acknowledgements

I would like to express my special appreciation and thanks to my main supervisors, Dr. Antonio J. Gil and Prof. J. Bonet. Both your advice on research as well as on my career prospects have been priceless. Thank you for encouraging and supporting my research initiative by allowing me to grow as a researcher in the field of computational mechanics. I would also like to thank my group member, Dr. A. Arranz Carreño. It has been great pleasure working with you. I also want to thank all my friends in the Zienkiewicz Centre for Computational Engineering for accompanying me during my PhD studies. I gratefully acknowledge the financial support received from the Zienkiewicz Scholarship Programme, which allowed me to concentrate on this research. A special thanks also go to my family, my parents (Xiaoping Li and Pingjian Yang), for giving me birth and making me feel blessed and supported throughout my academic career. Without their encouragement, it would have been extremely difficult for me to carry out this work. This thesis is completely dedicated to them.

Abstract

In the field of computational modelling of fluid-structure interaction problems, two main families of methodologies have been used in practice: body fitted (boundary fitted) approaches [1,2] and immersed type methods [3]. Methods within each family have both strengths and weaknesses. Within the body fitted methodologies, the main disadvantage is the computational cost related to the underlying fluid mesh update or the need to use expensive re-meshing algorithms, a factor particularly important in the case of three-dimensional simulations.

As an alternative to boundary fitted methods, the Immersed Boundary Method (IBM) [3] was introduced by Peskin in 1972 for the solution of heart valve problems, where the computation of the fluid structure interaction problem is performed on a background Eulerian Cartesian grid and a body force is added to the fluid to account for the presence of any immersed solid. There have been several extensions of this initial immersed methodology since its inception. As an example, one of its extensions is the Immersed Structural Potential Method (ISPM) [4,5]. This method overcomes some of the shortcomings of the IBM, enabling the consideration of continuum deformable solids or the reduction in the number of interpolation and spreading operations. The ISPM was originally developed with focus on the solution of single-phase fluid-structure interaction problems, with particular emphasis in haemodynamic applications [4,5].

Immersed methods have not yet been fully exploited in the context of hydrodynamics applications. Solving hydrodynamic problems, such as the interaction between air, water and offshore structures, is dominated by the use of body fitted methods [6] or some specific immersed methodologies based on the discrete forcing approach [7], including the Cartesian grid method [8], the ghost cell method [9] and the direct forcing method [10,11].

The objective of this thesis is to further extend the application range of immersed computational approaches in the context of the hydrodynamics and present a novel general framework for the simulation of fluid-structure interaction problems involving rigid bodies, flexible solids and multiphase flows. The proposed method aims to overcome shortcomings such as the restriction of having to deal with similar density ratios among different phases or the restriction to solve single-phase flows. The new framework will be capable of coping with large density ratios, multiphase flows and will be focussed on hydrodynamic problems. The two main challenges to be addressed are:

- The representation, evolution and compatibility of the multiple fluid-solid interface.

- The proposition of a unified framework containing multiphase flows, flexible structures and rigid bodies with possibly large density ratios.

First, a new variation of the original IBM is presented by rearranging the governing equations which define the behaviour of the multiple physics involved. The formulation is compatible with the 'one-fluid' equation [12] for two phase flows and can deal with large density ratios with the help of an anisotropic Poisson solver [12].

Second, deformable structures and fluid are modelled in an identical manner except for the deviatoric part of the Cauchy stress tensor. The challenging part is the calculation of the deviatoric part of the Cauchy stress in the structure, which is expressed as a function of the deformation gradient tensor. The technique followed in this thesis is that of the original ISPM [4, 5], but re-expressed in terms of the Cauchy stress tensor.

Any immersed rigid body is considered as an incompressible non-viscous continuum body with an equivalent internal force field which constraints the velocity field to satisfy the rigid body motion condition. The 'rigid body' spatial velocity is evaluated by means of a linear least squares projection of the background fluid velocity, whilst the immersed force field emerges as a result of the linear momentum conservation equation. This formulation is convenient for arbitrary rigid shapes and for different cases of rigid body motion: uniform translational velocity, rotation around a fixed point and the most general translation-rotation.

A characteristic or indicator function, defined for each interacting continuum phase, evolves passively with the velocity field. Generally, there are two families of algorithms for the description of the interfaces, namely, Eulerian grid based methods (interface-capturing) and Lagrangian particle/mesh based methods (interface tracking). In this thesis, the interface capturing Level Set method [13] is used to capture the fluid-fluid interface, due to its advantages to deal with possible topological changes. In addition, an interface tracking Lagrangian based meshless technique is used for the fluid-structure interface due to its benefits at ensuring mass preservation.

From the fluid discretisation point of view, the discretisation is based on the standard Marker-and-Cell method [14] in conjunction with a fractional step approach for the pressure/velocity decoupling. The thesis presents a wide range of applications for multiphase flows interacting with a variety of structures (i.e. rigid and deformable). Several numerical examples are presented in order to demonstrate the robustness and applicability of the new methodology.

"Anyone who has never made a mistake has never tried anything new".

Albert Einstein (1879 – 1955)

Contents

List of Figures

List of Tables

Nomenclature

Classical symbols and notations:

δ_{ij}	Kronecker symbol equal to 1, if $i = j$ and to 0 if $i \neq j$
$\mathcal{E}_{ij}$	Levi-Civita symbol
$\phi_{,i}$	Partial derivative $\partial\phi/\partial x_i$

Einstein summation convention will be used:

$$\sigma_{ij,j} = \sum_{j=1}^{3} \frac{\partial \sigma_{ij}}{\partial x_j}, \qquad \sigma_{ij} n_j = \sum_{j=1}^{3} \sigma_{ij} n_j$$

Introducing the following notation

Ω	Domain in $\mathbb{R}^2$ or $\mathbb{R}^3$
$\Omega_f, \Omega_r, \Omega_s$	Fluid domain, rigid body domain and deformable structure domain
$\boldsymbol{X}, X_I$	Material or Lagrangian coordinates
t	Time variable
$\boldsymbol{\Psi}$	Mapping from $\boldsymbol{X}$ to $\boldsymbol{x}$
$\boldsymbol{x}, x_i$	Spatial or Eulerian coordinates $\boldsymbol{x} = \boldsymbol{\Psi}(\boldsymbol{X}, t)$
$\boldsymbol{F}, F_{iI}$	Deformation gradient tensor, $F_{iI} = \partial x_i / \partial X_I$
$\boldsymbol{C}$	Right Cauchy-Green deformation tensor, $\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}$
$\boldsymbol{b}$	Left Cauchy-Green deformation tensor, $\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T$
$\boldsymbol{E}$	Lagrangian or Green strain tensor, $\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I})$
J	Jacobian of the transformation $J = \det(\boldsymbol{F})$
$\boldsymbol{\nabla}$	Gradient operator in the deformed space, $\partial/\partial x_i$
$\boldsymbol{\nabla}\cdot$	Divergence operator in the deformed space
Δ	Laplacian operator in the deformed space
$\boldsymbol{\nabla}_0$	Gradient operator in the undeformed space, $\partial/\partial X_i$
$\frac{\mathrm{d}}{\mathrm{d}t}$	Material derivative with respect to time
$\frac{\partial}{\partial t}$	Spatial derivative with respect to time
$\boldsymbol{u}$	Velocity field, $\boldsymbol{u} = \partial \boldsymbol{x}(\boldsymbol{X}, t)/\partial t$
$\int_\Omega dv$	Volume integration over spatial volume
$\int_{\partial\Omega} da$	Surface integration over spatial area
$\int_{\Omega_0} dV$	Volume integration over material volume
$\int_{\partial\Omega_0} dA$	Surface integration over material area

$\boldsymbol{t}$	Traction vector
$\boldsymbol{\sigma}, \sigma_{ij}$	Cauchy stress tensor
$\boldsymbol{\sigma}', \sigma'_{ij}$	Deviatoric part of the Cauchy stress tensor
$\boldsymbol{\tau}, \tau_{ij}$	Kirchhoff stress tensor
$\boldsymbol{P}, P_{ij}$	First Piola-Kirchhoff stress tensor
$\boldsymbol{S}, S_{ij}$	Second Piola-Kirchhoff stress tensor
p	Pressure field
$\boldsymbol{n}, n_i$	Unit outward normal on the boundary in the deformed space
$\boldsymbol{N}, N_i$	Unit outward normal on the boundary in the undeformed space
κ	Surface curvature
Γ	Interface between two difference phases
ρ_0	Density in the undeformed space
ρ	Density in the deformed space
$\boldsymbol{g}, g_i$	Gravitational acceleration
$\boldsymbol{\phi}$, ϕ_a	Level set function vector (component)
$\boldsymbol{\chi}$, χ_a	Indicator function vector (component)
$\hat{\boldsymbol{\chi}}$, $\hat{\chi}_a$	Approximation of indicator function vector (component)
$\delta(\boldsymbol{x})$	Dirac delta distribution evaluated at $\boldsymbol{x}$
$\hat{\delta}(\boldsymbol{x})$	Approximation of the Dirac delta distribution evaluated at $\boldsymbol{x}$
sgn	Sign function or signum function
tr	Trace of a second order tensor
det	Determinant of a second order tensor
ψ	Strain energy functional per unit undeformed volume
μ	Dynamic viscosity of the fluid
γ	Surface tension coefficient of the fluid
G	Shear modulus of the solid
E	Young's modulus of the solid
$\boldsymbol{\omega}, \omega_i$	Angular velocity of the rigid body
$\boldsymbol{u}_c$	Translational velocity of the rigid body
$\boldsymbol{x}_c$	Centre of gravity of the rigid body
$\boldsymbol{J}$	Moment of inertia tensor of the rigid body
M	Mass of the rigid body
$\boldsymbol{f}(\boldsymbol{x}_s)$	Immersed force evaluated on Lagrangian particle $\boldsymbol{x}_s$
$\boldsymbol{f}(\boldsymbol{x})$	Immersed force evaluated on Eulerian grid position $\boldsymbol{x}$
$\mathsf{P}(\boldsymbol{u})$	Projection operator of the velocity field $\boldsymbol{u}$ into a rigid body motion velocity

Abbreviations

ALE	Arbitrary Lagrangian Eulerian
BFECC	Back and Forth Error Compensation and Correction
DLM	Distributed Lagrange Multiplier
DF/FD	Direct-Forcing Fictitious Domain
DF/IB	Direct-Forcing Immersed Boundary
FEM	Finite Element Method
FSI	Fluid-Structure Interaction
FVM	Finite Volume Method
HVF	Helicoidal Vector Field
IBM	Immersed Boundary Method
IFEM	Immersed Finite Element Method
ISPM	Immersed Structural Potential Method
LES	Large Eddy Simulation
LSM	Level Set Method
MAC	Marker-And-Cell
MPM	Material Point Method
N-S	Navier-Stokes
PFEM	Particle Finite Element Method
QUICK	Quadratic Upstream Interpolation for Convective Kinematics
RANS	Reynolds-Averaged Navier-Stokes
RKPM	Reproducing Kernel Particle Method
SPH	Smoothed-Particle Hydrodynamics
SVF	Solenoidal Vector Field
VOF	Volume-Of-Fluid

Part I

Preliminaries

Chapter 1

Introduction

1.1 Motivation

Ships, offshore infrastructures, wind and hydropower related civil engineering constructions and oil and gas platforms are typical coastal engineering systems. The design, maintenance and protection of this infrastructure is now widely recognised as critically important at national and worldwide levels. The above considerations provide a strong motivation for the research presented in this work (see Figure 1.1).

Figure 1.1: Photographs of coastal areas, by Liang Yang.

In the past few decades, a wealth of numerical methods have been developed to gain a deeper insight into these complex engineering problems. Out of the different stages of the design/engineering process, computational modelling is frequently used for prediction and validation. In the coastal and marine engineering fields, where experimental studies are usually expensive or not even possible, development of

sound and robust numerical methods is essential.

Such problems often involve air, seawater, sediments and oil interacting with the surrounding naval or civil engineering structures. Due to the complex physical interaction, establishing a mathematical model and obtaining its numerical solution is not a trivial task. This thesis aims to offer an efficient and unified computational framework to solve some of these complex multi-physics phenomena.

1.2 Scope of the thesis

In the field of fluid mechanics, potential flow equations, Euler equations and Navier-Stokes equations are normally used as the basis for the description of typical flow phenomena [21]. Among these equations, the Navies-Stokes equations are regarded as the most difficult to solve, and the others can be considered as simplifications [22]. In the past few decades, there has been huge advancement in the development of numerical methods for the solution of flow equations, including advanced Finite Difference based methods [23], vortex and cell centred Finite Volume based methods [24], stabilised Finite Element based methods [25] and Discontinuous Galerkin based methods [26], to name but a few. However, there are still some important topics which require further investigation, such as the modelling of multi-phase flows, turbulence or the interaction with immersed structures [27]. Prior to moving into these topics, it is necessary to limit the scope of this work.

The objective of this thesis is the study of multi-phase flows and their interaction with possible surrounding solids. These problems are extremely important and commonplace in coastal and marine engineering applications, specifically

- Multiphase flow interaction.[1]
- Fluid-Structure Interaction.
- Fluid and rigid-body interaction.

It is NOT the purpose of this thesis to discuss

- Nanoscale phenomena (atoms or molecules): The starting point of this work is based on the use of the continuum mechanics assumption, stating that a continuum body can be sub-divided into infinitesimal elements [28].
- Turbulence: Although many engineering problems of interest involve turbulent phenomena, popular turbulence modelling techniques [29] such as boundary layer modelling [30], Reynolds-Averaged Navier-Stokes (RANS) [31], Large Eddy Simulation (LES) [32] or Detached Eddy Simulation (DES) [33] will not be discussed in this thesis.

[1] The multiphase flow denotes a mixture of multiple fluids that can be miscible or immiscible. The multiphase flow investigated in this work concerns only an idealised case, where all fluids are strictly immiscible and are separated by clear interfaces.

- Shock waves and discontinuities: There are no pressure waves generated under the assumption that the continuum model under consideration is incompressible.

- New discretisation techniques: There are many new discretisation techniques which have been established in recent years, such as high order Discontinuous Galerkin methods [34], Variational Multiscale methods [35] or Spectral methods [36], which will not be pursued in this thesis. The discretisation used herein is based on the well-established Marker-And-Cell method [37], for its simplicity and stable nature due to the staggered interpolation of the velocity and pressure fields.

1.3 Literature review

Over the past few decades and with greater emphasis in the last few years, the computational mechanics research community has devoted great effort to the modelling of multiphase flows, Fluid-Structure Interaction (FSI) phenomena and fluid-rigid body coupling problems. In the following, some relevant bibliographic references are presented and organised by topics of interest.

1.3.1 Multiphase flows: the 'one-fluid' model and the description of the interface

In the simplest case of immiscible fluids, the most common used multiphase model is the so-called 'one-fluid' formulation [38].

The computational modelling of a multi-phase flow is carried out in a similar way to that of the single-phase flow, apart from the consideration of the interface evolution, which is considered dependent on the flow velocity field. Therefore, the interface kinematics problem and the so-called physical dynamics (momentum conservation equation) can be studied as separate yet coupled processes. In this manner, the difficult part of the 'one-fluid' formulation rests in the modelling of the interface kinematics.

Generally speaking, there are two possible descriptions of the surface kinematics, based on a material description or based on a spatial description. In other words, the interface can be 'tracked' via an explicit conforming mesh or be 'captured' with an implicit marker function. The most popular implicit descriptions of the interfaces are established by the Volume Of Fluid (VOF) method [39–41], the Level Set method [42] and the Phase Field method [43]. In the case of using an explicit description, a conforming mesh is usually employed to follow the interface in conjunction with an Arbitrary Lagrangian Eulerian (ALE) formulation accompanied possibly with a re-meshing algorithm [44]. Alternatively, the interface can be described as the boundary of an immersed body defined by a cloud of interpolation points or particles, as in the case of the front tracking method [45–47] or the hybrid method [48–52].

It is worth emphasising that the front tracking method [45], the Volume Of Fluid method [53], the Level Set method [54] and the Phase Field method [55] have all been adopted by the computer science and computer graphics communities, where the main applications have been in the film and virtual reality industries. However, computer graphics interests tend to focus on the generation of plausible graphics, yet not necessarily in the accurate modelling of the physical phenomena.

1.3.2 Fluid-Structure Interaction: body fitted vs immersed methods

In the field of Fluid-Structure Interaction[2] problems, two families of methodologies have been used in practice: body fitted approaches [1,2,56] and immersed methods [3]. Methods within each family have some strengths and weaknesses. Within the group of body fitted methodologies, the main disadvantage is the computational cost related to the update of the mesh and the possible need to resort to re-meshing algorithms, a factor particularly critical in the case of three-dimensional simulations.

The Immersed Boundary Method (IBM) was first introduced by Peskin [3] in 1972 to simulate the deformation of heart valves. The distinguishing feature of this method is the fact that the simulation is carried out on a fixed Eulerian Cartesian grid, which does not conform to the current geometry of that of the deformed immersed structure. A solid-to-fluid interpolated body force is added to the fluid to account for the presence of any immersed deformable solid. Figure 1.2 shows an immersed type methodology compared against a body fitted ALE approach (with and without re-meshing).

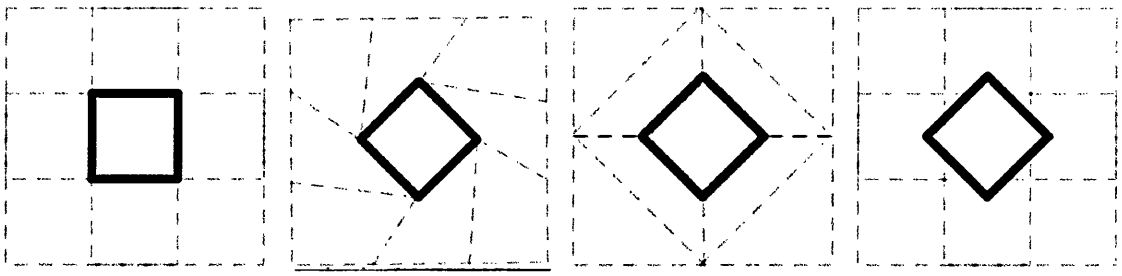


Figure 1.2: Comparison of different FSI methodologies. From left to right: 1.) Initial undeformed reference state. 2.) Body fitted approach. 3.) Body fitted (with re-meshing) approach. 4.) Immersed approach.

Early versions of the IBM were limited to the case of immersed solids made up of fibres satisfying the generalised Hooke's law [57–60]. The original method was defined on the basis of the Finite Difference Method, and an approximation of the Dirac delta distribution was used to map information between the Eulerian fluid and the Lagrangian immersed solid.

[2]The 'structure' referred herein is a flexible solid.

The mathematical formulation of the IBM has been constantly evolving since its inception. The IBM was first extended to allow for the consideration of any immersed hyperelastic material by means of the introduction of a strain energy functional [61]. The immersed structure was then generalised to the case of any continuum derived from a potential energy functional [62]. The material properties of the continuum can be those of a viscoelastic structure [62], fluid [63] or dry foam [64]. In addition, adaptive re-meshing [65] was also introduced to improve the accuracy of the technique.

It is worth noticing that one of the extensions to the original IBM is the Immerse Structural Potential Method (ISPM) [4,5,66] by A. J. Gil et al., introduced for the solution of single-phase FSI problems interacting with highly deformable structures in the context of haemodynamic applications. From a continuum modelling point of view, the structure is modelled as a deviatoric strain energy functional and from a spatial discretisation point of view, as a collection of integration points. A structure preserving time integration scheme is used without the need to resort to a moving computational mesh in order to evaluate the deformation gradient of the immersed structure, yielding improved accuracy and stability over previous immersed based methodologies.

1.3.3 Fluid and rigid body interaction: the Eulerian rigid body approach

The modelling of an immersed rigid body as a deformable case of a very stiff solid can lead to numerical instabilities or even an ill-posed problem [67]. As a result, it is advisable to carry out the modelling of a flexible body and a rigid body by employing different approaches. In the majority of engineering applications, rigid bodies tend to be modelled via classical Lagrangian mechanics [1,68].

The first two immersed methodologies that model a computational region subject to a velocity constraint are the Distributed Lagrange-Multiplier/Fictitious-Domain (DLM) method of Patankar [69] and the Stress-Distributed Lagrange-Multiplier (Stress-DLM) of Glowinski [70]. Both of these methods considered the rigid body to be of the same density as that of the underlying fluid. Glowinski [70] proposes that within the rigid body region, the space varying velocity field $\boldsymbol{u}$ must satisfy the constraint $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_c + \boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{x}_c)$, where $\boldsymbol{u}_c$ and $\boldsymbol{x}_c$ are the velocity and location of the centre of gravity of the rigid body and $\boldsymbol{\omega}$ is the angular velocity. Alternatively, Patankar [69] considers that the deformation rate tensor describing the kinematics of the rigid body is zero. By using the deformation rate constraint, the emerging Lagrangian multiplier does not involve angular or translational velocity and it has the same structure as that of a stress tensor; hence the name of the method (stress-DLM) [71]. In general, these two DLM methods are solved in a weak manner with the help of an iterative Uzawa algorithm [69,70], showing to be very time consuming.

To avoid the need for a Lagrange multiplier formulation, Patankar [72] proposed

an alternative method for the direct approximation of the rigid velocity field. A projection procedure is introduced in order to ensure the conservation of linear and angular momenta.

In the computer graphics community, Carlson [73] adopted this idea and called it the 'rigid fluid method'. However, this approach cannot deal with the consideration of light solids. In the computational physics community, the direct-forcing immersed boundary (DF/IB) [74] method and the direct-forcing fictitious domain (DF/FD) [75] method can also be regarded as particular cases of this general family of methods. In references [76,77], inertia effects are treated implicitly by solving an anisotropic Poisson equation and the immersed rigid body force (i.e. Lagrange multiplier) is constructed in a very similar way to that of the direct forcing method [78,79]. The general advantage of these methods is the consideration of the rigid body dynamics in an Eulerian way, analogous to a fluid, facilitating the modelling of rigid and fluid coupling problems by means of the two-phase flow approach described above.

1.3.4 Limitations and gaps of the existing techniques

From the above literature review, there are several limitations and gaps in the current methodologies which will be the focus of this thesis, specifically:

- The solution of problems involving multiphase flows, immersed structures and immersed rigid bodies is a complex problem, which still requires further attention.

- The evolution of multi-interface problems with immersed structures and multiple phases is still an open problem. Most of the existing research work thus far is based on two-phase flows using a level set function.

- In the case of FSI problems, immersed computational techniques suffer when having to deal efficiently with large density ratio problems, as typically encountered in hydrodynamic engineering applications.

- The consideration of an Eulerian based method for the modelling of rigid bodies immersed in fluid is still an open problem.

1.4 Contribution of this thesis to the state of the art in this field of science

To overcome the above limitations and solve some relevant FSI problems, a general immersed methodology is proposed in this thesis. This includes:

- A unified computational framework for the multi-physics problem. The proposed immersed methodology is capable of solving multi-phase fluids interacting with immersed deformable structures and rigid bodies on a continuum

level, unlike most of other immersed based methods, where the methodology is based on the use of specific interpolating kernel functions. As a result, the choice of the indicator function for the immersed structure, rigid body and fluid phases is completely arbitrary.

- Identification of the different phases. The characteristic or indicator function, which is evaluated for each interacting phase, moves passively with the velocity field. In this work, the level set method [13] is preferred in order to capture the fluid interface, which is advantageous when dealing with topological changes. Moreover, Lagrangian particles or integration points are used to describe the motion of the immersed rigid/flexible body facilitating the preservation of mass. However, as mentioned earlier, this choice is not mandatory, namely, tracking the fluid interface by using particles or capturing the solid interface via an Eulerian based method can also be an alternative.

- The deformable solid governing equations are written in an Eulerian form, compatible with the classical Navies-Stokes equations used to describe the viscous fluid domain. The update of the deformation gradient tensor from the spatial velocity gradient field prevents the appearance of locking, which is a problem often encountered in the context of incompressible problems when using low order approximations.

- The rigid body momentum equation is modelled along with the Navier-Stokes equations and supplemented with a rigid motion constraint. It is well known that the incompressibility condition requires the velocity to belong to a Solenoidal Vector Field (SVF). Alternatively, the rigid body condition can be understood to yield the velocity field to be a Helicoidal Velocity Field (HVF) [21].

- A rigid projection operator is proposed for the rigid motion constraint by using a linear least square approach, where the immersed rigid body forces are evaluated explicitly. Therefore, the rigid body coupling can be easily implemented in a single-phase fluid solver. Large density ratio problems are overcome via the fractional step method by solving an anisotropic Poisson equation.

1.5 Layout of the thesis

In order to elaborate the objectives indicated in the previous section, the main body of this thesis is broken down into four parts with the following chapters:

- **Part II: General formulation**

- Chapter 2 introduces the classical Eulerian conservation laws (conservation of linear momentum, conservation of the mass) for a single-phase continuum. It also presents constitutive models for a Newtonian flow and a hyperelastic

deformable solid. In addition, it discusses the rigid case as a degenerate limit of a deformable solid, named 'rigid continuum'.

- Chapter 3 proposes a unified framework for a multiphase continuum. It starts by describing the classical boundary fitted formulation for a two-phase continuum with the necessary interface conditions. Then it provides a 'one-phase' model by introducing an indicator function. The proposed 'one-phase' model is capable of dealing with multiple fluids, structures and rigid bodies interacting in a 'one-phase' description. In order to treat the interface problem, this chapter presents some of the available techniques for the description of interfaces, namely as Eulerian (capturing) Level Set method and Lagrangian (tracking) Particle Method. The chapter concludes with a multiple Level Set method proposed for multiple phase problems.

- **Part III: Numerical techniques**

- Chapter 4 discusses the details of the numerical discretisation of the multiphase flow governing equations. A very efficient low order finite volume scheme set in a Cartesian staggered mesh is chosen for the spatial discretisation. A well established fractional step method is considered for the fluid-pressure decoupling. The consideration of surface tension effects is also discussed in this chapter for the modelling of bubble dynamics.

- Chapter 5 presents the discrete representation of the deformable immersed solid. The numerical techniques are adopted from the Immersed Structural Potential Method, but reinterpreted in a fully Eulerian manner.

- Chapter 6 presents first an iterative procedure for the solution of immersed rigid bodies, consistent with the field introduced by a deformable solid. The chapter then describes a linear least squares method for the projection of the velocity of the rigid body. The novel method is capable of dealing with arbitrary rigid body motions.

- **Part IV: Numerical examples**

- Chapter 7 presents some interesting numerical examples, ranging from two-phase bubble dynamics to the classical dam break problem. It also includes three-phase problems, which are not commonly solved by using level sets.

- Chapter 8 presents some hydrodynamics problems which involve air, water and immersed structures, illustrating the capability of the proposed method. The solution of these problems using a boundary fitted method would be prohibitively expensive in terms of computational cost, or even not possible in the case of using the classical Immersed Boundary Method.

- Chapter 9 presents some numerical examples for rigid structures interacting with water and air, often encountered in ship and coastal engineering applications. It starts by presenting a fixed rigid body, then moves to the consideration of the modelling of a rigid body driven by an external velocity field.

The guideline to the overall organisation of this thesis is summarised on the following chart.

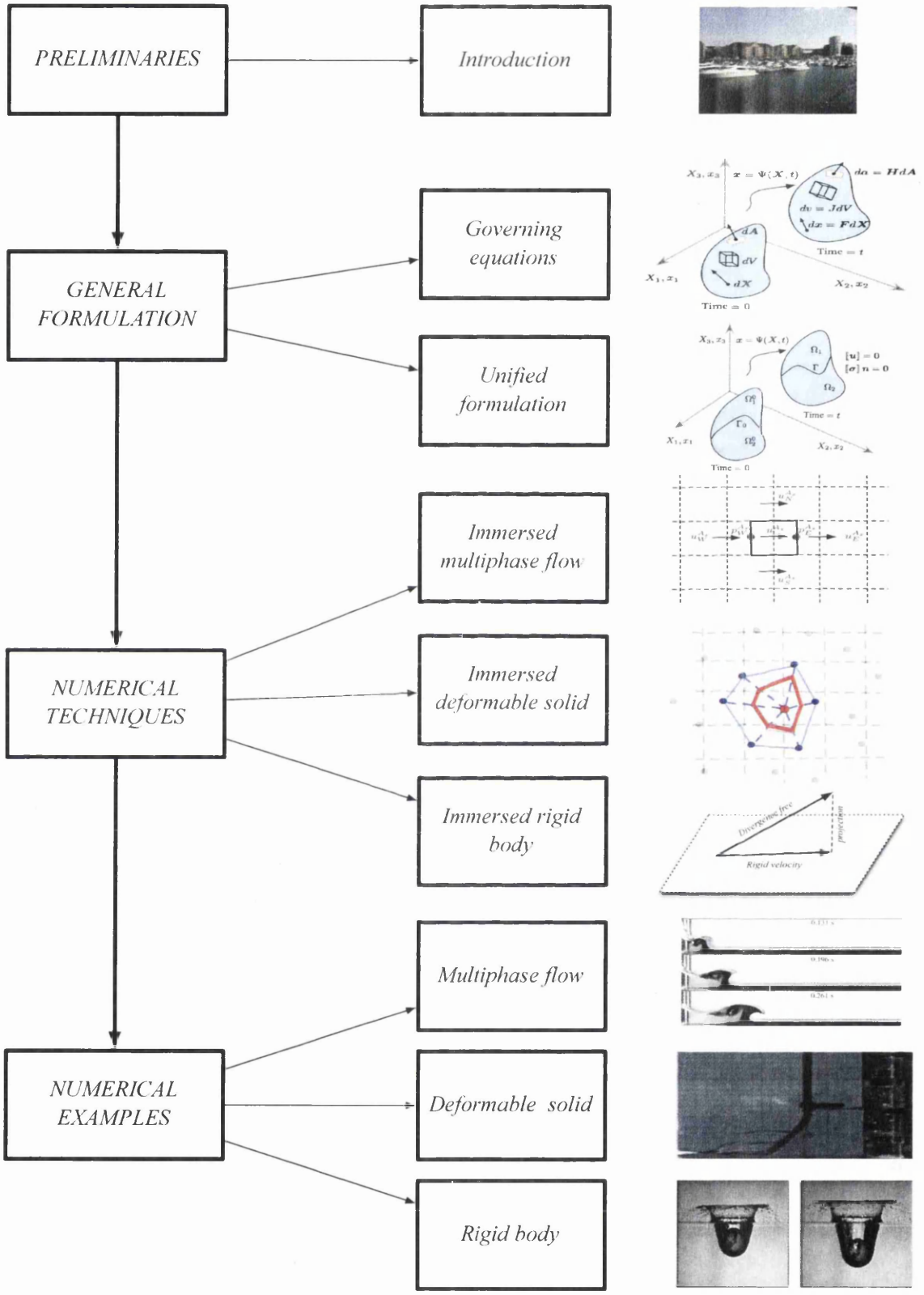


Figure 1.3: Structure and content of this thesis.

Part II

General Formulation

Chapter 2

Governing equations

2.1 Kinematics of a continuum

This section presents the fundamentals of the finite deformation of a continuum, that is, the study of the motion without reference to its cause.

2.1.1 Motion

Let us consider the motion of a continuum from its reference or material configuration $\Omega_0 \subset \mathbb{R}^3$, with boundary $\partial\Omega_0$, into its spatial or current configuration $\Omega \subset \mathbb{R}^3$ at time t, with boundary $\partial\Omega$, by means of a mapping function $\boldsymbol{\Psi}$, defined as

$$\boldsymbol{x}(t) = \boldsymbol{\Psi}(\boldsymbol{X}, t), \tag{2.1}$$

where $\boldsymbol{x} = [x_1, x_2, x_3]^T$ represent the current position of a particle initially at $\boldsymbol{X} =$

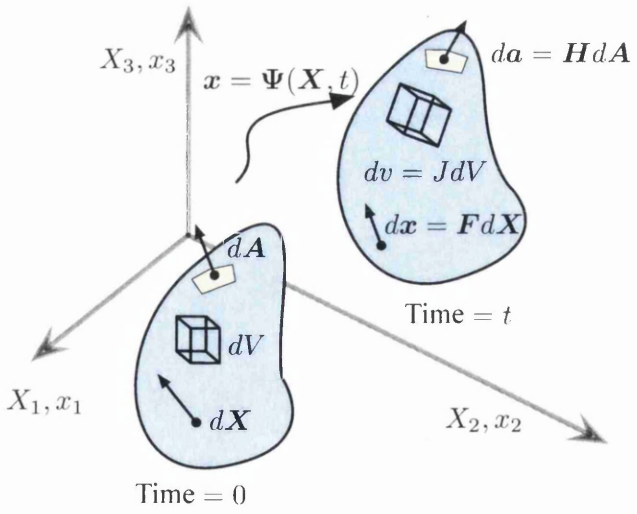


Figure 2.1: The general motion of a system.

$[X_1, X_2, X_3]^T$ (see Figure 2.1). For a given value of time t, the above equations Eq.

(2.1) represents the mapping between the undeformed and deformed configurations.

2.1.2 Material and spatial description

A careful distinction has to be made between the description that can be chosen to make precise the deformation of the continuum. In simple words, the material description is used to describe relevant physical magnitudes in terms of where the continuum was before the deformation whilst the spatial description is employed to describe them in terms of where the continuum is during the deformation. Alternatively, these are often referred to as Lagrangian (material) and Eulerian (spatial) descriptions.

Fluid mechanicians almost exclusively work in terms of a spatial description, because it is not convenient to describe the behaviour of a material particle. Solid mechanicians will, in general, at some stage of a formulation, have to consider the constitutive behaviour of the material particles, which requires a material description. However, there are not unbreakable laws. In this thesis, a fully spatial description will be used for both solid and fluid phases. In addition, the velocity of a particle is defined as the time derivative of the mapping function as

$$\boldsymbol{u}(\boldsymbol{X},t) = \frac{\partial \boldsymbol{\Psi}(\boldsymbol{X},t)}{\partial t}. \tag{2.2}$$

Observe that the velocity is a spatial vector despite the fact that the equation has been expressed in terms of the material coordinates of the particle $\boldsymbol{X}$. By inverting Eq. (2.1) the velocity can be expressed as function of the spatial position $\boldsymbol{x}$ and time as

$$\boldsymbol{u}(\boldsymbol{x},t) = \boldsymbol{u}(\boldsymbol{\Psi}^{-1}(\boldsymbol{x},t),t). \tag{2.3}$$

Given a general scalar, vector or tensor quantity $\boldsymbol{f}$, expressed in material coordinates $\boldsymbol{X}$, the time derivative of $\boldsymbol{f}(\boldsymbol{X},t)$ is denoted henceforth by $\frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{X},t)}{\mathrm{d}t}$. This expression measures the change in $\boldsymbol{f}$ associated with a specific particle initially located at $\boldsymbol{X}$, and it is known as the material time derivative of $\boldsymbol{f}$. Using the chain rule, the material derivative $\frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{X},t)}{\mathrm{d}t}$ can be expressed as

$$\frac{\mathrm{d}\boldsymbol{f}(\boldsymbol{X},t)}{\mathrm{d}t} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},t)}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{x},t)}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{\Psi}(\boldsymbol{X},t)}{\partial t} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},t)}{\partial t} + (\boldsymbol{\nabla} \boldsymbol{f})\boldsymbol{u}. \tag{2.4}$$

The first term on the last right hand side of Eq. (2.4) is called the 'local time derivative', whilst the second term is referred to as the 'convective derivative'.

2.1.3 Deformation gradient

A key quantity in defining the deformation of a continuum is the deformation gradient tensor $\boldsymbol{F}$, which is defined as the material gradient of the spatial position

as,

$$\boldsymbol{F} = \nabla_0 \boldsymbol{x} = \frac{\partial \boldsymbol{x}(\boldsymbol{X}, t)}{\partial \boldsymbol{X}}, \tag{2.5}$$

where ∇_0 denotes the gradient with respect to the material configuration. This two-point deformation gradient tensor maps a differential elemental material vector $d\boldsymbol{X}$ into the corresponding spatial vector $d\boldsymbol{x}$ as

$$d\boldsymbol{x} = \boldsymbol{F} d\boldsymbol{X}. \tag{2.6}$$

Analogously, volume elements in the reference and current configurations are related through the determinant of $\boldsymbol{F}$, denoted as J. That is,

$$dv = J dV, \quad J = \det \boldsymbol{F}. \tag{2.7}$$

The two-point cofactor tensor of $\boldsymbol{F}$, denoted by $\boldsymbol{H}$, express the relation between an area vector in the reference domain, $d\boldsymbol{A} = \boldsymbol{N} dA$, and an area vector in the spatial domain, $d\boldsymbol{a} = \boldsymbol{n}\, da$, as

$$d\boldsymbol{a} = \boldsymbol{H} d\boldsymbol{A}, \quad \boldsymbol{H} = J\boldsymbol{F}^{-T}. \tag{2.8}$$

2.1.4 Strain

As a general measure of deformation, consider the change in the scalar product of the two elemental vectors $d\boldsymbol{X}_1$ and $d\boldsymbol{X}_2$ as they deform to $d\boldsymbol{x}_1$ and $d\boldsymbol{x}_2$. This change involves both the stretching and variation in the enclosed angle between the two vectors. The spatial scalar product $d\boldsymbol{x}_1 \cdot d\boldsymbol{x}_2$ can be found in terms of the material vectors $d\boldsymbol{X}_1 \cdot d\boldsymbol{X}_2$ as

$$d\boldsymbol{x}_1 \cdot d\boldsymbol{x}_2 = d\boldsymbol{X}_1 \cdot \boldsymbol{C} d\boldsymbol{X}_2, \tag{2.9}$$

where $\boldsymbol{C}$ is the right Cauchy-Green deformation tensor, which is given in terms of the deformation gradient $\boldsymbol{F}$ as

$$\boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}. \tag{2.10}$$

Note that as in Eq. (2.9) the tensor $\boldsymbol{C}$ operates on material vectors and $\boldsymbol{C}$ is called a material tensor quantity. Alternatively, the initial material scalar product $d\boldsymbol{X}_1 \cdot d\boldsymbol{X}_2$ can be obtained in terms of the spatial vector $d\boldsymbol{x}_1$ and $d\boldsymbol{x}_2$ via the left Cauchy-Green or Finger tensor $\boldsymbol{b}$ as

$$d\boldsymbol{X}_1 \cdot d\boldsymbol{X}_2 = d\boldsymbol{x}_1 \cdot \boldsymbol{b}^{-1} d\boldsymbol{x}_2, \tag{2.11}$$

where $\boldsymbol{b}$ is the spatial tensor defined as

$$\boldsymbol{b} = \boldsymbol{F}\boldsymbol{F}^T. \tag{2.12}$$

The Lagrangian strain tensor $\boldsymbol{E}$ and the Almansi strain tensor $\boldsymbol{e}$ are defined as [28],

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}); \quad \boldsymbol{e} = \frac{1}{2}(\boldsymbol{I} - \boldsymbol{b}^{-1}), \tag{2.13}$$

where $\boldsymbol{I}$ is the identity tensor. The strain or deformation rate tensor $\boldsymbol{d}$ gives the rate of extension per unit of current length of a spatial line element as

$$\boldsymbol{d} = \frac{1}{2}\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right). \tag{2.14}$$

In terms of the language of pull back and push forward operations [28], the material and spatial strain measures defined above (2.13) can be related. The push-forward and pull-back operations can be written as

Push forward

$$\boldsymbol{e} = \boldsymbol{F}^{-T}\boldsymbol{E}\boldsymbol{F}^{-1} \tag{2.15}$$

Pull backward

$$\boldsymbol{E} = \boldsymbol{F}^{T}\boldsymbol{e}\boldsymbol{F} \tag{2.16}$$

When dealing with incompressible and nearly incompressible materials it is necessary to separate the volumetric contribution from the distortional components of the deformation. Such a separation ensures that the distortional component, namely $\hat{\boldsymbol{F}}$, does not imply any change in volume. Noting that the determinant of the deformation gradient gives the volume ratio, $\det(\hat{\boldsymbol{F}}) = 1$, then the above condition can be achieved as follows

$$\hat{\boldsymbol{F}} = J^{-1/3}\boldsymbol{F}. \tag{2.17}$$

2.1.5 Rigid body motion

In this section, we briefly review the principles of Classical Mechanics for the description of a rigid body. An object which does not have any internal degree freedom is called a 'rigid body' [21]. An alternative definition can be that of a collection of N points constrained so that the distance between the points remains fixed. As a result, for any pair of points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ within the rigid body

$$||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = \text{constant}. \tag{2.18}$$

Alternatively, the motion of a rigid body in three (two) dimensions can be defined by means of six (three) degrees of freedom: 3 (2) translations + 3 (1) rotations. The position of the whole body is usually represented by the position of the centre of mass $\boldsymbol{x}_c$ and the rotation $\boldsymbol{\theta}$, as seen in Figure 2.2. Eq. (2.18) leads to several necessary and sufficient conditions for the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$. Eq. (2.18) holds if and only if

$$\frac{\mathrm{d}}{\mathrm{d}t}||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = 0. \tag{2.19}$$

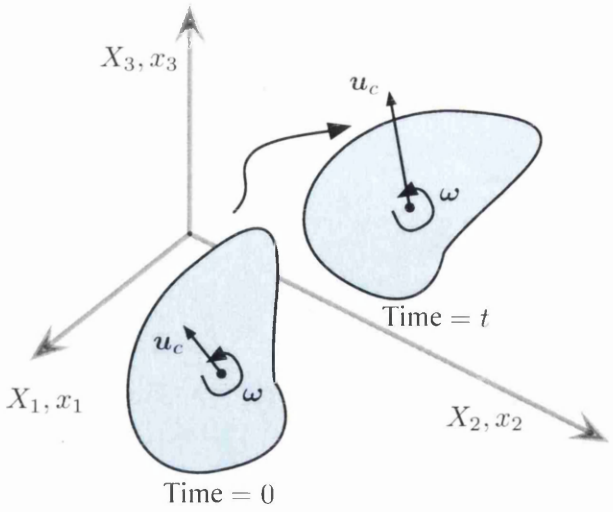


Figure 2.2: The motion of a rigid body

Alternatively, above Eq. (2.19) can be re-written as one of the following options:

Option 1, the velocity must satisfy

$$(\boldsymbol{x}_i - \boldsymbol{x}_j) \cdot (\boldsymbol{u}(\boldsymbol{x}_i) - \boldsymbol{u}(\boldsymbol{x}_j)) = 0. \tag{2.20}$$

Option 2, the velocity field must satisfy

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_c + \boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{x}_c). \tag{2.21}$$

where $\boldsymbol{x}_c$ is the position of the centre of mass, $\boldsymbol{u}_c = \boldsymbol{u}(\boldsymbol{x}_c)$ is the velocity of the centre of mass and $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]^T$ is the angular velocity of the rigid body motion.

Option 3, not necessarily knowing the centre of mass, the rigid body motion Eq. (2.21) can also be written as

$$\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{\omega} \times \boldsymbol{x} + \boldsymbol{u}_{rb}. \tag{2.22}$$

where the constant $\boldsymbol{u}_{rb} = [u_{rb,1}, u_{rb,2}, u_{rb,3}]^T$ is defined as $\boldsymbol{u}_{rb} = \boldsymbol{u}_c - \boldsymbol{\omega} \times \boldsymbol{x}_c$. In index notation, the above Eq. (2.22) can be written as follows

$$u_1 = -\omega_3 x_2 + \omega_2 x_3 + u_{rb,1};\ u_2 = \omega_3 x_1 - \omega_1 x_3 + u_{rb,2};\ u_3 = -\omega_2 x_1 + \omega_1 x_2 + u_{rb,3}. \tag{2.23}$$

As it can be observed, the above equations (2.22) and (2.23) are fully spatial, hence, there is no Lagrangian property associated to the rigid body description, but only two unknown spatial vectors, namely $\boldsymbol{\omega}$ and $\boldsymbol{u}_{rb}$. This description for the velocity field $\boldsymbol{u}$ is referred in the literature as a Helicoidal Vector Field (HVF) [21].

Option 4, strain rate tensor $\boldsymbol{d}$ should be equal to zero

$$\boldsymbol{d} = \frac{1}{2}\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right) = \boldsymbol{0}. \tag{2.24}$$

Above options 1 to 4 are all necessary and sufficient conditions to describe the velocity field of a rigid body [21]. In addition, there are two interesting corollaries that can be inferred from the velocity field description of a rigid body.

First, this velocity field is a divergence free field $\nabla \cdot \boldsymbol{u} = 0$. Second, the velocity field is a linear function with respect to its spatial coordinates $\boldsymbol{x}$. The latter condition will inspire the use of a linear least squares method for the projection of an arbitrary vector velocity field.

2.2 Dynamics of a continuum

2.2.1 Conservation of mass

Using the hypothesis of mass conservation, the mass of an arbitrary spatial volume Ω must be preserved, leading to the following equation in integral form

$$0 = \frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega} \rho(\boldsymbol{x}, t)\, dv. \tag{2.25}$$

where ρ is the mass density per unit of deformed volume. Application of the Reynolds transport theorem yields

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega} \rho(\boldsymbol{x}, t)\, dv =& \frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_0} \rho(\boldsymbol{x}, t) J dV \\
=& \int_{\Omega_0} \left[\frac{\mathrm{d}\rho(\boldsymbol{x}, t)}{\mathrm{d}t} J + \rho(\boldsymbol{x}, t) \frac{\mathrm{d}J}{\mathrm{d}t} \right] dV \\
=& \int_{\Omega_0} \left[\left(\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} + \nabla \rho(\boldsymbol{x}, t) \cdot \boldsymbol{u} \right) J + \rho(\boldsymbol{x}, t) J \nabla \cdot \boldsymbol{u} \right] dV \\
=& \int_{\Omega} \left[\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} + \nabla \rho(\boldsymbol{x}, t) \cdot \boldsymbol{u} + \rho(\boldsymbol{x}, t) \nabla \cdot \boldsymbol{u} \right] dv \\
=& \int_{\Omega} \left[\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot (\rho(\boldsymbol{x}, t) \boldsymbol{u}) \right] dv.
\end{aligned} \tag{2.26}$$

As a result, the local form of the conservation of mass (in the absence of jump conditions) is

$$\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} + \nabla \cdot (\rho(\boldsymbol{x}, t) \boldsymbol{u}) = 0. \tag{2.27}$$

Equation (2.27) is known as the mass-conservation equation or continuity equation. For incompressible materials, the density ρ is constant in space and time, and the conservation law for the mass equation (2.27) then simplifies to[1]

$$\nabla \cdot \boldsymbol{u} = 0. \tag{2.28}$$

[1] In the case of using a Lagrangian description, an alternative expression for the incompressibility constraint is $J = 1$.

This description for the velocity field $\boldsymbol{u}$ is referred in the literature as a Solenoidal Vector Field (SVF) [21]. The conservation of mass becomes then a purely kinematic constraint.

2.2.2 Cauchy stress tensor

Consider a general deformable body at its current configuration. In order to develop the concept of stress, it is necessary to study the action of forces applied on the element area Δa of unit normal $\boldsymbol{n}$ in the neighbourhood of a spatial point $\boldsymbol{p}$. If the resultant force on this area is $\Delta \boldsymbol{p}$, the traction vector $\boldsymbol{t}$ corresponding to the normal $\boldsymbol{n}$ at $\boldsymbol{p}$ is defined as

$$\boldsymbol{t}(\boldsymbol{n}) = \lim_{\Delta a \to 0} \frac{\Delta \boldsymbol{p}}{\Delta a} \tag{2.29}$$

where the relationship between $\boldsymbol{t}$ and $\boldsymbol{n}$ must be such that it satisfies Newton's third law of action and reaction, which is expressed as

$$\boldsymbol{t}(-\boldsymbol{n}) = -\boldsymbol{t}(\boldsymbol{n}). \tag{2.30}$$

The traction vector $\boldsymbol{t}$ is related to the Cauchy stress tensor $\boldsymbol{\sigma}$ through the spatial unit outward normal $\boldsymbol{n}$ as follows

$$\boldsymbol{t}(\boldsymbol{n}) = \boldsymbol{\sigma}\boldsymbol{n}, \quad \boldsymbol{\sigma} = \sum_{i,j=1}^{3} \sigma_{ij} \boldsymbol{e}_i \otimes \boldsymbol{e}_j. \tag{2.31}$$

The Cauchy stress tensor $\boldsymbol{\sigma}$ is a spatial (Eulerian) tensor [28], with a dyadic representation in terms of three Cartesian vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ as presented in the above Eq. (2.31).

2.2.3 Conservation of linear momentum

The conservation of linear momentum for an arbitrary spatial volume can be expressed in integral form as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega} \rho \boldsymbol{u} \, dv = \int_{\Omega} \rho \boldsymbol{g} \, dv + \int_{\partial\Omega} \boldsymbol{t} \, da \tag{2.32}$$

where in Eq. (2.32), $\boldsymbol{t} = \boldsymbol{\sigma}\boldsymbol{n}$ is the traction vector and $\boldsymbol{g}$ represents the gravitational acceleration. Decomposition of the stress tensor $\boldsymbol{\sigma}$ into its volumetric $-p\boldsymbol{I}$ (where p represents the pressure) and deviatoric $\boldsymbol{\sigma}'$ components leads to

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega} \rho \boldsymbol{u} \, dv = \int_{\Omega} \rho \boldsymbol{g} \, dv - \int_{\partial\Omega} p\boldsymbol{n} \, da + \int_{\partial\Omega} \boldsymbol{\sigma}'\boldsymbol{n} \, da. \tag{2.33}$$

By making use of the Gauss divergence theorem, the above Eq. (2.33) can be rewritten as follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\boldsymbol{u}\,dv=\int_{\Omega}\rho\boldsymbol{g}\,dv-\int_{\Omega}\boldsymbol{\nabla}p\,dv+\int_{\Omega}\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}'\,dv. \tag{2.34}$$

Following a similar procedure to that of Eq. (2.26) for the left hand side of Eq. (2.34), we obtain

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\boldsymbol{u}\,dv=&\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_0}\rho\boldsymbol{u}J\,dV\\ =&\int_{\Omega_0}\frac{\mathrm{d}(\rho\boldsymbol{u}J)}{\mathrm{d}t}\,dV\\ =&\int_{\Omega_0}\left(J\rho\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}+J\boldsymbol{u}\frac{\mathrm{d}\rho}{\mathrm{d}t}+\rho\boldsymbol{u}\frac{\mathrm{d}J}{\mathrm{d}t}\right)dV\\ =&\int_{\Omega_0}J\rho\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}+\underbrace{J\boldsymbol{u}\left(\frac{\partial\rho}{\partial t}+\boldsymbol{\nabla}\rho\cdot\boldsymbol{u}\right)+J\boldsymbol{u}\rho\boldsymbol{\nabla}\cdot\boldsymbol{u}}_{\text{zero due to the conservation of mass}}\,dV\\ =&\int_{\Omega}\rho\left[\frac{\partial\boldsymbol{u}}{\partial t}+(\boldsymbol{\nabla}\boldsymbol{u})\boldsymbol{u}\right]dv.\end{aligned} \tag{2.35}$$

If the conservation of mass is not substituted into the above equation (2.35), an alternative formula for the conservation of linear momentum is

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\boldsymbol{u}\,dv=\int_{\Omega}\left[\frac{\partial}{\partial t}(\rho\boldsymbol{u})+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\otimes\boldsymbol{u})\right]dv. \tag{2.36}$$

By realising that the above equation (2.34) holds for any spatial domain Ω, the following local forms of the conservation of the linear momentum equation can be obtained[2]

$$\rho\left[\frac{\partial\boldsymbol{u}}{\partial t}+(\boldsymbol{\nabla}\boldsymbol{u})\boldsymbol{u}\right]=\rho\boldsymbol{g}-\boldsymbol{\nabla}p+\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}' \tag{2.37a}$$

$$\frac{\partial}{\partial t}(\rho\boldsymbol{u})+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\otimes\boldsymbol{u})=\rho\boldsymbol{g}-\boldsymbol{\nabla}p+\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}'. \tag{2.37b}$$

Eq. (2.37a) and (2.37b) are referred as the non-conservative and conservative forms of the linear momentum equation, respectively.

[2]In some of the literature available, the convective term in Eq. (2.37a) can alternatively be written as $(\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}=(\boldsymbol{\nabla}\boldsymbol{u})\boldsymbol{u}$.

2.2.4 Conservation of angular momentum

Let us consider the rotational equilibrium of a general body under the action of traction and body forces. This implies

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}(\boldsymbol{x}\times\rho\boldsymbol{u})\,dv=\int_{\Omega}(\boldsymbol{x}\times\rho\boldsymbol{g})\,dv+\int_{\partial\Omega}(\boldsymbol{x}\times\boldsymbol{\sigma}\boldsymbol{n})\,da \tag{2.38}$$

Let $\boldsymbol{\mathcal{E}}$ be the third-order alternating tensor, where $\mathcal{E}_{ijk}=1$ if the permutation of $\{i,j,k\}$ is even, $\mathcal{E}_{ijk}=-1$ if the permutation is odd and 0 if any indices are repeated. The cross product $\times$ between two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ to render a vector $\boldsymbol{c}$ defined as $\boldsymbol{c}_i=(\boldsymbol{a}\times\boldsymbol{b})_i=\mathcal{E}_{ijk}a_jb_k$. The left hand side of equation (2.38) yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}(\boldsymbol{x}\times(\rho\boldsymbol{u}))_i\,dv&=\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_0}(\boldsymbol{x}\times(\rho\boldsymbol{u}))_iJ\,dV\\&=\int_{\Omega}(\boldsymbol{x}\times\frac{\partial}{\partial t}(\rho\boldsymbol{u}))_i\,dv+\int_{\Omega}(\boldsymbol{x}\times\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u}\otimes\boldsymbol{u}))_i\,dv\end{aligned} \tag{2.39}$$

$$\begin{aligned}\int_{\partial\Omega}(\boldsymbol{x}\times\boldsymbol{\sigma}\boldsymbol{n})_i\,da&=\int_{\partial\Omega}\mathcal{E}_{ijk}x_j\sigma_{kl}n_l\,da\\&=\int_{\Omega}\frac{\partial}{\partial x_l}\mathcal{E}_{ijk}x_j\sigma_{kl}\,dv\\&=\int_{\Omega}\mathcal{E}_{ijk}x_j\frac{\partial\sigma_{kl}}{\partial x_l}\,dv+\int_{\Omega}\mathcal{E}_{ijk}\sigma_{kj}\,dv\\&=\int_{\Omega}(\boldsymbol{x}\times\boldsymbol{\nabla}\cdot\boldsymbol{\sigma})_i\,dv+\int_{\Omega}(\boldsymbol{\mathcal{E}}:\boldsymbol{\sigma}^T)_i\,dv.\end{aligned} \tag{2.40}$$

Substitution of Eq. (2.39) and (2.40) into Eq. (2.38), yields after consideration of the linear momentum equation[3]

$$\int_{\Omega}\boldsymbol{\mathcal{E}}:\boldsymbol{\sigma}^T\,dv=\boldsymbol{0}. \tag{2.41}$$

Or in index notation:

$$\boldsymbol{\mathcal{E}}:\boldsymbol{\sigma}^T=\begin{bmatrix}\sigma_{32}-\sigma_{23}\\\sigma_{13}-\sigma_{31}\\\sigma_{21}-\sigma_{12}\end{bmatrix}=\boldsymbol{0}. \tag{2.42}$$

This clearly implies the symmetry of the Cauchy stress tensor $\boldsymbol{\sigma}$, namely $\boldsymbol{\sigma}=\boldsymbol{\sigma}^T$.

2.2.5 Principle of virtual work

Let $\delta\boldsymbol{u}$ denote an arbitrary velocity field defined in the current position of the continuum. The virtual work, δW, can be expressed by multiplying the linear

[3] $\boldsymbol{A}:\boldsymbol{B}=A_{ij}B_{ij}$ is defined as the inner product between two second order tensors $\boldsymbol{A}$ and tensor $\boldsymbol{B}$.

momentum equation with the corresponding conjugate virtual velocity field

$$\delta W = \int_\Omega \left(\rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u} \right] - \rho \boldsymbol{g} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \right) \cdot \delta \boldsymbol{u} \, dv = 0. \tag{2.43}$$

Application of the Gauss theorem in the above Eq. (2.43) yields

$$\delta W = \int_\Omega \rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u} \right] \cdot \delta \boldsymbol{u} \, dv - \int_{\partial\Omega} \boldsymbol{t} \cdot \delta \boldsymbol{u} \, da + \int_\Omega \boldsymbol{\sigma} : \boldsymbol{\nabla} \delta \boldsymbol{u} \, dv - \int_\Omega \rho \boldsymbol{g} \cdot \delta \boldsymbol{u} \, dv. \tag{2.44}$$

Finally, expressing the virtual velocity gradient in terms of the symmetric virtual rate of the deformation $\delta \boldsymbol{d} = \frac{1}{2}(\boldsymbol{\nabla} \delta \boldsymbol{u} + \boldsymbol{\nabla} \delta \boldsymbol{u}^T)$, the spatial virtual work equation with dynamic, internal and external virtual work contributions can be expressed as

$$\delta W = \underbrace{\int_\Omega \rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u} \right] \cdot \delta \boldsymbol{u} \, dv}_{\delta W^{dyn}} + \underbrace{\int_\Omega \boldsymbol{\sigma} : \delta \boldsymbol{d} \, dv}_{\delta W^{int}} - \underbrace{\int_\Omega \rho \boldsymbol{g} \cdot \delta \boldsymbol{u} \, dv + \int_{\partial\Omega} \boldsymbol{t} \cdot \delta \boldsymbol{u} \, da}_{\delta W^{ext}}. \tag{2.45}$$

As seen in Eq. (2.45), the summation of the dynamic, internal and external virtual work contribution equals to zero

$$\delta W = \delta W^{dyn} + \delta W^{int} - \delta W^{ext} = 0. \tag{2.46}$$

2.2.6 Other stress measures

From Eq. (2.44) it can be deduced that pairs such as $\boldsymbol{\sigma}$ and $\boldsymbol{d}$ are said to be work conjugate with respect to the current deformed volume, as their inner product yields work per unit of current volume. For completeness, alternative work conjugate pairs of stresses and strain rates are presented when dealing with work per unit of undeformed volume. The Kirchhoff stress tensor, first Piola-Kirchhoff stress tensor and second Piola-Kirchhoff stress tensor are listed below with their corresponding strain conjugate measures.

Kirchhoff stress tensor

The internal virtual work can be expressed in term of the Kirchhoff stress tensor $\boldsymbol{\tau}$ as

$$\delta W^{int} = \int_{\Omega_0} \boldsymbol{\tau} : \delta \boldsymbol{d} \, dV; \quad \boldsymbol{\tau} = J \boldsymbol{\sigma}. \tag{2.47}$$

First Piola-Kirchhoff stress tensor

The internal virtual work can be expressed in term of the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ as

$$\delta W^{int} = \int_{\Omega_0} \boldsymbol{P} : \delta \dot{\boldsymbol{F}} \, dV^4; \quad \boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-T}. \tag{2.48}$$

Second Piola-Kirchhoff stress tensor

The internal virtual work can be expressed in term of the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ as

$$\delta W^{int} = \int_{\Omega_0} \boldsymbol{S} : \delta \dot{\boldsymbol{E}} \, dV; \quad \boldsymbol{S} = \boldsymbol{F}^{-1}\boldsymbol{P} = J\boldsymbol{F}^{-1}\boldsymbol{\sigma}\boldsymbol{F}^{-T}. \tag{2.49}$$

In this thesis, the Cauchy stress tensor will be preferred for the derivation of formulas. The other alternative stress measures will mainly be used for comparison purposes.

2.3 Constitutive equations

In order to close the coupled system of momentum and mass conservation equations, a constitutive law has to be considered. In the following section, two constitutive models are considered, namely a Newtonian fluid and a hyperelastic solid.

2.3.1 Fluid constitutive equation

For a large class of fluids, the stress-strain constitutive law can be written in the form

$$\boldsymbol{\sigma} = \boldsymbol{f}(\boldsymbol{\nabla u}). \tag{2.50}$$

A Newtonian viscous fluid is a special case for which the stress-strain law is linear as

$$\boldsymbol{\sigma} = \mu(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T) + c\boldsymbol{I}, \tag{2.51}$$

where μ is the dynamics viscosity coefficient and c is a positive material parameter.

For thermodynamical reasons, c is of the form $\frac{2}{3}\mu\boldsymbol{\nabla}\cdot\boldsymbol{u} - p$, where $p = \frac{1}{3}\operatorname{tr}\boldsymbol{\sigma}$ is the pressure[5]. In addition, for an incompressible continuum, the velocity field must be solenoidal $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$, which implies that $\boldsymbol{\nabla}\cdot(\boldsymbol{\nabla u})^T = \boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u}) = \boldsymbol{0}$. This leads to the simplification of the divergence of deviatoric part of stress tensor as

$$\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}' = \boldsymbol{\nabla}\cdot(\boldsymbol{\sigma} + p\boldsymbol{I}) = \mu\Delta\boldsymbol{u}, \tag{2.52}$$

[4]The term $\dot{\boldsymbol{F}}$ represents the material time derivative of $\boldsymbol{F}$, where $\dot{\boldsymbol{F}} = \frac{d\boldsymbol{F}}{dt}$.

[5]The pressure p is considered to be positive when in compression and negative when in tension, as it is standard in the context of fluid dynamics.

where $\Delta = \nabla \cdot \nabla$ is the Laplacian operator. With the expression of $\boldsymbol{\sigma}'$ in Eq. (2.52) and the consideration of appropriate Dirichlet and Neumann boundary conditions in $\partial\Omega_D$ and $\partial\Omega_N$, respectively[6], the complete mass and linear momentum system can be summarised as follows

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\nabla \boldsymbol{u})\boldsymbol{u}\right] = -\nabla p + \nabla \cdot \boldsymbol{\sigma}' + \rho \boldsymbol{g} \qquad \text{in } \Omega \times [0,T] \tag{2.53a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \qquad \text{in } \Omega \times [0,T] \tag{2.53b}$$

$$\nabla \cdot \boldsymbol{\sigma}' = \mu \Delta \boldsymbol{u} \qquad \text{in } \Omega \times [0,T] \tag{2.53c}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \partial\Omega_D \times [0,T] \tag{2.53d}$$

$$\boldsymbol{\sigma n} = \bar{\boldsymbol{t}} \qquad \text{on } \partial\Omega_N \times [0,T] \tag{2.53e}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{in } \bar{\Omega} \times 0 \tag{2.53f}$$

which represents the strong form of the problem governed by the so-called Navier-Stokes equations in Eulerian setting.

2.3.2 Solid constitutive equations

Recall that the first Piola-Kirchhoff stress tensor and Cauchy stress tensor are related via[7]

$$\boldsymbol{P} = J\boldsymbol{\sigma}\boldsymbol{F}^{-T}. \tag{2.54}$$

Most of the classical textbooks in solid mechanics start deriving constitutive models from the consideration of Piola-Kirchhoff stresses. In this thesis, we will break this common practice.

For an incompressible neo-Hookean (NH) model, the deviatoric part of the Cauchy stress tensor is expressed as

$$\boldsymbol{\sigma}' = GJ^{-5/3}\left[\boldsymbol{b} - \frac{1}{3}\operatorname{tr}(\boldsymbol{b})\boldsymbol{I}\right], \tag{2.55}$$

where G represents the shear modulus of the material. For a hyperelastic model, a Helmholtz free energy functional exists, and is defined per unit of undeformed volume. This strain (or stored) energy functional can be conveniently decomposed into the summation of deviatoric $\psi'(J^{-1/3}\boldsymbol{F})$ and volumetric components $\psi^v(J)$ as

$$\psi(\boldsymbol{F}) = \psi'(J^{-1/3}\boldsymbol{F}) + \psi^v(J). \tag{2.56}$$

The definition of the first Piola-Kirchhoff stress tensor

$$\boldsymbol{P} = \frac{\partial \psi}{\partial \boldsymbol{F}} \tag{2.57}$$

[6] $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$ and $\partial\Omega_D \cap \partial\Omega_N = \varnothing$.

[7] This equation can elegantly be proven by means of a pull-back operation as $\int_{\partial\Omega} \boldsymbol{\sigma n}\, da = \int_{\partial\Omega_0} \boldsymbol{\sigma H N}\, dA = \int_{\partial\Omega_0} \underbrace{J\boldsymbol{\sigma}\boldsymbol{F}^{-T}}_{\boldsymbol{P}} \boldsymbol{N}\, dA$.

leads to

$$\boldsymbol{P} = \boldsymbol{P}' + \boldsymbol{P}^v; \quad \boldsymbol{P}' = \frac{\partial \psi'}{\partial \boldsymbol{F}}; \quad \boldsymbol{P}^v = \frac{\partial \psi^v}{\partial \boldsymbol{F}}. \tag{2.58}$$

The volumetric stress term can be further simplified by introducing the pressure p,

$$\boldsymbol{P}^v = \frac{\partial \psi^v}{\partial J}\frac{\partial J}{\partial \boldsymbol{F}} = pJ\boldsymbol{F}^{-T}; \quad p = \frac{d\psi^v(J)}{dJ}. \tag{2.59}$$

The simplest model considered here is the nearly incompressible neo-Hookean (NH) material. Its deviatoric and volumetric parts are given by

$$\psi' = \frac{1}{2}G\left[J^{-2/3}(\boldsymbol{F} : \boldsymbol{F}) - 3\right]; \quad \psi^v = \frac{1}{2}\kappa(J-1)^2. \tag{2.60}$$

Here, κ is the bulk modulus and G is the shear modulus already defined. Substitution of Eq. (2.60) into Eq. (2.58), yields

$$\boldsymbol{P}' = GJ^{-2/3}\left[\boldsymbol{F} - \frac{1}{3}(\boldsymbol{F} : \boldsymbol{F})\boldsymbol{F}^{-T}\right] \tag{2.61}$$

and by applying the relation between the first Piola-Kirchhoff stress tensor and the Cauchy stress tensor in Eq. (2.54), we have

$$\boldsymbol{\sigma}' = GJ^{-5/3}\left[\boldsymbol{b} - \frac{1}{3}\mathrm{tr}(\boldsymbol{b})\boldsymbol{I}\right]. \tag{2.62}$$

Alternatively, the Cauchy stress tensor can be directly derived from a strain energy functional. Note that the time derivative of $\boldsymbol{b}$ is

$$\dot{\boldsymbol{b}} = \dot{\boldsymbol{F}}\boldsymbol{F}^T + \boldsymbol{F}\dot{\boldsymbol{F}}^T = \boldsymbol{l}\boldsymbol{b} + \boldsymbol{b}\boldsymbol{l}^T; \quad \boldsymbol{l} = \boldsymbol{\nabla}\boldsymbol{u} \tag{2.63}$$

and therefore the internal energy rate per unit of undeformed volume $\dot{\psi}$ is given by

$$\dot{\psi} = \frac{\partial \psi}{\partial \boldsymbol{b}} : \dot{\boldsymbol{b}} = \frac{\partial \psi}{\partial \boldsymbol{b}} : (\boldsymbol{l}\boldsymbol{b} + \boldsymbol{b}\boldsymbol{l}^T) = 2\frac{\partial \psi}{\partial \boldsymbol{b}}\boldsymbol{b} : \boldsymbol{l}. \tag{2.64}$$

Combining this equation with the fact the $\boldsymbol{\sigma}$ is work conjugate to $\boldsymbol{l}$ with respect to the current volume and $J^{-1}\dot{\psi} = \boldsymbol{\sigma} : \boldsymbol{l}$, we obtain

$$J\boldsymbol{\sigma} = 2\frac{\partial \psi}{\partial \boldsymbol{b}}\boldsymbol{b} \tag{2.65}$$

leading again to the same expression for the deviatoric component of the Cauchy stress

$$\boldsymbol{\sigma}' = GJ^{-5/3}\left[\boldsymbol{b} - \frac{1}{3}\mathrm{tr}(\boldsymbol{b})\boldsymbol{I}\right] \tag{2.66}$$

With the expression of $\boldsymbol{\sigma}'$ at hand, the complete set of mass and linear momentum

system can be formulated as

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{\nabla u})\boldsymbol{u}\right]=-\boldsymbol{\nabla}p+\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}'+\rho\boldsymbol{g} \qquad \text{in } \Omega\times[0,T] \qquad (2.67a)$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0 \qquad \text{in } \Omega\times[0,T] \qquad (2.67b)$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}'=\boldsymbol{\nabla}\cdot\left[GJ^{-5/3}[\boldsymbol{b}-\frac{1}{3}\mathrm{tr}(\boldsymbol{b})\boldsymbol{I}]\right] \qquad \text{in } \Omega\times[0,T] \qquad (2.67c)$$

$$\boldsymbol{u}=\bar{\boldsymbol{u}} \qquad \text{on } \partial\Omega_D\times[0,T] \qquad (2.67d)$$

$$\boldsymbol{\sigma n}=\bar{\boldsymbol{t}} \qquad \text{on } \partial\Omega_N\times[0,T] \qquad (2.67e)$$

$$\boldsymbol{u}=\boldsymbol{u}_0 \qquad \text{in } \bar{\Omega}\times 0 \qquad (2.67f)$$

which is the strong form of the governing equations for a neo-Hookean material presented in a spatial description.

2.3.3 Rigid continuum

In this section, the rigid body is unconventionally modelled as a continuum and named 'rigid continuum'. The rigid continuum is a continuum like a fluid or solid, but with no requirement for constitutive modelling. Because of the rigid motion description, the system is closed. The linear momentum equation for a rigid continuum is the same as for a fluid or a structure. In the author's opinion, the deviatoric part of the Cauchy stress behaviour changes from hyperbolic (shear wave propagation) to elliptic character when passing from the deformable to the rigid regime. The kinematic options 1 to 4, defined previously in section 2.1.5, are all necessary and sufficient conditions to describe a rigid body motion.

Weak form of the rigid continuum

The weak formulation Eq. (2.45) holds for any continuum, including the rigid continuum. The Cauchy stress tensor $\boldsymbol{\sigma}$ and the deformation rate tensor $\boldsymbol{d}$ are always work conjugate and their inner product yields work per unit current volume $\int_\Omega \boldsymbol{\sigma} : \delta\boldsymbol{d}\, dv$. These work pairs also include: the Kirchhoff stress tensor and the rate of deformation tensor with respect to the initial volume: $\int_{\Omega_0} \boldsymbol{\tau} : \delta\boldsymbol{d}\, dV$, the first Piola-Kirchhoff stress tensor and the deformation gradient tensor with respect to the initial volume: $\int_{\Omega_0} \boldsymbol{P} : \delta\dot{\boldsymbol{F}}\, dV$ and the second Piola-Kirchhoff stress tensor and the material strain rate tensor with respect to the initial volume: $\int_{\Omega_0} \boldsymbol{S} : \delta\dot{\boldsymbol{E}}\, dV$. In this thesis, the formulation is preferred in spatial coordinates, but a similar weak form can be written in other stress measures.

For any rigid body motion, the internal virtual work is zero, which is also consistent with the velocity constraint option 4 stated in section 2.1.5. For this reason,

we have the weak formulation for the rigid continuum as

$$\int_\Omega \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u}\right] \cdot \delta \boldsymbol{u} \; dv + \int_\Omega \boldsymbol{\sigma} : \delta \boldsymbol{d} \, dv - \int_\Omega \rho \boldsymbol{g} \cdot \delta \boldsymbol{u} \, dv - \int_{\partial\Omega} \boldsymbol{t} \cdot \delta \boldsymbol{u} \, da = \boldsymbol{0}, \tag{2.68a}$$

$$\int_\Omega \delta \boldsymbol{\sigma} : \boldsymbol{d} \, dv = \boldsymbol{0}. \tag{2.68b}$$

The satisfaction of the rigid body kinematics constraints leads to a symmetric tensor valued Lagrange multiplier field, which can be seen as the Cauchy stress tensor.[8]

However, by enforcing any other constraints featuring in Section 2.1.5 through a Lagrangian multiplier approach, a weak formulation can be written and solved [66,69,80]. In the author's opinion, these alternative constraints could also generate a traction vector/Cauchy tensor liked value, but may not have such strong physical interpretation.

Strong form of the rigid continuum

The strong form can be derived from the weak formulation (2.68) above presented. The initial/boundary value problem of the rigid continuum is defined as follows

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u}\right] = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} + \rho \boldsymbol{g} \qquad \text{in } \Omega \times [0,T] \tag{2.70a}$$

$$\boldsymbol{d}(\boldsymbol{u}) = \frac{1}{2}\left(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T\right) = \boldsymbol{0} \qquad \text{in } \Omega \times [0,T] \tag{2.70b}$$

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \qquad \text{on } \partial\Omega_D \times [0,T] \tag{2.70c}$$

$$\boldsymbol{\sigma n} = \bar{\boldsymbol{t}} \qquad \text{on } \partial\Omega_N \times [0,T] \tag{2.70d}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 \qquad \text{in } \bar{\Omega} \times 0 \tag{2.70e}$$

The first equation represents the conservation of momentum of the rigid continuum, and the second equation enforces the rigid kinematics. This system of equations define a constrained equilibrium problem, of a similar structure to the unsteady Stokes equations, which can be solved with a well-established saddle point problem solver [81]. The unknown variables of the rigid continuum equations are represented by the Cauchy stress tensor and the velocity, which is consistent with a deformable

[8] A similar virtual field appear on the general incompressible continuum. With the incompressible constraints, the virtual work will be

$$\int_\Omega \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u}\right] \cdot \delta \boldsymbol{u} \; dv - \int_\Omega p(\boldsymbol{\nabla} \cdot \delta \boldsymbol{u}) \, dv + \int_\Omega \boldsymbol{\sigma}' : \delta \boldsymbol{d} \, dv - \int_\Omega \rho \boldsymbol{g} \cdot \delta \boldsymbol{u} \, dv - \int_{\partial\Omega} \boldsymbol{t} \cdot \delta \boldsymbol{u} \, da = 0, \tag{2.69a}$$

$$\int_\Omega \delta p \, (\boldsymbol{\nabla} \cdot \boldsymbol{u}) \, dv = 0. \tag{2.69b}$$

The pressure term acts as the Lagrange multiplier terms and no constitutive modelling is required for the pressure component of the stress.

solid.

Alternative constraint via HVF projection

The Eq. (2.70b) can be replaced by an alternative rigid constraint. Let us introduce a rigid velocity field projection operator $\mathsf{P}(\boldsymbol{u})$, which projects any arbitrary velocity field into a Helicoidal Vector Field (HVF). Regardless of the kinematic constraint formulation, the rigid motion kinematic constraint becomes

$$\boldsymbol{u} - \mathsf{P}(\boldsymbol{u}) = \boldsymbol{0}. \tag{2.71}$$

Alternative expression of the Cauchy stress tensor

The Eq. (2.70) is a saddle point problem and the Cauchy stress tensor and the strain rate tensor are fully coupled. As a result, there is no explicit formulation for the Cauchy stress tensor in terms of the deformation field. Substitution of the above new kinematic constrain Eq. (2.71) into the spatial derivative of the linear momentum equation (2.70a), after separating the pressure and deviatoric part of the stress tensor yieldings

$$\begin{aligned}
\rho\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u}\right] &= -\boldsymbol{\nabla} p + \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}' + \rho\boldsymbol{g} && \text{in } \Omega \times [0,T] \\
\boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 && \text{in } \Omega \times [0,T] \\
\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}' &= \rho\left[\left(\frac{\partial \mathsf{P}(\boldsymbol{u})}{\partial t}\right) + (\boldsymbol{\nabla}\mathsf{P}(\boldsymbol{u}))\mathsf{P}(\boldsymbol{u})\right] + \boldsymbol{\nabla} p - \rho\boldsymbol{g} && \text{in } \Omega \times [0,T] \\
\boldsymbol{u} &= \bar{\boldsymbol{u}} && \text{on } \partial\Omega_D \times [0,T] \\
\boldsymbol{\sigma}\boldsymbol{n} &= \bar{\boldsymbol{t}} && \text{on } \partial\Omega_N \times [0,T] \\
\boldsymbol{u} &= \boldsymbol{u}_0 && \text{in } \bar{\Omega} \times 0
\end{aligned} \tag{2.72}$$

The above system of equations (2.72) is the new set of equilibrium equations for the rigid body in spatial coordinates. The formulation introduces the divergence of Cauchy stress explicitly and avoids the use to evaulate Lagrange multipliers.

Note that the pressure is separated from the Cauchy stress tensor and only the deviatoric part is considered. The rigid body velocity field constraint is more strict than the incompressibility condition. In other words, if a velocity field is a HVF, then $\frac{1}{2}(\boldsymbol{\nabla u} + (\boldsymbol{\nabla u})^T) = \boldsymbol{0}$, which will guarantee the divergence free vector field $\boldsymbol{\nabla}\cdot\boldsymbol{u} = 0$. As the rigid continuum is incompressible, this leads to a pressure field for the rigid body.

2.4 Conclusion

In this chapter, the balance equations for the general incompressible single-phase continuum have been studied. All the balance equations are introduced in spatial

configuration, and the differences among the governing equations lie on the divergence of each material's deviatoric part of the Cauchy stress tensor $\nabla \cdot \boldsymbol{\sigma}'$. For notational convenience, this term can be defined as $\boldsymbol{f} = \nabla \cdot \boldsymbol{\sigma}'$, where

$$\boldsymbol{f} = \mu \Delta \boldsymbol{u} \qquad \text{fluid} \qquad (2.73a)$$

$$\boldsymbol{f} = \nabla \cdot \left[G J^{-5/3} [\boldsymbol{b} - \frac{1}{3} \operatorname{tr}(\boldsymbol{b}) \boldsymbol{I}] \right] \qquad \text{solid} \qquad (2.73b)$$

$$\boldsymbol{f} = \rho \left[\left(\frac{\partial \mathsf{P}(\boldsymbol{u})}{\partial t} \right) + (\nabla \mathsf{P}(\boldsymbol{u})) \mathsf{P}(\boldsymbol{u}) \right] + \nabla p - \rho \boldsymbol{g} \qquad \text{rigid} \qquad (2.73c)$$

Let us finally revisit the formulations of the general continuum equations introduced in this chapter.

Integral form of linear momentum equation

If we combine Eqs. (2.53) (2.67) and (2.72), the conservation law for a general continuum in an integral form can be written as

$$\int_\Omega \rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\nabla \boldsymbol{u}) \boldsymbol{u} \right] \, dv = \int_\Omega \rho \boldsymbol{g} \, dv - \int_\Omega \nabla p \, dv + \int_\Omega \boldsymbol{f} \, dv, \qquad (2.74)$$

where $\boldsymbol{f}$ is defined in Eq. (2.73).

Weak form of balance equations

In general, numerical methods, such as the Finite Element Method, Finite Volume Method or particle based methods, can be established in terms of a weak form. For this purpose, let $\delta \boldsymbol{u}$ denote an arbitrary virtual velocity field imposed on the current configuration of the body. The virtual work in spatial configuration is then

$$\begin{aligned} 0 = \delta W(\boldsymbol{\Psi}, \delta \boldsymbol{u}) = & \underbrace{\int_\Omega \rho \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\nabla \boldsymbol{u}) \boldsymbol{u} \right] \cdot \delta \boldsymbol{u} \, dv}_{\delta W^{dyn}} - \underbrace{\int_\Omega (\boldsymbol{f} - \nabla p) \cdot \delta \boldsymbol{u}}_{\delta \tilde{W}^{int}} \\ & - \underbrace{\int_\Omega \rho \boldsymbol{g} \cdot \delta \boldsymbol{u} \, dv}_{\delta \tilde{W}^{ext}} . \end{aligned} \qquad (2.75)$$

The variational form or the integral form is preferred in this thesis for describing a continuum.

Strong form of balance equations

The initial/boundary-value problem of an incompressible continuum is defined as to find the velocity $\boldsymbol{u}$ and the stress tensor $\boldsymbol{\sigma}$ (which consists of pressure p and the deviatoric part of the Cauchy stress tensor $\boldsymbol{\sigma}'$) as follows

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{\nabla u})\boldsymbol{u}\right]=-\boldsymbol{\nabla}p+\boldsymbol{f}+\rho\boldsymbol{g} \qquad \text{in } \Omega\times[0,T] \qquad (2.76a)$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0 \qquad \text{in } \Omega\times[0,T] \qquad (2.76b)$$

$$\boldsymbol{u}=\bar{\boldsymbol{u}} \qquad \text{on } \partial\Omega_D\times[0,T] \qquad (2.76c)$$

$$\boldsymbol{\sigma n}=\bar{\boldsymbol{t}} \qquad \text{on } \partial\Omega_N\times[0,T] \qquad (2.76d)$$

$$\boldsymbol{u}=\boldsymbol{u}_0 \qquad \text{in } \bar{\Omega}\times 0 \qquad (2.76e)$$

The strong form of the balance equations is used in both boundary fitted methods and immersed boundary methods in order to describe the Fluid-Structure Interaction problem. In addition, relevant constitutive laws as presented in Eq.(2.73) are needed in order to close the system.

Chapter 3

Unified 'one-phase' formulation for fluid-structure interaction

This chapter will describe the set of conservation laws required to describe the multiphase fluid-structure problem as a unified phase. A conservation law states that a particular property of a physical system does not change as the system evolves over time [82]. Conservation laws include conservation of mass, conservation of linear momentum, conservation of angular momentum, conservation of energy, etc. These conservation equations can be written in a unified form as

$$\frac{\partial \phi}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F}(\phi) = 0, \tag{3.1}$$

where ϕ is the property being preserved and $\mathbf{F}$ is its flux vector. In addition, Eq. (3.1) is complemented by the following jump conditions across the interface defined by the normal vector $\boldsymbol{n}$

$$c\,[\![\phi]\!] = [\![\mathbf{F}(\phi)]\!] \cdot \boldsymbol{n}, \tag{3.2}$$

where c is the normal component of the velocity of the jump, and the jump of the variable ϕ across the interface is defined as $[\![\phi]\!] = \phi_1 - \phi_2$, where ϕ_1 and ϕ_2 are the variable values on each side of the interface.

3.1 Multi-phase governing equations

3.1.1 Interface condition

In addition to the equations presented in the previous chapter, for the Fluid-Structure Interaction or Fluid-Fluid Interaction problems, additional jump conditions need to be considered on the interfaces. Let us consider a two material system filling the domain Ω_1, Ω_2 at time t with an interface Γ. Let us define the velocity on Ω_1 as $\boldsymbol{u}_1$, that on Ω_2 as $\boldsymbol{u}_2$ and the interface Γ moves with a normal velocity $c\boldsymbol{n}$, which is the speed of propagation of the discontinuity, as shown in Figure 3.1.

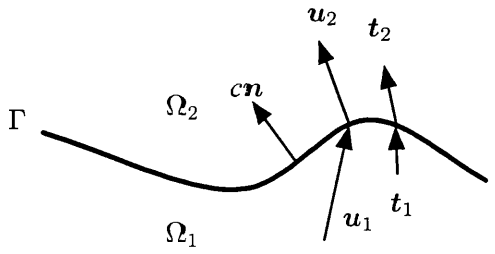


Figure 3.1: Interface conservation condition

Kinematic interface condition The conservation of mass equation is

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u}) = 0. \tag{3.3}$$

Assuming there is no mass change on the interface (mersible, cavitation, fracture, etc), then we have the jump condition on the interface

$$c\,[\![\rho]\!] = [\![\rho \boldsymbol{u}]\!] \cdot \boldsymbol{n}; \quad \rho_1(\boldsymbol{u}_1 \cdot \boldsymbol{n} - c) = \rho_2(\boldsymbol{u}_2 \cdot \boldsymbol{n} - c). \tag{3.4}$$

For incompressibility, the density jump $[\![\rho]\!] = 0$, thus

$$[\![\boldsymbol{u}]\!] \cdot \boldsymbol{n} = 0; \quad \boldsymbol{u}_1 \cdot \boldsymbol{n} = \boldsymbol{u}_2 \cdot \boldsymbol{n} \quad \text{on} \quad \Gamma. \tag{3.5}$$

To satisfy the jump condition on the interface, there is no restriction on the tangential velocity components. However, in most of the cases, a non-slip condition is imposed, which will be assumed throughout the rest of the thesis. Then the kinematic interface condition is

$$[\![\boldsymbol{u}]\!] = \boldsymbol{0}; \quad \boldsymbol{u}_1 = \boldsymbol{u}_2 \quad \text{on} \quad \Gamma. \tag{3.6}$$

Dynamic interface condition The conservation of linear momentum is

$$\frac{\partial (\rho \boldsymbol{u})}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} - \boldsymbol{\sigma}) = \rho \boldsymbol{g} \tag{3.7}$$

with jump condition

$$c\,[\![\rho \boldsymbol{u}]\!] = [\![\rho \boldsymbol{u} \otimes \boldsymbol{u} - \boldsymbol{\sigma}]\!]\,\boldsymbol{n} \quad \text{on} \quad \Gamma. \tag{3.8}$$

In case of incompressibility, using the kinematic interface conditions Eq. (3.6), the dynamic interface condition on the interface becomes

$$[\![\boldsymbol{\sigma}]\!]\,\boldsymbol{n} = \boldsymbol{0}; \quad \boldsymbol{\sigma}_1 \boldsymbol{n} = \boldsymbol{\sigma}_2 \boldsymbol{n} \quad \text{on} \quad \Gamma. \tag{3.9}$$

Eq. (3.9) is called the dynamic interface condition.

3.1.2 Boundary fitted formulation

The boundary fitted method is the most commonly used technique to model fluid-structure interaction problems. If the boundary is explicitly expressed as Γ, the boundary equilibrium can be directly enforced [6, 83]. In the boundary fitted description, there is a list of equations to be solved for the two-phase system.

Let us consider a system comprising in incompressible fluid and solid, occupying the domains Ω_f and Ω_s, respectively, at time t, with interface Γ. The following differential form of the balance equations emerge as [1]

$$\rho_f \frac{\mathrm{d}\boldsymbol{u}_f}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'_f + \rho_f \boldsymbol{g} \qquad \text{in } \Omega_f, \tag{3.10a}$$

$$\rho_s \frac{\mathrm{d}\boldsymbol{u}_s}{\mathrm{d}t} = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'_s + \rho_s \boldsymbol{g} \qquad \text{in } \Omega_s, \tag{3.10b}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}_f = 0 \qquad \text{in } \Omega_f, \tag{3.10c}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u}_s = 0 \qquad \text{in } \Omega_s, \tag{3.10d}$$

$$\boldsymbol{u}_f = \boldsymbol{u}_s \qquad \text{on } \Gamma, \tag{3.10e}$$

$$\boldsymbol{t}_s = \boldsymbol{\sigma}_s \boldsymbol{n}; \quad \boldsymbol{t}_f = \boldsymbol{\sigma}_f \boldsymbol{n}; \quad \boldsymbol{t}_s = \boldsymbol{t}_f \qquad \text{on } \Gamma. \tag{3.10f}$$

Eqs. (3.10a) and (3.10b) represent the conservation law of the linear momentum for the fluid and structure, respectively; Eqs. (3.10c) and (3.10d) are the conservation law of the mass for fluid and structure, and Eqs. (3.10e) and (3.10f) are the kinematic and dynamic boundary conditions, respectively.

This set of Eqs. (3.10) can be solved iteratively or monolithically [84]. If the solid and fluid subproblem are solved iteratively, an efficiently transmission of the variables on the interface is helpful for a better convergence. To achieve this, two Robin-Robin boundary conditions are usually employed, $(\boldsymbol{u}_f - \boldsymbol{u}_s) + \alpha_{1,2}(\boldsymbol{t}_s - \boldsymbol{t}_f) = 0$, where $0 < \alpha_{1,2} < \infty$ [85].

Note that in traditional boundary fitted formulations, primary variables for describing a solid are the displacement field and stress tensor [84] while the primary variables for describing a rigid body are the velocity field and external force [2], rather than the velocity field and stress tensor in Eqs. (3.10). It can be observed that the velocity/stress based formulation is convenient for the numerical computation. In order to reach to an equilibrium between the interfaces, numerical computation of the velocity (time derivative of the displacement) and the traction (spatial derivative of the force) will lose one order of the accuracy. However, the Eqs. (3.10) do not have this problem.

[1] The additional interface kinematic equations will be discussed in Section 3.3.

3.2 Unified 'one-phase' framework

3.2.1 Formulation

In this section, the fundamental equations of this thesis will be presented, namely the unified 'one-phase' formulation. Let us consider a system with multiple incompress-

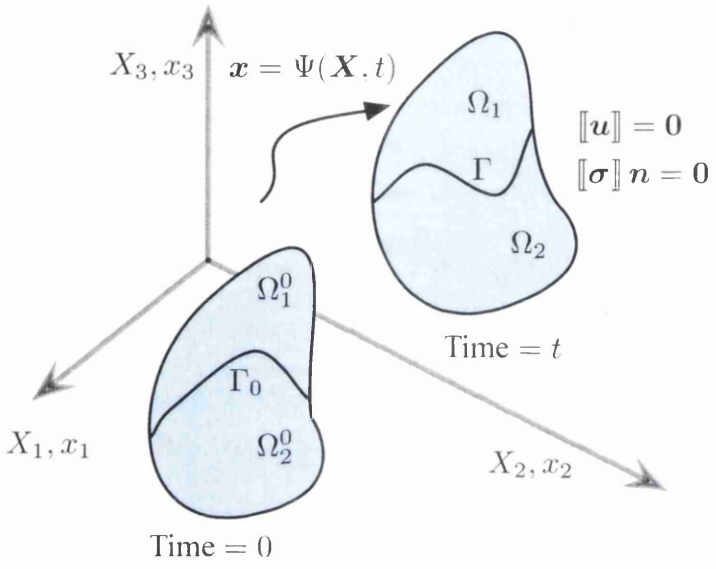


Figure 3.2: 'one-phase' deformation mapping

ible materials. Let $a = 1, ..., n$ indicate fluid, solid or rigid body phases, occupying spatial subdomains Ω_a, such that $\cup_{a=1}^{n} \Omega_a = \Omega$ and $\cap_{a=1}^{n} \Omega_a = \varnothing$, i.e. they fill an arbitrary domain Ω.

The conservation equation for the linear momentum for this $n-$phase system can be expressed in a single integral equation for an arbitrary domain Ω[2]

$$\int_{\Omega} \rho(\boldsymbol{\chi}) \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u} \right] dv = -\int_{\Omega} \boldsymbol{\nabla} p \, dv + \int_{\Omega} \boldsymbol{f}(\boldsymbol{\chi}) \, dv + \int_{\Omega} \rho(\boldsymbol{\chi}) \boldsymbol{g} \, dv \tag{3.11}$$

where $\rho(\boldsymbol{\chi}) = \sum_a \rho_a \chi_a$ and $\boldsymbol{f}(\boldsymbol{\chi}) = \sum_a \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'(\chi_a)$. In these expressions, χ_a is the indicator function for a phase a, which is defined as

$$\chi_a(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} \in \Omega_a \\ 0 & \text{if } \boldsymbol{x} \notin \Omega_a \end{cases} . \tag{3.12}$$

With the indicator function presented, the interface is identified via the sharp change from one value to the other. As discussed in the previous chapter, the divergence of the deviatoric part of the Cauchy stress tensor (in the various possible cases) is

[2]The integral formulation is used to avoid defining the discontinuous indicator function as a distribution.

defined as

$$\boldsymbol{f} = \begin{cases} \mu \Delta \boldsymbol{u} & \text{fluid} \\ \boldsymbol{\nabla} \cdot \left[G J^{-5/3} [\boldsymbol{b} - \frac{1}{3} \operatorname{tr}(\boldsymbol{b}) \boldsymbol{I}] \right] & \text{solid} \\ \rho \left[\left(\frac{\partial \mathsf{P}(\boldsymbol{u})}{\partial t} \right) + (\boldsymbol{\nabla} \mathsf{P}(\boldsymbol{u})) \mathsf{P}(\boldsymbol{u}) \right] + \boldsymbol{\nabla} p - \rho \boldsymbol{g} & \text{rigid} \end{cases} . \tag{3.13}$$

The mass conservation equation of the $n-$phase system is the same as that of the single-phase system in integral form for an arbitrary domain Ω as

$$\int_{\Omega} \boldsymbol{\nabla} \cdot \boldsymbol{u} \, dv = 0. \tag{3.14}$$

Eqs. (3.11)-(3.14) gather the proposed 'one-phase' formulation which holds for any arbitrary domain Ω. This formulation is named 'one-phase' because the governing equations are similar to those of the single phase N-S equations. In a nutshell, the formulation has the following properties

- This formulation can be understood as a re-expression of the multi-phase equations into a single 'one-phase' equation presented in integral form. The use of the integral formulation is preferred is because the stress and velocity on the interface may not be smooth to have derivative formulation and it is preferred not to introduce the indicator function as a generalised function or a distribution. The advantages of this treatment is that it allows the use of existing N-S solvers for the solution of the multiple phase coupled problem.

- The formulation relies on an indicator function, in contrast with the Immersed Boundary Method (IBM), which is based on the Dirac delta distribution. In fact, the indicator function can be constructed in different ways, and one of them is to re-express it via the Dirac delta distribution.

Numerical methods are often established in terms of a weak form[3]. Let $\delta \boldsymbol{u}$ and δp denote a virtual velocity and pressure fields established on the current configuration of the domain. Then, Eqs. (3.11) and (3.14) can be written as

$$\int_{\Omega} \delta \boldsymbol{u} \cdot \rho(\boldsymbol{\chi}) \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla} \boldsymbol{u}) \boldsymbol{u} \right] \, dv = - \int_{\Omega} \delta \boldsymbol{u} \cdot \boldsymbol{\nabla} p \, dv + \int_{\Omega} \delta \boldsymbol{u} \cdot \boldsymbol{f}(\boldsymbol{\chi}) \, dv + \int_{\Omega} \delta \boldsymbol{u} \cdot \rho(\boldsymbol{\chi}) \boldsymbol{g} \, dv, \tag{3.15a}$$

$$\int_{\Omega} \delta p \, (\boldsymbol{\nabla} \cdot \boldsymbol{u}) \, dv = 0. \tag{3.15b}$$

This formulation will be mainly used for an alternative derivation of the discrete immersed solid force from an energy potential in Section 5.1.4. Although the FVM

[3]Finite Element based methods are built upon a weak form, while Finite Volume Methods are based on the divergence theorem, which convert a volume integral into a surface integral. It is worthwhile to point out that FVMs can be re-interpreted as residual based methods where the virtual velocity field is considered piecewise constant element wise [86,87].

is preferred and used later on, the proposed framework is not limited to this discretisation technique.

3.2.2 Specified applications

The proposed unified 'one-phase' formulation is capable of dealing with the multiple phase coupled problem. In order to compare with other existing methods, a series of well-established problems are discussed individually.

Two-phase flow problem

For the two-phase flow problem, using the fluid constitutive model, the unified 'one-phase' formulation (3.11) becomes

$$\int_\Omega \rho(\boldsymbol{\chi}) \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u}\right] \, dv = -\int_\Omega \boldsymbol{\nabla} p \, dv + \int_\Omega \mu(\boldsymbol{\chi}) \Delta \boldsymbol{u} \, dv + \int_\Omega \rho(\boldsymbol{\chi}) \boldsymbol{g} \, dv \quad (3.16)$$

where $\mu(\boldsymbol{\chi}) = \sum_a \mu_a \chi_a$. The above equation can be identified with the so-called 'one-fluid' equation [47, 88] of the two-phase flow problem, except ignoring surface tension effects[4]. In this model, the two phases are treated as a single fluid with variable material properties that change abruptly at the phase interfaces, without resorting to jump conditions. An extra kinematic interface problem needs to be solved, which will be discussed in Section 3.3.

Fluid-Structure Interaction problem: from the IBM to the 'one-phase' formulation

For Fluid-Structure Interaction problems, the unified 'one-phase' formulation (3.11) remains the same except for the forcing term which is defined as

$$\boldsymbol{f} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}' = \begin{cases} \mu \Delta \boldsymbol{u} & \text{in } \Omega_f \\ \boldsymbol{\nabla} \cdot \left(GJ^{-5/3}(\boldsymbol{b} - \frac{1}{3}\operatorname{tr}(\boldsymbol{b})\boldsymbol{I})\right) & \text{in } \Omega_s \end{cases} \quad (3.17)$$

which is related to the pioneering Immersed Boundary Method [3, 57, 58]. Immersed Boundary Methods are based on the use of the Dirac delta distribution. The Dirac delta distribution[5] can be defined as zero everywhere except at 0, written as

$$\delta(\boldsymbol{x}) = \begin{cases} \infty & \text{if } \boldsymbol{x} = \boldsymbol{0} \\ 0 & \text{if } \boldsymbol{x} \neq \boldsymbol{0} \end{cases} \quad (3.18)$$

[4] To account for the surface tension, it is necessary to add an extra term to the equation, which will be introduced in Section 4.2.3.

[5] A comprehensive definition of the Dirac delta distribution can be found in [89].

which also satisfies

$$\int_{\Omega} \delta(\boldsymbol{x})\, dv = 1; \quad \int_{\Omega} f(\boldsymbol{x})\delta(\boldsymbol{x}_0 - \boldsymbol{x})\, dv = f(\boldsymbol{x}_0). \tag{3.19}$$

In the Immersed Boundary Method [3,57,58], an approximation of the Dirac delta distribution is used to evaluate the values over irregular domains. Using this technique, the IBM does not require body/boundary fitted meshes. Consider an incompressible solid material Ω_s immersed in the fluid Ω_f at time t where $\Omega_s \subset \Omega_f$. Then the conservation of linear momentum equation can be formulated as

$$\rho_f \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u}\right] = -\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'_f + \boldsymbol{f} + \rho_f \boldsymbol{g} \qquad \text{in } \Omega_f \tag{3.20a}$$

$$\boldsymbol{f}(\boldsymbol{x}, t) = \mathcal{S}(\boldsymbol{f}(\boldsymbol{x}_s, t)) = \int_{\Omega_s} \boldsymbol{f}(\boldsymbol{x}_s, t)\delta(\boldsymbol{x} - \boldsymbol{x}_s)\, dv \qquad \text{in } \Omega_s \tag{3.20b}$$

$$\boldsymbol{u}(\boldsymbol{x}_s, t) = \mathcal{I}(\boldsymbol{u}(\boldsymbol{x}, t)) = \int_{\Omega} \boldsymbol{u}(\boldsymbol{x}, t)\delta(\boldsymbol{x}_s - \boldsymbol{x})\, dv \qquad \text{in } \Omega \tag{3.20c}$$

Eq. (3.20a) matches the single phase Navier-Stokes linear momentum equation, with an extra body force $\boldsymbol{f}$ which represents the presence of any deformable solid. The function $\boldsymbol{f}$ depends on the constitutive model. In the case of a hyperelastic material, this can be derived from a strain energy potential [62].

During the early development of the IBM, the governing linear momentum equation was written as Eq. (3.20). However, as can be observed in Eq. (3.20), the Dirac delta distribution is only used for the interpolation $\mathcal{I}(\boldsymbol{u}(\boldsymbol{x}, t))$ and spreading $\mathcal{S}(\boldsymbol{f}(\boldsymbol{x}_s, t))$ rather than for continuum modelling of $\boldsymbol{f}$. In fact, the structure force $\boldsymbol{f}$ can be expressed without the Dirac delta distribution. Following the work in [4], the immersed body force $\boldsymbol{f}$ can be written as

$$\boldsymbol{f}(\boldsymbol{x}, t) = -(\rho_s - \rho_f)\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + \boldsymbol{\nabla} \cdot (\boldsymbol{\sigma}'_s - \boldsymbol{\sigma}'_f) + (\rho_s - \rho_f)\boldsymbol{g}, \quad \text{in } \Omega_s. \tag{3.21}$$

Eq. (3.21) is the fundamental governing equation of the IBM for FSI problems. From Eq.(3.21), we can observe that the immersed solid force consists of the following two parts. One part is comprised of the difference between structure and fluid constitutive stresses

$$\boldsymbol{f}^1(\boldsymbol{x}, t) = \boldsymbol{\nabla} \cdot (\boldsymbol{\sigma}'_s - \boldsymbol{\sigma}'_f). \tag{3.22}$$

For most FSI problems, $\boldsymbol{f}^1$ limits the use of explicit time stepping schemes, particularly for the case of large shear modulus of immeresed structures. Another part is the inertia difference due to non-matching densities

$$\boldsymbol{f}^2(\boldsymbol{x}, t) = -(\rho_s - \rho_f)\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + (\rho_s - \rho_f)\boldsymbol{g}. \tag{3.23}$$

The original IBM was originally developed for haemodynamic problems, where the term $\boldsymbol{f}^2$ is always small or zero. For hydrodynamic applications, however, the lim-

itation of explicit time-stepping caused by large density ratios renders the original IBM unstable in practice.

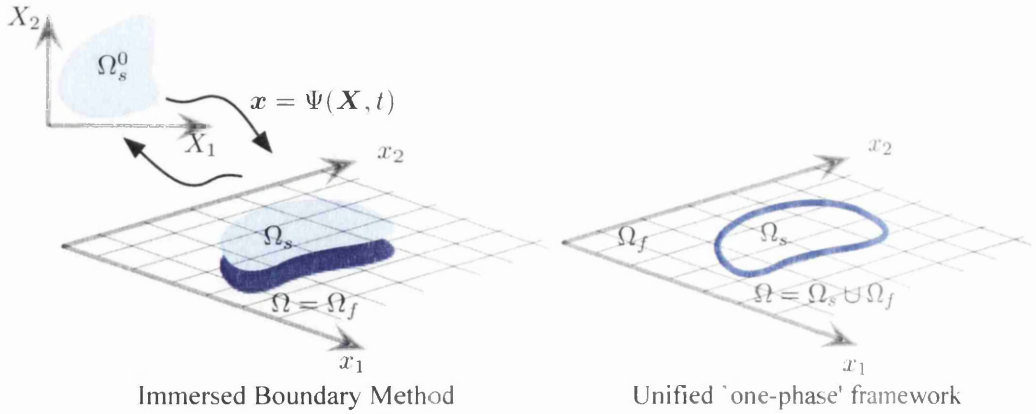


Figure 3.3: Comparison of the Immersed Boundary Methods and the unified 'one-phase' framework for the Fluid-Structure Interaction problem.

Figure 3.3 highlights the differences between the IBM and the unified 'one-phase' framework. As shown in the figure, the fluid domain Ω_f in the IBM occupies the whole domain, which means the solid is 'carried' by the fluid. In contrast, the fluid and solid domain occupies their own domain in the proposed 'one-phase' method. This indicates that the FSI coupling in the IBM happens in the whole solid domain (which consist of constitutive and inertia differences) whilst in the 'one-phase' formulation, the FSI coupling happens on the (smeared) interface. This makes it comparable with boundary fitted based methods where the coupling effect occurs on the (sharp) interface.

In addition, the IBM calculates the stress measures (Kirchhoff stress, first Piola-Kirchhoff stress, second Piola-Kirchhoff stress) in the material configuration, and then performs a push forward operation into the spatial configuration. In the proposed method, the formulation is completely Eulerian (by making solely use of the Cauchy stress). Table 3.1 lists several key points summarising the main aspects of these techniques.

	Classical IBM	Unified 'one-phase' formulation
Mathematical basis	Dirac delta distribution δ	Indicator function χ
Techniques	Interpolation and spreading	Integration
Coupling region	Whole structure domain	On the interface
Immersed force	$-(\rho_s - \rho_f)\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + (\rho_s - \rho_f)\boldsymbol{g}$, $\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}'_s$ and $-\boldsymbol{\nabla}\cdot\boldsymbol{\sigma}'_f$	$\int_{\partial\Omega_s} \boldsymbol{\sigma}'\boldsymbol{n}\, da$
Coordinates	Fluid: spatial; structure: material	Fully spatial
Density ratio	Constrained to be near 1 for the explicit method	Arbitrary

Table 3.1: Comparisons of the classical Immersed Boundary Method (IBM) and the current unified 'one-phase' formulation.

Fluid and rigid body coupling problem

Let us consider the Fluid-Structure Interaction problem between a fluid and a rigid body. Then the unified 'one-phase' formulation (3.11) remains the same except for the forcing term which is defined as

$$\boldsymbol{f} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}' = \begin{cases} \mu \Delta \boldsymbol{u} & \text{in} \quad \Omega_f \\ \rho \left[\frac{\partial \mathsf{P}(\boldsymbol{u})}{\partial t} + (\boldsymbol{\nabla} \mathsf{P}(\boldsymbol{u})) \mathsf{P}(\boldsymbol{u}) \right] + \boldsymbol{\nabla} p - \rho \boldsymbol{g} & \text{in} \quad \Omega_r \end{cases} \tag{3.24}$$

where Ω_r and Ω_f represent the rigid body region and the fluid region, respectively. Figure 3.4 illustrates the differences between the proposed methodology and alternative techniques. In classical rigid body coupling formulations, the coupling can be achieved through a sharp interface (boundary fitted method) or a smoothed interface (immersed boundary method). But regardless of which technique is utilised, the rigid body description is obtained by solving Classical rigid Mechanics equations, namely

$$M \frac{d\boldsymbol{u}_c}{dt} = M\boldsymbol{g} + \boldsymbol{f}_c \tag{3.25a}$$

$$\boldsymbol{J} \frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} = \boldsymbol{t}_c \tag{3.25b}$$

where $\boldsymbol{f}_c, \boldsymbol{t}_c, \boldsymbol{J}, M$ denote the external force, the external torque, the moment of inertia tensor and the mass, respectively.

The coupling force in the classical modelling appears through the torque $\boldsymbol{t}_c$ and the external force $\boldsymbol{f}_c$, whilst in the proposed method, the coupling is achieved via the balanced traction vector on the interface.

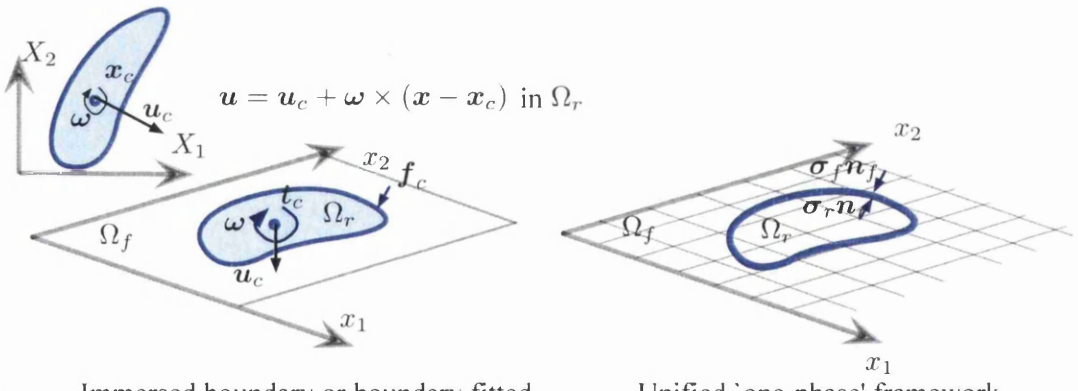


Figure 3.4: Illustration of Lagrangian Mechanics coupling and unified 'one-phase' framework for the fluid and rigid body Interaction problem

Note that for the rigid-fluid coupling problem, the treatment of the rigid body as a different phase of fluid can be found in the literature, often referred to as Distributed Lagrangian method [69,90], rigid-fluid method [73], direct-forcing fictitious-

domain [75], etc. These methods share some algorithmic similarities by considering some form of artificial body force in order to constraint the velocity field on the discrete level to be that of a rigid body, but none of these methods have an explicit expression of the immersed force on a continuum level.

Conclusions

Table 3.2 compares the proposed unified 'one-phase' formulation with boundary fitted methods [1, 2, 56], the classical Immersed Boundary Method [57–60] and the Immersed Structure Potential Method [4, 5] for the general multiphase coupling problem.

3.3 Description of the interface

This section is fundamentally independent of previous sections of this thesis and deals with the description and evolution of the interface. The section begins with description of the kinematics of the interface, where two different approaches are well established: Lagrangian tracking [45–47] and Eulerian capturing [39–42]. This section describes the Level Set Method approach [42] in detail, and finally, it provides a multiple level set approach for the description of multiple interfaces.

Many natural phenomena are examples of evolving interfaces, where there is a clear interface between one phase and another, as shown in Figure 3.5. The ability to describe the interface accurately is the cornerstone of studying these phenomena. Generally speaking, the interface can be described explicitly or implicitly [88,91]. In

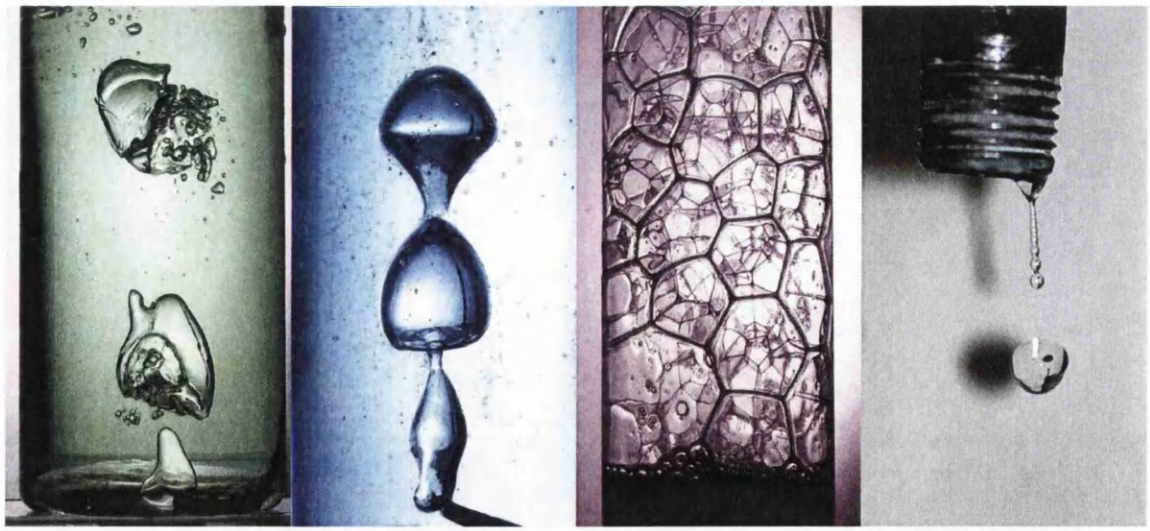


Figure 3.5: Photographs of interfaces, photo courtesy of Andrew Davidhazy.

explicit methods, the interface is explicitly described via a parametric representation [91]. For example, most Computer Aided Design (CAD) programs represent interfaces explicitly through the use of Non-Uniform Rational B-Spline (NURBS) [92].

	BFM	Classical IBM/ ISPM	Unified 'one-phase'
Methodology	Use of interface condition to transfer information.	Rely on a Dirac delta distribution δ for interpolation and spreading.	Based on indicator function χ.
Working coordinates	Fluid is often in spatial coordinates and structure is often in material coordinates.	Mixed Eulerian-Lagrangian	Fully Eulerian configuration.
Density ratio capability	Easy for large density ratio problems, but encounter added mass effects for similar densities.	Only capable to deal with density ratios near to 1 with explicit time-stepping.	Arbitrary density ratio problems with explicit time-stepping.
Interface description	Sharp interface with conforming meshes.	Dirac delta distribution.	Indicator function, which can be constructed from Dirac delta distribution, Level Set function, VOF, etc.
Coupling region	On the sharp interface	In the whole solid domain.	On the smoothed interface.
Immersed structure force	None	Kirchoff stress; inertia term; fluid viscous term.	Cauchy stress.
Computational cost	Expensive especially in 3D.	Inexpensive.	Inexpensive.
Multiphase flow	ALE/re-meshing	None.	Capable.
FSI	ALE/re-meshing.	Capable	Capable.
Rigid body coupling	ALE/re-meshing for the fluid; Lagrangian rigid body dynamics for the rigid body.	None.	Capable.
Applications	Wide arrange of engineering applications, but it is not suitable for cardiac dynamics simulation which involve large deformation and same density ratio.	Ideal for cardiac dynamics, but limited to one-phase, similar density ratio and deformable solids.	All range of applications, bubble-dynamics, cardiac dynamics and hydrodynamics problems.

Table 3.2: Comparisons of the boundary fitted method, classical Immersed Boundary Method and Immersed Structural Potential Method with the current Unified 'one-phase' formulation. For notational convenience, some abbreviations are introduced: BFM - boundary fitted methods; classical IBM - classical Immersed Boundary Methods; ISPM - Immersed Structural Potential Method.

Specifically, an interface Γ_t as a two-dimensional manifold embedded in $\mathbb{R}^3$ and defined parametrically via a mapping $\boldsymbol{r}(r_1, r_2) : \mathbb{R}^2 \to \mathbb{R}^3$ as

$$\Gamma_t = \{\boldsymbol{X} \in \mathbb{R}^3 \mid \boldsymbol{X} = \boldsymbol{X}(\boldsymbol{r}, t), \boldsymbol{r} \in \mathbb{R}^2\}. \tag{3.26}$$

Alternatively, implicit methods often describe the interface using a contour-based algorithm, as typically used in Magnetic Resonance Imaging (MRI)/computed tomography (CT) scans [93]. The interface Γ_t is implicitly defined by a marker function ϕ.

3.3.1 Lagrangian treatment of an interface

One straightforward way of representing interfaces is by means of the use of Lagrangian particles [45–47]. The Lagrangian description makes it easy to preserve the mass and convect the particles. The convection of a particle is achieved by integrating in time the ordinary differential equation

$$\frac{\mathrm{d}\boldsymbol{x}_s}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{x}_s), \tag{3.27}$$

where $\boldsymbol{u}(\boldsymbol{x}_s)$ is the velocity associated with each particle.

3.3.2 Eulerian treatment of an interface

There are two main families of Eulerian methods [88]: the continuous Level Set Method (LSM) and the discontinuous Volume Of Fluid (VOF) method. The difference lies in how the marker function ϕ is defined and convected. The marker function can be regarded as a signed distance function in the case of Level Set Method or a volume fraction in the case of Volume Of Fluid method. To enhance the understanding of the implicit marker function, some examples in the one-dimensional setting are shown below.

Level Set Method

The basic Level Set Method was developed by Osher and Sethian [94] and further developed by Sussman, Smereka, and Osher [13] for multiphase flow simulations. The function ϕ is defined as a signed distance function, and the contour $\phi = 0$ defines the interface (See Figure 3.6).

$$\Gamma_t = \{\boldsymbol{x} \in \mathbb{R}^3 \mid \phi(\boldsymbol{x}, t) = 0\}, \tag{3.28}$$

$$\phi(\boldsymbol{x}) = \operatorname{sgn} d(\boldsymbol{x}, \Gamma_t), \tag{3.29}$$

where $d(\boldsymbol{x}, \Gamma_t)$ denotes the distance to Γ_t.

Volume Of Fluid

The volume fraction ϕ is defined to be a piecewise constant function from 0 to 1. The interface can be postprcessed as the contour corresponding to a value 0.5 after

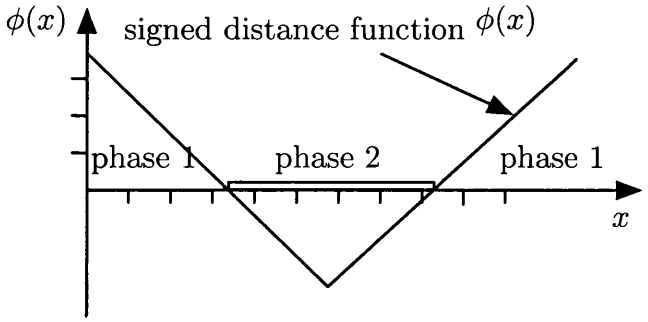


Figure 3.6: One-dimensional illustration of the Level Set Method.

application of a reconstruction procedure $f(\boldsymbol{x})$[6]. In one dimension, Figure 3.7 shows a fraction volume ϕ representing the interface.

$$\Gamma_t = \{\boldsymbol{x} \in \mathbb{R}^3 \mid f(\phi(\boldsymbol{x}, t)) = 0.5\}. \tag{3.30}$$

The Level Set Method is used in this work because of its advantages over others.

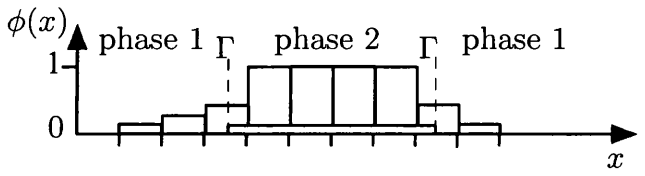


Figure 3.7: One-dimensional illustration of the Volume-Of-Fluid method.

In particular, it is easy to compute the normal $\boldsymbol{n}$ and curvature κ of the interface from

$$\boldsymbol{n} = \frac{\nabla\phi}{||\nabla\phi||}; \quad \kappa = -\nabla \cdot \left(\frac{\nabla\phi}{||\nabla\phi||}\right). \tag{3.31}$$

Another advantage of the Level Set Method is that topological changes are automatically handled, as shown in Figure 3.8. On the left, the two contours represent two distinct circular blobs, but on the right the blobs have been merged into one.

The zero Level Set of a continuous function $\phi(\boldsymbol{x}, t), \boldsymbol{x} \in \mathbb{R}^{2,3}$ represents an interface $\Gamma \subset \mathbb{R}^{2,3}$. Hence, the evolution equation of an interface moving in a medium with velocity $\boldsymbol{u}$, is defined in an Eulerian setting in non-conservative form as

$$\frac{\mathrm{d}\phi(\boldsymbol{x}, t)}{\mathrm{d}t} = \frac{\partial\phi(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{u}(\boldsymbol{x}, t) \cdot \nabla\phi(\boldsymbol{x}, t) = 0. \tag{3.32}$$

[6]Reconstruction of the interface from the discontinuous volume fraction is important for the post-processing and the evaluation of the curvature on the interface. These techniques include SLIC [95], PLIC [41], etc.

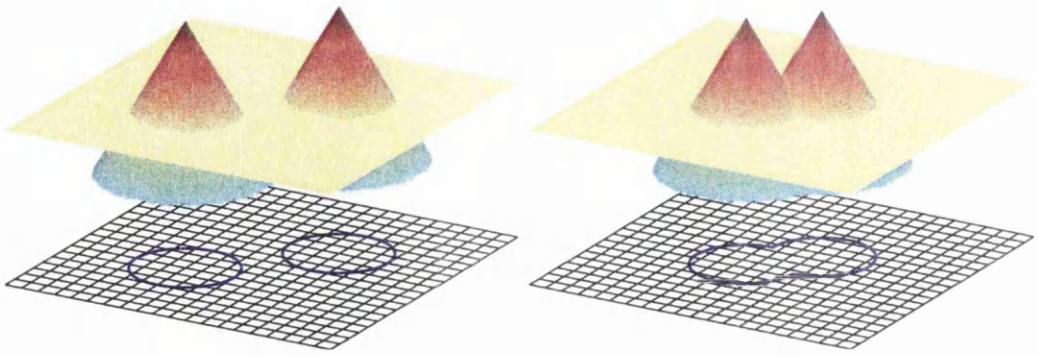


Figure 3.8: Level Set representation of an interface. The interface consists of two distinct circles on the left, but on the right the circles are closer and form one interface. The Level Set function is constructed as a distance function.

As the fluid is incompressible $\nabla \cdot \boldsymbol{u}(\boldsymbol{x},t) = 0$, above Eq. (3.32) can be re-written as

$$\begin{aligned} \frac{\mathrm{d}\phi(\boldsymbol{x},t)}{\mathrm{d}t} &= \frac{\partial\phi(\boldsymbol{x},t)}{\partial t} + \boldsymbol{u}(\boldsymbol{x},t)\cdot\nabla\phi(\boldsymbol{x},t) + \phi(\boldsymbol{x},t)\nabla\cdot\boldsymbol{u}(\boldsymbol{x},t) \\ &= \frac{\partial\phi(\boldsymbol{x},t)}{\partial t} + \nabla\cdot(\phi(\boldsymbol{x},t)\boldsymbol{u}(\boldsymbol{x},t)) = 0. \end{aligned} \tag{3.33}$$

The above Level Set equations can be convected using any standard numerical method for the solution of hyperbolic equations [88].

3.3.3 Multiple interfaces

Early work on the Level Set method dealt exclusively with two-phase problems [96,97]. If a two-phase material system S_1 and S_2 with an interface Γ is considered, as discussed earlier, the zero contour lines of the Level Set function are a convenient way to define the interface Γ. The sign of the Level Set can be used to identify each phase:

$$\phi(\boldsymbol{x},t)\begin{cases} > 0 & \boldsymbol{x}\in\Omega_1 \\ < 0 & \boldsymbol{x}\in\Omega_2 \end{cases}. \tag{3.34}$$

Only a few researchers have considered the problem of multiple Level Sets for the representation of multiple regions [98,99]. A multiphase material system S_a with an interface Γ, for each subregion S_a could be represented by

$$\phi_a(\boldsymbol{x},t)\begin{cases} > 0 & \boldsymbol{x}\notin\Omega_a \\ < 0 & \boldsymbol{x}\in\Omega_a \end{cases}. \tag{3.35}$$

Assuming that the multiphase material system is immiscible, then for any $\phi_a(\boldsymbol{x}) < 0$, it can be deduced that $\phi_b(\boldsymbol{x}) > 0$, for all $b \neq a$. The evolution of the Level Set functions ϕ_a is the same as the single-Level Set in a non-conservative form

$$\frac{\partial\phi_a}{\partial t} + \boldsymbol{u}\cdot\nabla\phi_a = 0. \tag{3.36}$$

Multi-Level Set evolution often leads to contradictions at the intersection of several interfaces, where the conditions

$$\cup_{a=1}^{n} \Omega_a = \Omega; \quad \Omega_a \cap \Omega_b = \varnothing, \ \forall a \neq b \tag{3.37}$$

can be violated. A method will then be needed to recover the above conditions (3.37). The idea of the multi-Level Sets method [98] is to design a projection method to

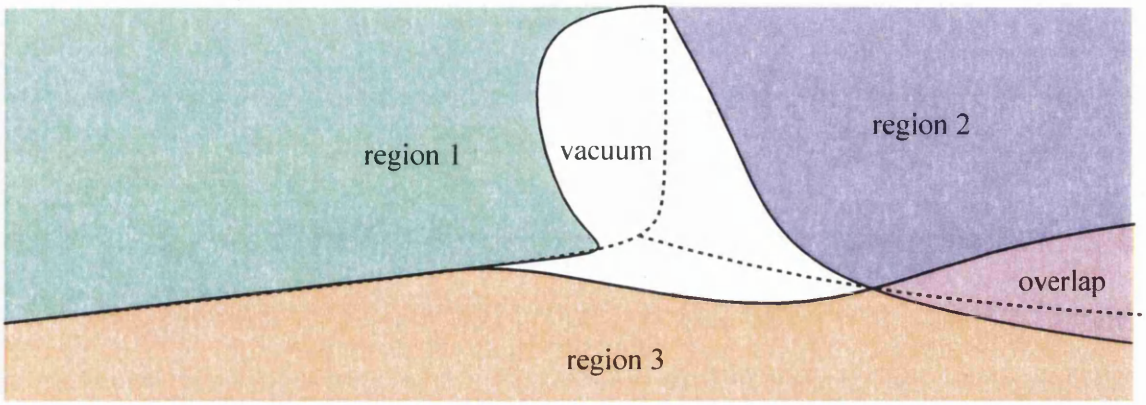


Figure 3.9: Each of the three regions is independently evolved in time, after which the interface locations do not agree. There are vacuums where all ϕ_a are positive, and overlaps where more than one ϕ_a is negative.

correct the multiple Level Sets. Observe that for any region, the interface position is determined by the two smallest Level Set components ϕ_a (negative) and ϕ_b (positive), other values which are larger than ϕ_b are not relevant to the interface[7]. The projection step can be summarised as follows

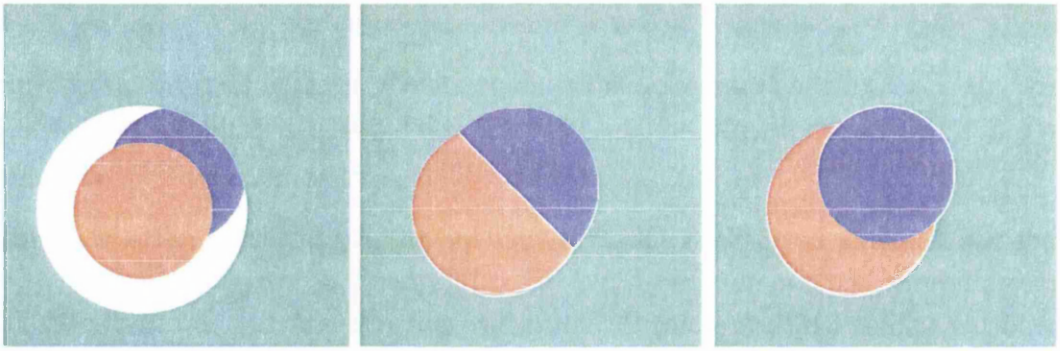
1 If ϕ_a is the smallest element, $-|\phi_a|$ is the only negative element whose magnitude represents the distance to the interface.

2 If condition 1 holds and ϕ_k is the second smallest element, then $\phi_k = |\phi_a|$.

During the evolution of the multiple Level Sets, the correction is assumed for all phases that can be moved (often fluids). If a structure or rigid phase is presented by one Level Set function, the first step to take is fix the level set describing the structure or rigid phases.

Three Level Sets are initialised to represent three phases. As illustrated in Figure 3.10(a), there is vacuum and overlap between these phases. Figure 3.10(b) shows the interfaces after the correction step, the interfaces move to the middle of the vacuum and overlap regions. Figure 3.10(c) shows the interfaces if the circle in the middle cannot be moved, which will be in the case of representing a solid or rigid body.

[7]The observation is easy to understand because the Level Set value ϕ is a signed distance function denoting the distance to the interface. A Level Set component with a large value implies that it is far from the interface.



(a) Initial Multiple Level Set (b) All can be moved (c) One phase fixed

Figure 3.10: Examples for the correction of the interface of multiple Level Sets

Part III

Numerical Techniques

Chapter 4

Immersed multiphase fluid

In this chapter, we present the numerical method employed for the Direct Numerical Simulations (DNS) of the proposed unified 'one-phase' formulation, with a focus on Newtonian fluids. The term 'DNS' is used because there are no modelling issues beyond the continuum hypothesis [88].

Under the constitutive law of Newtonian fluids, the Cauchy stress tensor is linearly related to the symmetric velocity gradient tensor. A Marker-And-Cell based discretisation technique [37] is utilised due to its relative simplicity and efficiency (section 4.1.1). Two convective terms feature on the equations, the velocity convective term $(\boldsymbol{\nabla u})\boldsymbol{u}$ and the Level Set convective term $\boldsymbol{u} \cdot \boldsymbol{\nabla}\phi$, which are both discretised by means of a QUICK scheme [100]. In addition, to uncouple the velocity and pressure field, a classical fractional step method is used (section 4.1.3) [101].

Compared with a typical single-phase N-S solver, two additional steps are required. One is to solve the Level Set equation, which involves convection, reinitialisation (section 4.2.2) and the correction step for the interface of multiple fluids, and the other is the solution of an anisotropic Poisson equation due to the different/non-constant density of the multiphase continuum. The overview of the algorithm is given in Section 4.3.

Note that there are two nonlinearities present in the system, namely the velocity convective term and the geometric nonlinearity introduced by the movement of the interface.

Integral form of balance equation

Recall the proposed unified 'one-phase' formulation (3.11)-(3.14). Together with the Level Set convection Eq. (3.33) and indicator function Eq. (3.12), the a-multiphase flow governing equations can be rewritten in an integral form, for an arbitrary Ω, leading to

$$\int_\Omega \frac{\partial \boldsymbol{u}}{\partial t}\, dv + \int_{\partial\Omega} \boldsymbol{u}(\boldsymbol{u}\cdot\boldsymbol{n})\, da = \frac{\mu(\boldsymbol{\chi})}{\rho(\boldsymbol{\chi})}\int_{\partial\Omega} \boldsymbol{\nabla u}\cdot\boldsymbol{n} da - \frac{1}{\rho(\boldsymbol{\chi})}\int_{\partial\Omega} p\boldsymbol{n} da + \int_\Omega \boldsymbol{g} dv \tag{4.1a}$$

$$\int_\Omega \boldsymbol{\nabla}\cdot\boldsymbol{u}\, dv = 0 \tag{4.1b}$$

$$\int_\Omega \frac{\partial \phi_a}{\partial t} dv + \int_{\partial\Omega} \boldsymbol{u}\phi_a\cdot\boldsymbol{n}\, da = 0 \tag{4.1c}$$

$$\text{where}\quad \chi_a(\boldsymbol{x}) = \begin{cases} 1 & \phi_a(\boldsymbol{x}) < 0 \\ 0 & \phi_a(\boldsymbol{x}) > 0 \end{cases} \tag{4.1d}$$

$$\mu(\boldsymbol{\chi}) = \sum_a \mu_a \chi_a; \tag{4.1e}$$

$$\rho(\boldsymbol{\chi}) = \sum_a \rho_a \chi_a \tag{4.1f}$$

Eqs. (4.1a-4.1b) represent the momentum and mass conservation laws. Eq. (4.1c) is the Level Set transport equation used to track the evolution of the interface corresponding to the a-phase. Eqs. (4.1d) (4.1e) and (4.1f) define the indicator function, the Level Set function and fluid physical properties throughout the entire computational domain. Eq. (4.1d) is the exact indicator function, however, a smoothed indicator function will preferred for computational purposes. Notice that in above Eq.(4.1a), the fluid properties are assumed to be piecewise constant within every control volume Ω.

4.1 Navier-Stokes solver

4.1.1 Cartesian staggered mesh

Particularising for the case of a two dimensional Cartesian staggered mesh, let $\Omega_{u^{A_x}}$ and $\Omega_{v^{A_y}}$ be the control volumes associated with the Cartesian components of the velocity u^{A_x} and v^{A_y}, respectively, with an arrangement similar to that of a Marker-And-Cell (MAC) grid[1] see Figure 4.1. Although methods based on co-located grids have been developed [82], staggered grids schemes result in a very robust numerical method easy to implement. Here, A_x and A_y denote fluid cell edges perpendicular to the OX and OY Cartesian axes, respectively, and u^{A_x} and v^{A_y} their corresponding

[1]Harlow & Welch [102] proposed the use of a special grid for incompressible flow computations. This specially defined grid decomposes the computational domain into cells with velocities defined on the cell faces and scalar pressure defined at cell centres. See [37] for a good review.

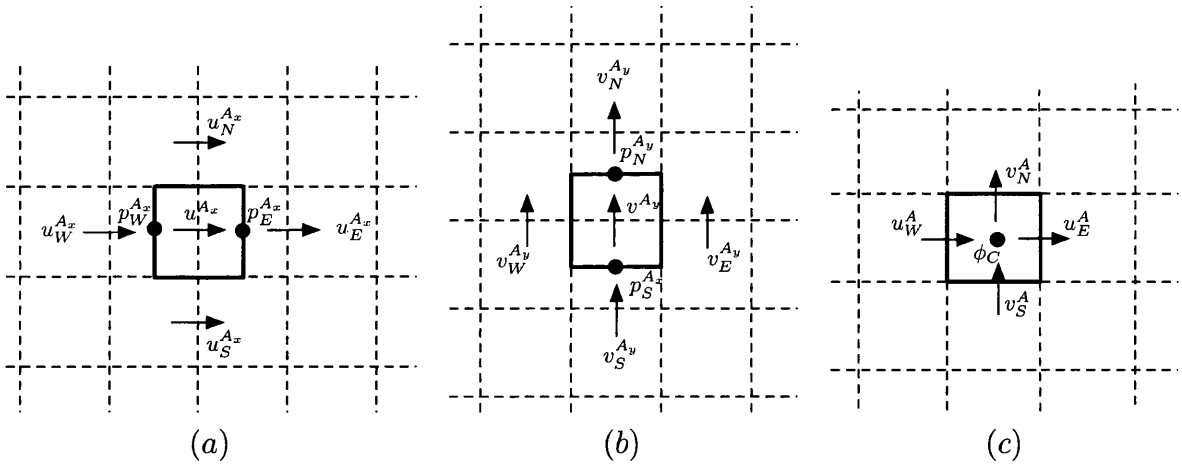


Figure 4.1: The notation used for a standard staggered MAC mesh. (a) Control volume $\Omega_{u^{A_x}}$ for velocity u; (b) Control volume $\Omega_{v^{A_y}}$ for velocity v; (c) Control volume Ω_{ϕ^A} for level set ϕ.

normal edge velocities. In this cases, the conservation of linear momentum reads,

$$\int_{\Omega_{u^{A_x}}} \frac{\partial}{\partial t} u \, dv + \int_{\partial\Omega_{u^{A_x}}} \mathbf{F_u} \cdot \boldsymbol{n} \, da - \int_{\Omega_{u^{A_x}}} \boldsymbol{g} \cdot \boldsymbol{e}_x \, dv = 0, \tag{4.2}$$

$$\int_{\Omega_{v^{A_y}}} \frac{\partial}{\partial t} v \, dv + \int_{\partial\Omega_{v^{A_y}}} \mathbf{F_v} \cdot \boldsymbol{n} \, da - \int_{\Omega_{v^{A_y}}} \boldsymbol{g} \cdot \boldsymbol{e}_y \, dv = 0, \tag{4.3}$$

where $\{\boldsymbol{e}_x, \boldsymbol{e}_y\}$ is the standard Cartesian basis. Eqs. (4.2-4.3) represent the conservation of linear momentum variables $\rho(\boldsymbol{\chi})u$ and $\rho(\boldsymbol{\chi})v$ in a variational integral form with $\mathbf{F_u}$ and $\mathbf{F_v}$ their corresponding interface fluxes, namely,

$$\mathbf{F_u} = u\boldsymbol{u} + \frac{1}{\rho(\boldsymbol{\chi})} p\boldsymbol{e}_x - \frac{\mu(\boldsymbol{\chi})}{\rho(\boldsymbol{\chi})} \boldsymbol{\nabla} u, \quad \mathbf{F_v} = v\boldsymbol{u} + \frac{1}{\rho(\boldsymbol{\chi})} p\boldsymbol{e}_y - \frac{\mu(\boldsymbol{\chi})}{\rho(\boldsymbol{\chi})} \boldsymbol{\nabla} v, \quad \boldsymbol{u} = [u, v]^T \tag{4.4}$$

The convective components of the numerical fluxes, that is $u\boldsymbol{u}$ and $v\boldsymbol{u}$, are obtained using a stabilised convective approximation, such as SMART [103], HLPA [104] or QUICK [100], which minimise numerical diffusion, avoid the creation of spurious oscillations and reduce the total variation of the solution by accounting for the transportive nature of the fluid [82]. According to the above procedure, all the terms appearing in the formulation are evaluated as

Time derivative part of the Level Set field

$$\int_{\Omega_{\phi^A}} \frac{\partial}{\partial t} \phi \, dv \simeq |\Omega_{\phi^A}| \frac{\Delta \phi^A}{\Delta t}. \tag{4.5}$$

Time derivative part of the velocity field

$$\int_{\Omega_{uAx}} \frac{\partial}{\partial t} u \, dv \simeq |\Omega_{uAx}| \frac{\Delta u^{Ax}}{\Delta t}; \tag{4.6}$$

$$\int_{\Omega_{vAy}} \frac{\partial}{\partial t} v \, dv \simeq |\Omega_{vAy}| \frac{\Delta v^{Ay}}{\Delta t}. \tag{4.7}$$

where Δt, $\Delta\phi$ Δx and Δy denote the a time step length $\Delta t = t^{n+1} - t^n$, $\Delta\phi = \phi^{n+1} - \phi^n$, cell length in $x-$direction and cell length in $y-$direction.
The spatial discretisation of the stress component of the numerical fluxes at the interfaces between velocity control volumes are evaluated as

$$-\int_{\partial\Omega_{uAx}} \boldsymbol{\nabla} u \cdot \boldsymbol{n} \, da \simeq -\Delta y \left(\frac{u_E^{A_x} - u^{A_x}}{\Delta x} - \frac{u^{A_x} - u_W^{A_x}}{\Delta x} \right) - \Delta x \left(\frac{u_N^{A_x} - u^{A_x}}{\Delta y} - \frac{u^{A_x} - u_S^{A_x}}{\Delta y} \right) \tag{4.8}$$

$$-\int_{\partial\Omega_{vAy}} \boldsymbol{\nabla} v \cdot \boldsymbol{n} \, da \simeq -\Delta y \left(\frac{v_E^{A_y} - v^{A_y}}{\Delta x} - \frac{v^{A_y} - v_W^{A_y}}{\Delta x} \right) - \Delta x \left(\frac{v_N^{A_y} - v^{A_y}}{\Delta y} - \frac{v^{A_y} - v_S^{A_y}}{\Delta y} \right) \tag{4.9}$$

Pressure term of Eq. (4.1a), see Figure 4.1

$$\int_{\partial\Omega_{uAx}} p \boldsymbol{e}_x \cdot \boldsymbol{n} \, da \simeq \Delta x (p_E^{A_x} - p_W^{A_x}), \tag{4.10}$$

$$\int_{\partial\Omega_{vAy}} p \boldsymbol{e}_y \cdot \boldsymbol{n} \, da \simeq \Delta y (p_N^{A_y} - p_S^{A_y}). \tag{4.11}$$

Gravity term of Eq. (4.1a)

$$\int_{\Omega_{uAx}} \boldsymbol{g} \cdot \boldsymbol{e}_x \, dv \simeq |\Omega_{uAx}| g_x^{A_x}, \tag{4.12}$$

$$\int_{\Omega_{vAy}} \boldsymbol{g} \cdot \boldsymbol{e}_y \, dv \simeq |\Omega_{vAy}| g_y^{A_y} \tag{4.13}$$

where $g_x^{A_x}$ and $g_y^{A_y}$ are the external forces per unit of volume allocated to control volumes $|\Omega_{uAx}|$ and $|\Omega_{vAy}|$, respectively.

The integrals $\int_{\partial\Omega} \boldsymbol{u}(\boldsymbol{u}\cdot\boldsymbol{n})da$ and $\int_{\partial\Omega} \boldsymbol{u}\phi\cdot\boldsymbol{n}da$ are the two convective components. A different convection scheme has been employed for the Level Set $\int_{\partial\Omega} \boldsymbol{u}\phi \cdot \boldsymbol{n}da$ (Appendix B) and 1-D stability and error analysis have been given in Appendix C.

4.1.2 Discretisation of convection term

For the fluid convection term, a central difference scheme was used for all spatial variables in the original MAC method [102]. However, for moderate Reynolds numbers, this approach can render unphysical oscillations. Subsequent implementations

used first-order upwind scheme, leading to excessive numerical dissipation [88]. To overcome these shortcomings, the high order upwind methods have been developed. A well known approach is the Quadratic Upstream Interpolation for Convective Kinetics (QUICK) scheme [100], where values at the cell edges are interpolated via upstream biased third order polynomials. A one-dimensional QUICK scheme is presented in detail in Appendix B.1.2.

The level set convection term, can be advected with any sufficiently accurate scheme standard in the context of hyperbolic problems. Here we choose to write these two convection terms using a QUICK scheme.

Step 1 Obtain the velocity u_W, u_E, v_S, v_N at control volume edges

In control volume $\Omega_u^{A_x}$

$$u_W = \frac{u_W^{A_x} + u^{A_x}}{2}; u_E = \frac{u_E^{A_x} + u^{A_x}}{2}; v_N = \frac{v_{NW}^{A_x} + v_{NE}^{A_x}}{2}; v_S = \frac{v_{SW}^{A_x} + v_{SE}^{A_x}}{2}. \tag{4.14}$$

In control volume $\Omega_v^{A_y}$

$$u_W = \frac{u_{NW}^{A_y} + u_{SW}^{A_y}}{2}; \quad u_E = \frac{u_{NE}^{A_y} + u_{SE}^{A_y}}{2}; \quad v_N = \frac{v_N^{A_y} + v^{A_y}}{2}; \quad v_S = \frac{v_S^{A_y} + v^{A_y}}{2}. \tag{4.15}$$

In control volume Ω_ϕ^A

$$u_W = u_W^A; \quad u_E = u_E^A; \quad v_N = v_N^A; \quad v_S = v_S^A. \tag{4.16}$$

Step 2 Let φ be the quantities to be convected in the upwind direction using second order interpolation. The scheme uses a three-point upstream-weighted quadratic interpolation for cell face values. The face value of φ is obtained from a quadratic function passing through two bracketing nodes (on each side of the face) and a node on the upwind side.

In control volume $\Omega_u^{A_x}$

$$\begin{aligned}
\varphi_W^u &= \begin{cases} \frac{1}{8}(3u^{A_x} + 6u_W^{A_x} - u_{WW}^{A_x}) & \text{if } u_W > 0 \\ \frac{1}{8}(3u_W^{A_x} + 6u^{A_x} - u_E^{A_x}) & \text{if } u_W < 0 \end{cases} \\
\varphi_E^u &= \begin{cases} \frac{1}{8}(3u_E^{A_x} + 6u^{A_x} - u_W^{A_x}) & \text{if } u_E > 0 \\ \frac{1}{8}(3u_{EE}^{A_x} + 6u_E^{A_x} - u^{A_x}) & \text{if } u_E < 0 \end{cases} \\
\varphi_N^u &= \begin{cases} \frac{1}{8}(3u_N^{A_x} + 6u^{A_x} - u_S^{A_x}) & \text{if } v_N > 0 \\ \frac{1}{8}(3u^{A_x} + 6u_N^{A_x} - u_{NN}^{A_x}) & \text{if } v_N < 0 \end{cases} \\
\varphi_S^u &= \begin{cases} \frac{1}{8}(3u^{A_x} + 6u_S^{A_x} - u_{SS}^{A_x}) & \text{if } v_S > 0 \\ \frac{1}{8}(3u_S^{A_x} + 6u^{A_x} - u_N^{A_x}) & \text{if } v_S < 0 \end{cases}
\end{aligned} \tag{4.17}$$

In control volume $\Omega_v^{A_y}$

$$\begin{aligned}
\varphi_W^v &= \begin{cases} \frac{1}{8}(3v^{A_y} + 6v_W^{A_y} - v_{WW}^{A_y}) & \text{if } u_W > 0 \\ \frac{1}{8}(3v_W^{A_y} + 6v^{A_y} - v_E^{A_y}) & \text{if } u_W < 0 \end{cases} \\
\varphi_E^v &= \begin{cases} \frac{1}{8}(3v_E^{A_y} + 6v^{A_y} - v_W^{A_y}) & \text{if } u_E > 0 \\ \frac{1}{8}(3v_{EE}^{A_y} + 6v_E^{A_y} - v^{A_y}) & \text{if } u_E < 0 \end{cases} \\
\varphi_N^v &= \begin{cases} \frac{1}{8}(3v_N^{A_y} + 6v^{A_y} - v_S^{A_y}) & \text{if } v_N > 0 \\ \frac{1}{8}(3v^{A_y} + 6v_N^{A_y} - v_{NN}^{A_y}) & \text{if } v_N < 0 \end{cases} \\
\varphi_S^v &= \begin{cases} \frac{1}{8}(3u^{A_y} + 6u_S^{A_x} - v_{SS}^{A_y}) & \text{if } v_S > 0 \\ \frac{1}{8}(3u_S^{A_y} + 6u^{A_x} - v_N^{A_y}) & \text{if } v_S < 0 \end{cases}
\end{aligned} \tag{4.18}$$

In control volume ϕ^A

$$\begin{aligned}
\varphi_W^\phi &= \begin{cases} \frac{1}{8}(3\phi_C^A + 6\phi_W^A - \phi_{WW}^A) & \text{if } u_W > 0 \\ \frac{1}{8}(3\phi_W^A + 6\phi_C^A - \phi_E^A) & \text{if } u_W < 0 \end{cases} \\
\varphi_E^\phi &= \begin{cases} \frac{1}{8}(3\phi_E^A + 6\phi_C^A - \phi_W^A) & \text{if } u_E > 0 \\ \frac{1}{8}(3\phi_{EE}^A + 6\phi_E^A - \phi_C^A) & \text{if } u_E < 0 \end{cases} \\
\varphi_N^\phi &= \begin{cases} \frac{1}{8}(3\phi_N^A + 6\phi_C^A - \phi_S^A) & \text{if } v_N > 0 \\ \frac{1}{8}(3\phi_C^A + 6\phi_N^A - \phi_{NN}^A) & \text{if } v_N < 0 \end{cases} \\
\varphi_S^\phi &= \begin{cases} \frac{1}{8}(3\phi^A + 6\phi_S^A - \phi_{SS}^A) & \text{if } v_S > 0 \\ \frac{1}{8}(3\phi_S^A + 6\phi_C^A - \phi_N^A) & \text{if } v_S < 0 \end{cases}
\end{aligned} \tag{4.19}$$

4.1.3 Fractional step

The coupled velocity and the pressure field can be solved using a monolithic approach [105]. However, it is also possible and advantageous to segregate [105] the pressure and the velocity. The most well-known methods for this type of approach are fractional step methods, which were developed in an independent manner by Chorin [23] and Temam [101]. In this approach, the velocity is first advanced in time without accounting for the pressure gradient effect, resulting in a field that is in general not divergence free. The pressure is then solved using a Poisson type equation solver and the velocity field is finally corrected by adding the pressure gradient. Traditionally, the fractional step method consists of three stages if we perform time discretisation before spatial discretisation:

Step 1: Compute an intermediate velocity $\boldsymbol{u}^\star$ by advancing explicitly

$$\int_\Omega \frac{\boldsymbol{u}^\star - \boldsymbol{u}^n}{\Delta t}\, dv = -\int_\Omega \mathrm{Conv}(\boldsymbol{u}^{n+\frac{1}{2}})\, dv + \frac{\mu(\boldsymbol{\chi})}{\rho(\boldsymbol{\chi})} \int_{\partial\Omega} \boldsymbol{\nabla u}^n \cdot \boldsymbol{n}\, da, \tag{4.20}$$

where superscript n denotes the variable at the beginning of a time step length

Δt and $n+1$ denotes the new value at the end of step. In this particular case[2], $\text{Conv}(\boldsymbol{u}^{n+\frac{1}{2}})$ is the convective term computed using the Adams-Bashforth formula

$$\text{Conv}(\boldsymbol{u})^{n+\frac{1}{2}} = \frac{3}{2}\text{Conv}(\boldsymbol{u})^{n} - \frac{1}{2}\text{Conv}(\boldsymbol{u})^{n-1}. \tag{4.21}$$

Step 2: Solve the Poisson pressure equation for p^{n+1} to satisfy the incompressibility condition:

$$\int_{\Omega} \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho(\chi)} \right) dv = - \int_{\Omega} \frac{\nabla \cdot \boldsymbol{u}^{\star}}{\Delta t} dv \tag{4.22}$$

with Neumann boundary condition on the domain boundaries $\boldsymbol{n} \cdot \left(\frac{\nabla p}{\rho(\chi)} \right) = 0$.

Step 3: Correct the velocity:

$$\int_{\Omega} \boldsymbol{u}^{n+1} dv = \int_{\Omega} \boldsymbol{u}^{\star} dv - \frac{\Delta t}{\rho(\chi)} \int_{\partial\Omega} p^{n+1} \boldsymbol{n} \, da \tag{4.23}$$

As can be observed, the algorithm described above is completely explicit, and allows us to alleviate the numerical difficulties related to the saddle-point problem. In the first step of the method, the convective and viscous terms in the 'one-phase' equations are treated explicitly using the value of the previous time step.

The second step of the fractional step method deals with solving a non-constant diffusion Poisson Equation (4.22). The MAC discretisation is shown in Figure 4.2, where the pressure and density variables are defined at the centre of the cell. After

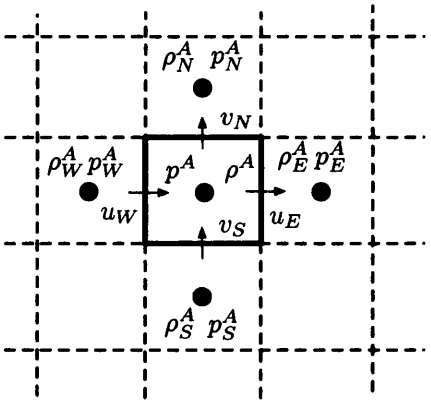


Figure 4.2: The notation used for a standard staggered MAC grid: the pressure control volume.

discretisation with the MAC grid, the discretisation of Equation (4.22) is exactly

[2]There are alternative ways to evaluate the convective term without increasing the nonlinearity of the scheme

the same as the five-point Finite Difference stencil [88]

$$\begin{aligned}&\frac{2}{\Delta x^2}\left(\frac{p_W^A - p^A}{\rho_W^A + \rho^A} - \frac{p^A - p_E^A}{\rho_E^A + \rho^A}\right) + \frac{2}{\Delta y^2}\left(\frac{p_N^A - p^A}{\rho_N^A + \rho^A} - \frac{p^A - p_S^A}{\rho_S^A + \rho^A}\right)\\ &= -\frac{1}{\Delta t}\left(\frac{u_W - u_E}{\Delta x} + \frac{v_N - v_S}{\Delta y}\right).\end{aligned} \tag{4.24}$$

Discretisation at each cell centre leads to a coupled linear system of equations for the unknown pressure. This system is defined via a positive semi-definite matrix that can be solved effectively using the Conjugate Gradient method. The implicit solver procedure is the most time consuming part of the overall algorithm. Estimate the 90% based on our numerical examples. From the implementation standpoint, the *hypre* [106] software library has been used to aid with the iterative solution of the large sparse system of equations.

In the case of dealing with multiple phases with the same density, the Poisson problem to be solved becomes isotropic [88]. The isotropic Poisson equation inspires the use of a Fast Fourier Transform (FFT) based fast Poisson solver [107], well known for its outstanding computational speed.

4.2 Computational aspects

4.2.1 Indicator function

Following the work of Sussman [13], the approximation of an indicator function $\hat{\chi}$ can be calculated as

$$\hat{\chi}(\boldsymbol{x}) = \begin{cases} \frac{1}{2}(1 + \phi/\epsilon + \sin(\pi\phi/\epsilon)/\pi) & \textit{if } |\phi/\epsilon| \leq 1 \\ 0 & \textit{if } \phi/\epsilon < -1 \\ 1 & \textit{if } \phi/\epsilon > 1 \end{cases}, \tag{4.25}$$

where ϵ is a parameter that represents the smearing bandwidth. Thus, the approximation of the indicator function changes from zero to one over the smearing interface, describing a smooth transition zone from one phase to the next. Once the indicator function has been constructed, the various material properties can be assigned. The indicator function is smeared out, so the corresponding density and viscosity field are smooth on the interface region.

In order to understand the above Eq. (4.25), let us consider the computational domain $\Omega = [-1.2, 1.2] \times [-1.2, 1.2]$ and domain Ω_f, which has a shape of sphere, centred in $[0, 0]$ with a radius $r = 1$ immersed in Ω. The Level set function is written as

$$\phi(x, y) = x^2 + y^2 - 1. \tag{4.26}$$

The Figure 4.3 illustrates the the approximation of the indicator function by the Level Set function.

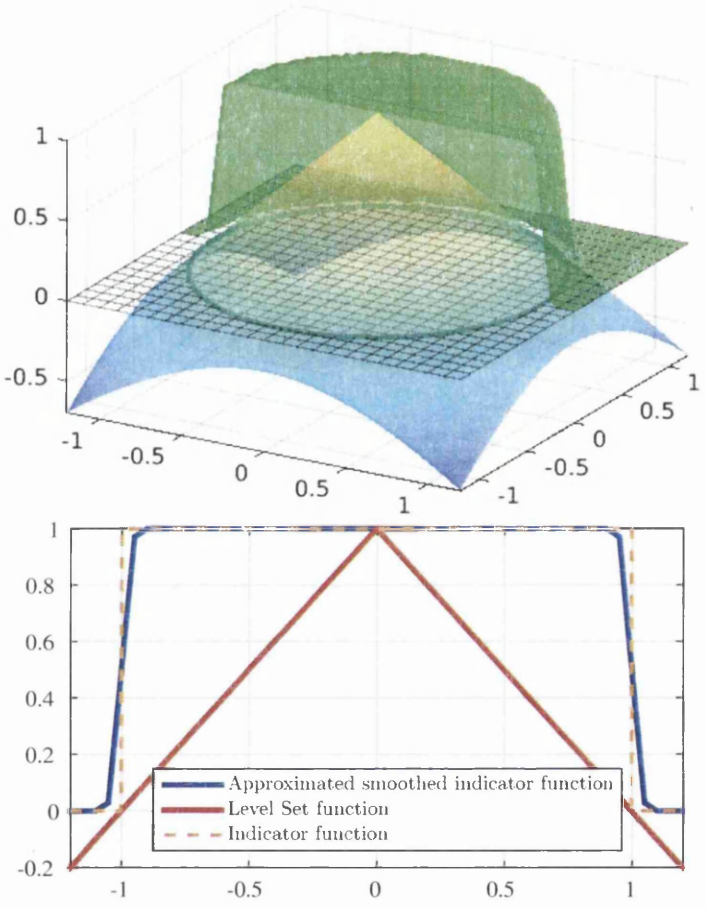


Figure 4.3: The approximation of the indicator function by the Level Set function. Up: The Level Set function and its indicator function. Down: Cross section of the approximated Indicator function.

4.2.2 Level Set reinitialisation

The evolution of the Level Sets Eq. (4.1c) often distorts the Level Set function so that its slope is either too flat or too steep near the interface and hence, loses the property of a signed distance function, that is $||\nabla\phi|| = 1$. A small perturbation of the Level Sets may change the interface location. To recover the distance property, the Level Set often needs to be rebuilt. This process is called reinitialisation. The interface Γ is defined by the Level Set function $\phi : \mathbb{R}^d \to \mathbb{R}$, $\Gamma = \{\boldsymbol{x} \in \mathbb{R}^d \mid \phi(\boldsymbol{x}) = 0\}$, where $d = 2, 3$. The reinitialisation process involves constructing ψ. The following three conditions are satisfied. Find a scalar field ψ such that

$$\begin{aligned} &(1)\ \ \Gamma = \{\boldsymbol{x} \in \mathbb{R}^d \mid \phi(\boldsymbol{x}) = \psi(\boldsymbol{x}) = 0\}; \\ &(2)\ \ ||\nabla\psi|| = 1; \\ &(3)\ \ \mathrm{sgn}(\psi) = \mathrm{sgn}(\phi). \end{aligned} \tag{4.27}$$

Condition 1 ensures that the interface does not change after reinitialisation. Condition 2 and 3 requires that ψ is a signed distance function.

Sussman [13] introduced a reinitialisation equation which turns any Level Set function into a signed distance function, which is first order accurate. Then some second-order or higher order reinitialisation method was proposed in the literature [108–110]. The reinitialisation equation are given by

$$\frac{\partial\phi}{\partial\tau} + \mathrm{sgn}(\phi)(||\nabla\phi|| - 1) = 0 \tag{4.28}$$

where τ is a pseudo time variable. If the equation converges within the pseudo time step, then ϕ will be a signed distance function. Unfortunately, the direct solution of

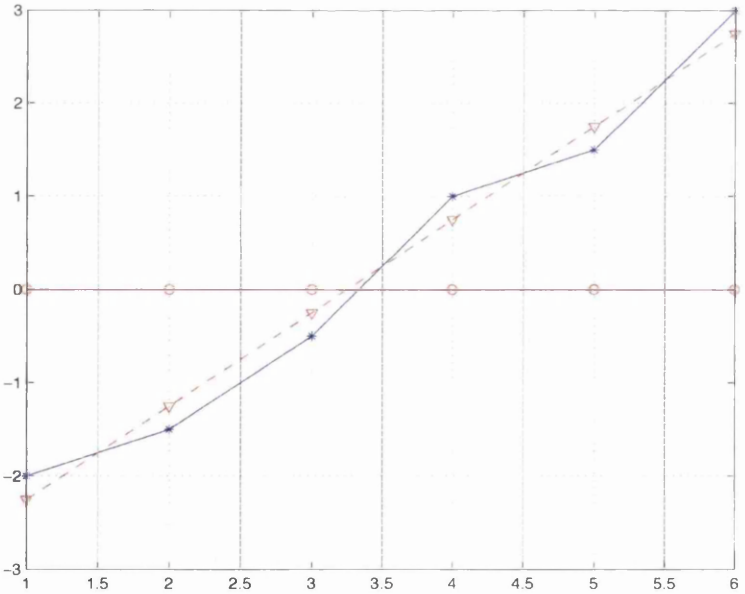


Figure 4.4: 1D test showing how the zero Level Set has moved. The solid line is the initial Level Set, dash line is the re reinitialised Level Set.

the reinitialisation Eq. (4.28) will move the interface, as shown in Figure 4.4. The solid line is the initial Level Set, dash line results after for direct solution of Eq. (4.28).

Fix the interface during the reinitialisation

To preserve the interface, two different regions should be defined and treated separately, as shown in Figure 4.5. The reinitialisation procedure is only performed in

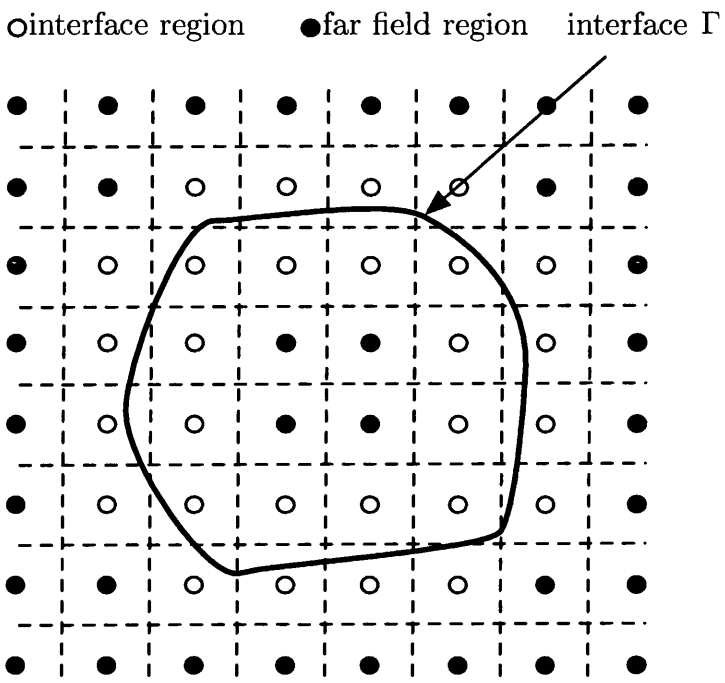


Figure 4.5: Computational domain partitioned in two regions: interface region Ω_Γ(white) and far-field region Ω_{far} (black).

the far-field region Ω_{far} (black), because the interface region Ω_Γ (white) defines the interface Γ. In order to preserve the reinitialisation condition 2, computation of the signed distance function in the interface region is required.

In a one-dimensional setting, Russo and Smereka [111] introduced an approximation of the interface location as

$$D_i = \frac{\phi_i^0}{\Delta\phi_i^0/\Delta x}, \tag{4.29}$$

where $\Delta\phi_i^0 = |\phi_{i+1}^0 - \phi_{i-1}^0|/2$.

In a two dimensional setting, if $\phi_{i,j}\phi_{i-1,j} < 0$ or $\phi_{i,j}\phi_{i+1,j} < 0$ or $\phi_{i,j}\phi_{i,j-1} < 0$ or $\phi_{i,j}\phi_{i,j+1} < 0$ (change of sign of the level set indicates the presence of the interface region), the quantity $D_{i,j}$ represents the distance of node (i,j) from the interface,

which can be computed by

$$D_{i,j} = \frac{\phi_{i,j}}{\sqrt{(\frac{\phi_{i+1,j}-\phi_{i-1,j}}{2\Delta x})^2 + (\frac{\phi_{i,j+1}-\phi_{i,j-1}}{2\Delta y})^2}}. \tag{4.30}$$

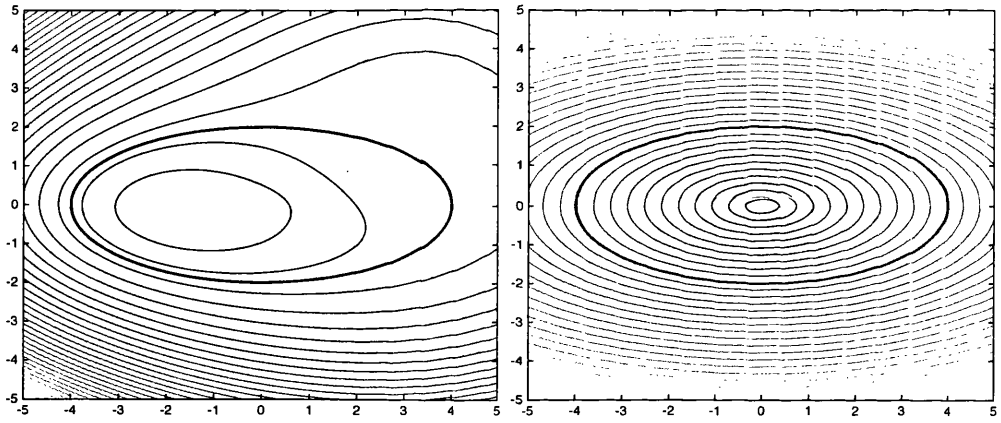
Numerical example on reinitialisation

Let us consider the following Level Set function $\phi(x, y)$ by perturbation of a signed distance function

$$\phi(x, y) = f(x, y)\left(\sqrt{\frac{x^2}{a^2} + \frac{y^2}{b^2}} - 1\right) \tag{4.31}$$

where $f(x, y) = \epsilon + (x - x_0)^2 + (y - y_0)^2$ and $a = 4$, $b = 2$, $\epsilon = 0.1$, $x_0 = 3.5$, $y_0 = 2$. This choice of $\phi(x, y)$ means that the initial condition has both small and large gradients near its zero Level Set.

Before the reinitialisation, the Level Set function is not a distance function, as observed from the contour lines in Figure 4.6(a). After reinitialisation, the Level Set function becomes a distance function in Figure 4.6(b). On the implementation,



(a) Initial state (b) Final state

Figure 4.6: Two dimensional reinitialisation test.

the narrow band Level Set is followed, which is introduced by Adalsteinsson and Sethian [112] by updating the band of the active Level Set area.

4.2.3 Surface tension

Up to now, the two-phase flow system is solved without consideration of surface tension effects on the interfaces. However, accurate computation of surface tension effects is, perhaps, one of the most critical aspects in the numerical simulation of small scale multiphase problems. In methods based on the 'one-phase' formulation, the surface tension is usually added as a body force concentrated in a band around the interface at the discrete level. Normally, the band is $2 - 4$ computational cells

in width and arranged in such a way that the force has a maximum on the interface and decays rapidly with distance from it. Sussman introduced an idea of modelling surface tension by the mean curvature flow to avoid strict time step s [113] [114] and has been since then used by the graphics community [115].

Surface tension can also be applied discontinuously on the interface, typically as a boundary condition on the pressure in the pressure projection step (Immersed Interface Method [116]). In this thesis, we will pursue the first method and, hence, we will model surface tension as a body force.

In this thesis, the presence of surface tension results in an unbalanced force acting on the interface defined as

$$\boldsymbol{f} = \gamma\kappa(\phi)\boldsymbol{n}(\phi)\delta(\phi) \tag{4.32}$$

where κ is the surface curvature, $\boldsymbol{n}$ is the unit vector normal across the interface, δ is the Dirac delta distribution and γ is the surface tension coefficient (physical property). The unit vector normal to interface is calculated as

$$\boldsymbol{n} = \frac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||}, \tag{4.33}$$

whereas curvature is calculated as

$$\kappa = -\boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||} \right). \tag{4.34}$$

As in the original work of Sussman [13], the traditional choice for $\delta(\phi)$ in the Level Set formulation has been the Cosine delta function defined on the nodes as

$$\hat{\delta}(\phi) = \begin{cases} \frac{1}{2\epsilon}(1 + \cos(\pi\phi/\epsilon)) & \text{if} \quad |\phi/\epsilon| \leq 1 \\ 0 & \text{if} \quad |\phi/\epsilon| > 1 \end{cases}. \tag{4.35}$$

4.3 Algorithm

In this section, a complete numerical algorithm for the multiphase flow solver is summarised.

Algorithm 4.1 : MULTIPHASE FLUID SOLVER

1. Initialise the Level Set function ϕ_a as a signed distance function, fluid velocity $\boldsymbol{u}$, density ρ and viscosity $\mu(\boldsymbol{\chi})$ and indicator function χ_a

- Loop over time step n

   1. Compute the surface tension term if necessary (Eq. 4.32)
   2. Compute the intermediate velocity

$$\int_\Omega \frac{\boldsymbol{u}^\star - \boldsymbol{u}^n}{\Delta t}\, dv = -\int_\Omega \mathrm{Conv}(\boldsymbol{u}^{n+\frac{1}{2}})\, dv + \frac{\mu(\boldsymbol{\chi})}{\rho(\boldsymbol{\chi})}\int_{\partial\Omega} \boldsymbol{\nabla}\boldsymbol{u}^n \cdot \boldsymbol{n}\, da$$

   3. Compute the new pressure to satisfy the incompressibility constraint with Neumann boundary condition $\boldsymbol{n}\cdot\left(\frac{\boldsymbol{\nabla} p}{\rho(\boldsymbol{\chi})}\right) = 0$.

$$\int_\Omega \boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla} p^{n+1}}{\rho(\boldsymbol{\chi})}\right) dv = -\int_\Omega \frac{\boldsymbol{\nabla}\cdot\boldsymbol{u}^\star}{\Delta t}\, dv$$

   4. Apply velocity correction

$$\int_\Omega \boldsymbol{u}^{n+1}\, dv = \int_\Omega \boldsymbol{u}^\star\, dv - \frac{\Delta t}{\rho(\boldsymbol{\chi})}\int_{\partial\Omega} (p^{n+1}\boldsymbol{I})\cdot\boldsymbol{n}\, da$$

   5. Compute the Level Set convection equation (section 4.1.1)

$$\int_\Omega \frac{\partial \phi_a^n}{\partial t}\, dv + \int_{\partial\Omega} (\boldsymbol{u}^{n+1}\phi_a^n)\cdot\boldsymbol{n}\, da = 0$$

   6. Reinitialisation of the Level Set (section 4.2.2)

$$\phi_\tau + \mathrm{sgn}(\phi)(||\boldsymbol{\nabla}\phi|| - 1) = 0$$

   7. Update the density $\rho(\boldsymbol{\chi})^{n+1}$ and viscosity $\mu(\boldsymbol{\chi})^{n+1}$ to the next time step

$$\mu(\boldsymbol{\chi})^{n+1} = \sum_a \mu_a \chi_a^{n+1};\quad \rho(\boldsymbol{\chi})^{n+1} = \sum_a \rho_a \chi_a^{n+1}$$

Chapter 5

Immersed deformable solid

Once established the multiphase fluid solver in chapter 4, the difficulty for solving the FSI problem shifts to the evaluation of the immersed deformable structure's Cauchy stress tensor $\boldsymbol{\sigma}$. As discussed in chapter 2, the structure's Cauchy stress is a function of the deformation gradient tensor $\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}}$. However, the deformation gradient tensor involves the use of material spatial variables $\boldsymbol{X}$. To achieve this, a set of material points can be introduced to carry this information.

In order to link the spatial grid computation and the material points immersed information, the Dirac delta distribution is introduced for the 'interpolation' of information from grid to particles and 'spreading' of information in the reverse way. These two operations are called interpolation operation and spreading operation, respectively, which will be discussed in section 5.1.2.

The discrete immersed deformable solid force can be directly written from the Cauchy stress. Also, to be consistent with the Immersed Structure Potential Method [4,5] (which is the starting point of this thesis), the same expression is derived from a structure store energy potential, which is a similar procedure to that of deriving a stress measure [28].

It is worth noting that in the proposed framework, the coupling appears only on the smoothed interface between fluid and structure, whilst in other methods like the ISPM, the coupling is modelled in the whole structure domain. This change in the immersed methodology will remove two parts of the original ISPM force, the dynamic part due to the different density and the viscous part due to the fluid viscosity, which will allow for the explicit solution of problems with arbitrary density ratios, as discussed in section 3.2.2.

A robust fixed point iteration algorithm is used for solving the nonlinear coupled problem, which mainly consists of the interface and the structure's nonlinearity. Here we simply ignore the fluid convection nonlinearity. The convection nonlinearity is a source of turbulent phenomena [29] which is not the main focus of this thesis.

Integral form of the balance equations

Let us review the governing equations discussed in Chapter 3. Consider the continuum domain Ω and a partition into disjoint sets Ω_i that represents each of the possible fluid phases, i.e. $\Omega = \cup_i \Omega_i$, $\Omega_i \cap \Omega_j = \emptyset, i \neq j$. A regularised indicator function χ_a, is evaluated for each phase Ω_a. Such regularisations are constructed so that the partition of unity property that the true characteristic functions satisfy also holds, i.e. $\sum_a \chi_a = 1$. This identity allows us to consider the linear momentum conservation equation for a control volume as

$$\int_\Omega \rho(\boldsymbol{\chi}) \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla u})\boldsymbol{u} \right] dv = -\int_\Omega \boldsymbol{\nabla} p \, dv + \int_\Omega \boldsymbol{f}(\boldsymbol{\chi}) \, dv + \int_\Omega \rho(\boldsymbol{\chi}) \boldsymbol{g} \, dv \qquad (5.1)$$

where $\boldsymbol{\chi}$ is the vector of regularised indicator functions, $\rho(\boldsymbol{\chi})$ is the (non-constant, space-varying) density of the fluid as a function of the regularised indicator vector $\boldsymbol{\chi}$ and $\boldsymbol{f}(\boldsymbol{\chi})$ is the deviatoric component of the stress tensor of the corresponding continuum phase. In addition, to enforce incompressibility, the following constraint has to be satisfied $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$. This chapter will mainly focus on the discretisation of the deviatoric Cauchy stress tensor $\boldsymbol{f} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'$ in the structure domain.

5.1 Particle representation

In the proposed method, the fluid discretisation is performed on an underlying Eulerian mesh, whilst the structure is discretised by a cloud of particles or integration points. An interpolation-spreading procedure between the underlying grid and the particles must be constructed. Numerous mapping techniques are available in the literature, ranging from discretised versions of the Dirac delta distribution [3,57,58], ideal for background structured fluid meshes, to more sophisticated kernel functions, such as the Reproducing Kernel Particle Method (RKPM) delta functions [117], ideal for non-structured background fluid meshes. In this thesis, we will be focussed on using the Dirac delta distribution on structured fluid meshes. The Dirac delta distribution is only related with the underlying Eulerian mesh, and is irrelevant with the position of the particles.

5.1.1 Quadrature rules

Quadrature refers to the method for numerically approximating the value of an integral $\int_\Omega f(x) \, dv$. In the ISPM [4,5], these particles are the quadrature points/integration points. For each particle, there is an associated quadrature weight. The integral is numerically evaluated as

$$\int_\Omega f(\boldsymbol{x}) \, dv \simeq \sum_i w_i f(\boldsymbol{x}_i), \qquad (5.2)$$

where w_i is the quadrature weight and $f(\boldsymbol{x}_i)$ is the value at the quadrature point $\boldsymbol{x}_i$. A simple procedure was employed for the construction of a low-order quadrature rule for an arbitrary immersed solid domain [4]. Given a solid domain in Ω_s^0, a classical mesh is constructed with any of the many existing mesh generator computer packages [118], and the corresponding dual mesh is considered (see Fig. 5.1). Using this approach, the integration points are taken as the nodes of the original mesh, and the integration weights as the tributary areas/volumes of the corresponding dual element to that integration point.

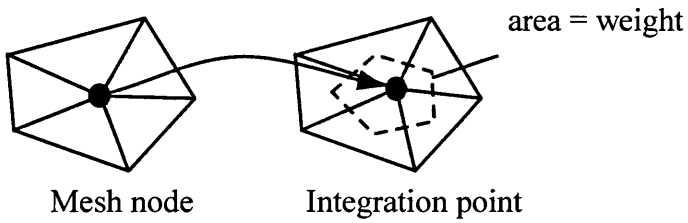


Figure 5.1: Construction of dual mesh and assignment of integration point position and weight. Left: Classical FEM unstructured triangular mesh. Right: corresponding dual mesh and weight

5.1.2 Interpolation-spreading operators

In this section, we will discuss the interpolation-spreading algorithmic procedure between the grid and the particles for the exchange of information. For illustration purposes, a collocated grid rather than a staggered grid is shown in Figure 5.2 and 5.3.

Interpolation from grid to particle

Consider f representing a scalar function to be evaluated on the grid and particles. The value $f(\boldsymbol{x}_s)$ on a particle located at $\boldsymbol{x}_s$, as well as its gradient value $\boldsymbol{\nabla} f(\boldsymbol{x}_s)$, can be interpolated from the grid spatial information. A schematic diagram of the interpolation operation is shown in Figure 5.2. On a discrete level, the interpolation operator $\mathcal{I}$ can be written as

$$\mathcal{I}(f(\boldsymbol{x}_s)) = \int_\Omega f(\boldsymbol{x})\delta(\boldsymbol{x}_s - \boldsymbol{x})\, dv \simeq \sum_{A_j} f(\boldsymbol{x}^{A_j})\hat{\delta}(\boldsymbol{x}_s - \boldsymbol{x}^{A_j}), \tag{5.3a}$$

$$\boldsymbol{\nabla}\mathcal{I}(f(\boldsymbol{x}_s)) = \int_\Omega f(\boldsymbol{x})\boldsymbol{\nabla}\delta(\boldsymbol{x}_s - \boldsymbol{x})\, dv \simeq \sum_{A_j} f(\boldsymbol{x}^{A_j})\boldsymbol{\nabla}\hat{\delta}(\boldsymbol{x}_s - \boldsymbol{x}^{A_j}), \tag{5.3b}$$

where A_j denotes the grid edges and the $\boldsymbol{x}^{A_j}$ is the position of the corresponding node. The $\hat{\delta}(\boldsymbol{x} - \boldsymbol{x}^{A_j})$ denotes a suitable interpolating kernel function centred at node A_j. The interpolation procedure is the summation over $\boldsymbol{x}^{A_j}$. The equations (5.3) are used for evaluation of the velocity and its gradient on the particles.

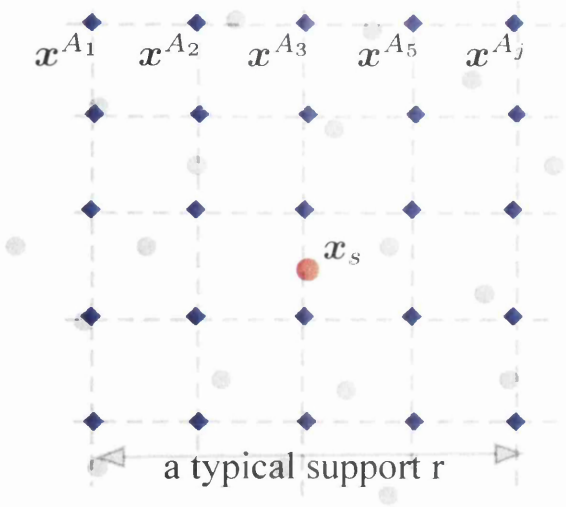


Figure 5.2: Schematic diagram of interpolation operation: the value on the particle $\boldsymbol{x}_s$ (red) is approximated by the summation over the value on the grid $\boldsymbol{x}^{A_j}$ (blue).

Spreading from particle to grid

Let us consider a piecewise constant approximation for the grid value averaged by unite volume. From the quadrature rule (5.2)

$$\int_{\Omega} \boldsymbol{f}\, dv \simeq \sum_{a_p} W^{a_p} \boldsymbol{f}(\boldsymbol{x}^{a_p}). \tag{5.4}$$

A schematic diagram of the spreading operations is shown in Figure 5.3. Consider

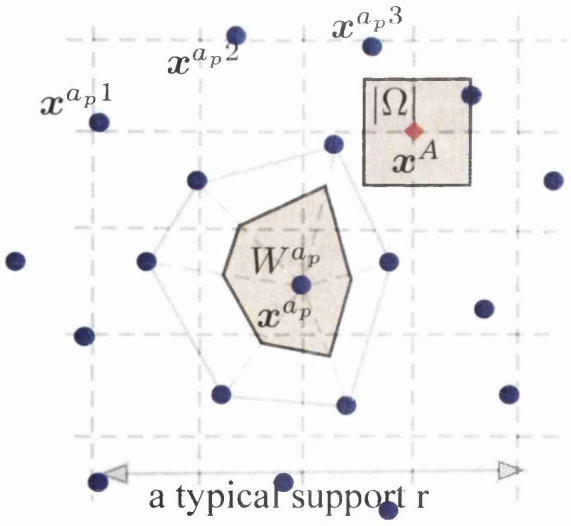


Figure 5.3: Schematic diagram of spreading operation: the value on the grid $\boldsymbol{x}^A$ (red) is approximated by the summation over the value on the particle $\boldsymbol{x}^{a_p}$ (blue).

a vector value $\boldsymbol{f}$ spread from particle to grid. On the discrete level, the value $\boldsymbol{f}(\boldsymbol{x})$

on grid $\boldsymbol{x}$ and its divergence $\boldsymbol{\nabla} \cdot \boldsymbol{f}(\boldsymbol{x})$ can be obtained by

$$\mathcal{S}(\boldsymbol{f}(\boldsymbol{x})) = \int_{\Omega_s} \boldsymbol{f}(\boldsymbol{x}_s)\delta(\boldsymbol{x}-\boldsymbol{x}_s)\, dv \simeq \sum_{a_p} \frac{W^{a_p}}{|\Omega|}\boldsymbol{f}(\boldsymbol{x}^{a_p})\hat{\delta}(\boldsymbol{x}-\boldsymbol{x}^{a_p}), \tag{5.5a}$$

$$\boldsymbol{\nabla} \cdot \mathcal{S}(\boldsymbol{f}(\boldsymbol{x})) = \int_{\Omega_s} \boldsymbol{f}(\boldsymbol{x}_s) \cdot \boldsymbol{\nabla}\delta(\boldsymbol{x}-\boldsymbol{x}_s)\, dv \simeq \sum_{a_p} \frac{W^{a_p}}{|\Omega|}\boldsymbol{f}(\boldsymbol{x}^{a_p}) \cdot \boldsymbol{\nabla}\hat{\delta}(\boldsymbol{x}-\boldsymbol{x}^{a_p}), \tag{5.5b}$$

where W_{a_p} is the weight of quadrature node a_p and $|\Omega|$ is the area of a grid cell.

5.1.3 Derivation of the immersed force from the Cauchy stress tensor

Applying Eq. (5.5b), the immersed structure force $\boldsymbol{f} = \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'$ can be evaluated using the spreading operator on a discrete level

$$\begin{aligned}
\tilde{f}_x^{A_x} &= \mathcal{S}(\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'^{A_x}) \simeq \sum_{a_p} \frac{W^{a_p}}{|\Omega|}\boldsymbol{\sigma}_x'^{s,a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}-\boldsymbol{x}^{A_x}), \\
\tilde{f}_y^{A_y} &= \mathcal{S}(\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}'^{A_y}) \simeq \sum_{a_p} \frac{W^{a_p}}{|\Omega|}\boldsymbol{\sigma}_y'^{s,a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}-\boldsymbol{x}^{A_y}).
\end{aligned} \tag{5.6}$$

A key advancement in methodology, clearly visible in Eq. (5.6), is the direct integration of the deformable solid stresses from solid integration points to fluid cell edges, similarly to the Material Point Method (MPM) [119]. In this approach the structure will be treated exclusively as an assemblage or collection of particles a_p with an associated weight. This is clearly in contrast to previous approaches that require classical FEM meshes (IFEM [120]). In the method presented here, there is no Lagrangian FEM mesh associated with the structure, and therefore, interpolation/spreading operators are not based upon classical FEM shape functions.

Specifically, in alternative immersed methodologies [120,121], first, the force field at the nodes a of a deformable solid Finite Element mesh is computed as a result of the spatial integration of the Cauchy stress tensor through the use of the spatial gradient of standard nodal Finite Element shape functions $\boldsymbol{f}^a = \int_{\Omega_0^s} \boldsymbol{\sigma}'^s \nabla N^a(\boldsymbol{x})\, dV$; second, this force field is spread to the fluid cell edges $\boldsymbol{A}$ through the use of suitable kernel functions, that is, $\boldsymbol{f}^A = \sum_a \boldsymbol{f}^a \hat{\delta}^A(\boldsymbol{x}^a)$. However, in the immersed method presented the force field at the fluid cell edges is obtained directly from the spatial integration of the Cauchy stress tensor through the use of the spatial gradient of the kernel functions as $\boldsymbol{f}^A = \int_{\Omega^s} \boldsymbol{\sigma}'^s \nabla\hat{\delta}^A(\boldsymbol{x})\, dv$. Therefore, the method in this thesis reduces the number of steps (thus, diffusion errors) needed to achieve the force field.

5.1.4 Alternative derivation from an energy potential

In the previous section, the discrete immersed solid force is directly derived from the Cauchy stress. To be consistent with the Immersed Structural Potential Method (ISPM), the same formulation can be derived from a energy potential in the weak formulation. Within the framework of low order Finite Volume schemes, Ω can be regarded as a control volume where the above vector equation can be re-interpreted according to an Eulerian variational formulation as the following weak form,

$$\begin{aligned} 0 = \delta W(\boldsymbol{\Psi}, \delta \boldsymbol{u}) = & \underbrace{\int_{\Omega} \delta \boldsymbol{u} \cdot \left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{\nabla} \boldsymbol{u}) \boldsymbol{u} \right] dv}_{\delta W^{dyn}} \\ & + \underbrace{\int_{\Omega} \delta \boldsymbol{u} \cdot \boldsymbol{\nabla} p \, dv - \int_{\Omega_f} \delta \boldsymbol{u} \cdot \mu \nabla^2 \boldsymbol{u} \, dv + \delta W^s(\boldsymbol{\Psi}, \delta \boldsymbol{u})}_{\delta \tilde{W}^{int}} \\ & + \underbrace{\int_{\Omega} \delta \boldsymbol{u} \cdot \rho \boldsymbol{g} \, dv}_{\delta \tilde{W}^{ext}} \end{aligned} \tag{5.7}$$

where $\delta \boldsymbol{u}$ is an arbitrary piecewise constant virtual velocity field with support $\overline{\Omega} = \Omega \bigcup \partial\Omega$ and δW^s is a energy potential from the structure.

In order to describe the constitutive behaviour of the structure and within the context of hyperelasticity, the Helmholtz's free energy functional Π^s of the structure can be defined as follows,

$$\Pi^s(\boldsymbol{\Psi}) = \int_{\Omega_0^s} \psi^s(\boldsymbol{\Psi}) \, dv \simeq \sum_{a_p} \psi^s(\boldsymbol{\Psi}^{a_p}) \, W^{a_p} \tag{5.8}$$

where ψ^s, also known as the stored strain energy density functional, is defined per unit of undeformed volume Ω_0^s. In the above formula (5.8), factor W^{a_p} represents the material or Lagrangian weight associated with a structure integration point or particle a_p. It is important to emphasise that only the initial location of the integration points $\boldsymbol{x}^{a_p}$ and their Lagrangian weights are required to describe the structural domain, insofar as no interpolating/spreading operations will be carried out by using information related to the solid domain, which differs from the current existing immersed methodologies.

In order to ensure kinematic compatibility (i.e. non-slip condition in the case of a viscous fluid) between the fluid and structure phases, suitable interpolating kernel functions carefully chosen to satisfy appropriate reproducibility conditions are constructed in order to transfer kinematic information between the Eulerian fluid mesh and the structure integration points.

Thus, in the case of a two-dimensional problem, the structure velocity field vector $\boldsymbol{u}^s = [u^s, \, v^s]^{\mathrm{T}}$ evaluated at a structure particle a_p is interpolated from the Eulerian

fluid mesh as follows,

$$\boldsymbol{u}^{a_p} = [u^{a_p},\ v^{a_p}]^{\mathrm{T}}; \qquad u^{a_p} = \sum_{A_x} u^{A_x}\hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}); \qquad v^{a_p} = \sum_{A_y} v^{A_y}\hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}) \tag{5.9}$$

where A_x and A_y denote fluid cell edges perpendicular to the ox and oy Cartesian axis respectively. Both $\hat{\delta}^{A_x}$ and $\hat{\delta}^{A_y}$, defined as

$$\hat{\delta}^{A_x}(\boldsymbol{x}) = \hat{\delta}(\boldsymbol{x} - \boldsymbol{x}^{A_x}), \quad \hat{\delta}^{A_y}(\boldsymbol{x}) = \hat{\delta}(\boldsymbol{x} - \boldsymbol{x}^{A_y}) \tag{5.10}$$

stand for the kernel functions centred at fluid cell edges A_x (vertical edge, for horizontal velocity components) and A_y (horizontal edge, for vertical velocity components), respectively, where $\boldsymbol{x}^{A_x}$ and $\boldsymbol{x}^{A_y}$ is the midpoint of the corresponding mesh edge. Note that in what follows superindices $(\cdot)^{A_x}$ and $(\cdot)^{A_y}$ will be used when referring to fluid cell edges A_x and A_y, respectively. It is noteworthy to emphasise that as a result of the staggering of the Cartesian components of the fluid velocity field (a key aspect of MAC schemes), namely u^{A_x} in vertical edges and v^{A_y} in horizontal edges, the interpolation of the velocity field into the structural domain has also to be split into these Cartesian directions. A more compact formulation could be achieved if a non-staggering methodology had been adopted to solve the Navier-Stokes equations. For the subsequent development of the so-called internal virtual work or weak form, a virtual velocity field vector $\delta\boldsymbol{u}^s = [\delta u^s\ \delta v^s]^{\mathrm{T}}$ evaluated at a structure particle a_p can also be described as,

$$\delta\boldsymbol{u}^{a_p} = [\delta u^{a_p}\ \delta v^{a_p}]^{\mathrm{T}}; \qquad \delta u^{a_p} = \sum_{A_x} \delta u^{A_x}\hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}); \qquad \delta v^{a_p} = \sum_{A_y} \delta v^{A_y}\hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}) \tag{5.11}$$

where a consistent interpolating methodology is employed as in Eq. (5.9), to ensure conservation of the overall scheme. Thus, recalling energy principles, the internal virtual work formulated in the case of the structure domain is defined as the directional derivative of the Helmholtz's free energy functional with respect to a virtual velocity field vector as follows,

$$\delta W^s_{int}(\boldsymbol{\Psi}, \delta\boldsymbol{u}^s) = D\Pi^s(\boldsymbol{\Psi})\,[\delta\boldsymbol{u}^s] \tag{5.12}$$

and after substitution of Eq. (5.8) into (5.12), we arrive at

$$\delta W^s_{int}(\boldsymbol{\Psi}, \delta\boldsymbol{u}^s) = \int_{\Omega^0_s} \frac{\partial\psi^s}{\partial\boldsymbol{F}} : D\boldsymbol{F}[\delta\boldsymbol{u}^s]\, dV \simeq \sum_{a_p} W^{a_p} \left(\frac{\partial\psi^s}{\partial\boldsymbol{F}} : D\boldsymbol{F}[\delta\boldsymbol{u}^s]\right)^{a_p}. \tag{5.13}$$

By noticing the work-conjugacy property between the pair formed by the deformation gradient tensor $\boldsymbol{F}$ and the structure first Piola-Kirchhoff stress tensor $\boldsymbol{P}^s$, Eq.

(5.13) can be re-written as follows

$$\delta W^s_{int}(\boldsymbol{\Psi}, \delta \boldsymbol{u}^s) = \int_{\Omega^0_s} \boldsymbol{P}^s : D\boldsymbol{F}[\delta \boldsymbol{u}^s]\, dV \simeq \sum_{a_p} W^{a_p} \boldsymbol{P}^{s,a_p} : D\boldsymbol{F}[\delta \boldsymbol{u}^{a_p}]. \tag{5.14}$$

Thus, linearisation of the deformation gradient tensor $\boldsymbol{F}$ with respect to the virtual velocity vector $\delta \boldsymbol{u}^{a_p}$ leads to

$$D\boldsymbol{F}[\delta \boldsymbol{u}^{a_p}] = \boldsymbol{\nabla}_0 \delta \boldsymbol{u}^{a_p} = [\boldsymbol{\nabla}_0 \delta u^{a_p}\ \boldsymbol{\nabla}_0 \delta v^{a_p}]^{\mathrm{T}} \tag{5.15}$$

where $\boldsymbol{\nabla}_0$ is the standard material gradient operator. After use of the expansion (5.11), the last term in Eq. (5.15) can be formulated as follows

$$\boldsymbol{\nabla}_0 \delta u^{a_p} = \sum_{A_x \in I_x^{a_p}} \delta u^{A_x} \boldsymbol{\nabla}_0 \hat{\delta}^{A_x}(\boldsymbol{x}^{a_p});\ \ \boldsymbol{\nabla}_0 \delta v^{a_p} = \sum_{A_y \in I_y^{a_p}} \delta v^{A_y} \boldsymbol{\nabla}_0 \hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}) \tag{5.16}$$

After re-writing the first Piola-Kirchhoff stress tensor in the form $\boldsymbol{P}^s = \left[\boldsymbol{P}^s_x\ \boldsymbol{P}^s_y\right]^{\mathrm{T}}$, Eq. (5.14) can be reformulated in a continuum manner as

$$\delta W^s_{int}(\boldsymbol{\Psi}, \delta \boldsymbol{u}^s) = \int_{\Omega^0_s} \left(\boldsymbol{P}^s_x \cdot \boldsymbol{\nabla}_0 \delta u^s + \boldsymbol{P}^s_y \cdot \boldsymbol{\nabla}_0 \delta v^s \right) dV \tag{5.17}$$

or in terms of the collection of material particles a_p as

$$\delta W^s_{int}(\boldsymbol{\Psi}, \delta \boldsymbol{u}^s) \simeq \sum_{a_p} W^{a_p} \left(\sum_{A_x \in I_x^{a_p}} \delta u^{A_x} \boldsymbol{P}^{s,a_p}_x \cdot \boldsymbol{\nabla}_0 \hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}) + \sum_{A_y \in I_y^{a_p}} \delta v^{A_y} \boldsymbol{P}^{s,a_p}_y \cdot \boldsymbol{\nabla}_0 \hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}) \right). \tag{5.18}$$

After interchanging the summation operators, a final re-arrangement of above formula (5.18) yields

$$\delta W^s_{int}(\boldsymbol{\Psi}, \delta \boldsymbol{u}^s) = \sum_{A_x \in I_x^{a_p}} \delta u^{A_x} f_x^{A_x} + \sum_{A_x \in I_y^{a_p}} \delta v^{A_y} f_y^{A_y} \tag{5.19}$$

where $\boldsymbol{f} = \left[f_x^{A_x}\ f_y^{A_y}\right]^{\mathrm{T}}$ represent the equivalent internal fluid-structure interaction forces that must be applied at the fluid cell edges A_x and A_y. These equivalent internal forces, in a Total Lagrangian Formulation (TLF) can be finally presented as

$$\begin{aligned} f_x^{A_x} &= \int_{\Omega^0_s} \boldsymbol{P}^s_x \cdot \boldsymbol{\nabla}_0 \hat{\delta}^{A_x}(\boldsymbol{x}^s) dV \simeq \sum_{a_p} W^{a_p} \boldsymbol{P}^{s,a_p}_x \cdot \boldsymbol{\nabla}_0 \hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}), \\ f_y^{A_y} &= \int_{\Omega^0_s} \boldsymbol{P}^s_y \cdot \boldsymbol{\nabla}_0 \hat{\delta}^{A_y}(\boldsymbol{x}^s) dV \simeq \sum_{a_p} W^{a_p} \boldsymbol{P}^{s,a_p}_y \cdot \boldsymbol{\nabla}_0 \hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}). \end{aligned} \tag{5.20}$$

Alternatively, after a suitable push forward operation, above formulae (5.20) can

be re-expressed according to an Updated Lagrangian Formulation (ULF) by the introduction of the Kirchhoff stress tensor $\boldsymbol{\tau}^s = \left[\boldsymbol{\tau}_x^s \boldsymbol{\tau}_y^s\right]$ and the spatial gradient operator $\boldsymbol{\nabla}$ as

$$\begin{aligned} f_x^{A_x} &= \int_{\Omega_s^0} \boldsymbol{\tau}_x^s \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^s) dV \simeq \sum_{a_p} W^{a_p} \boldsymbol{\tau}_x^{s,a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}) \\ f_y^{A_y} &= \int_{\Omega_s^0} \boldsymbol{\tau}_y^s \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^s) dV \simeq \sum_{a_p} W^{a_p} \boldsymbol{\tau}_y^{s,a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}) \end{aligned} \tag{5.21}$$

where it must be recalled that the Kirchhoff stress tensor $\boldsymbol{\tau}^s$ and the first Piola-Kirchhoff stress tensor $\boldsymbol{P}^s$ are related as

$$\boldsymbol{\tau}^s = \boldsymbol{P}^s \boldsymbol{F}^{\mathrm{T}}; \quad \boldsymbol{\tau}^{s,a_p} = \boldsymbol{P}^{s,a_p} \boldsymbol{F}^{a_p \mathrm{T}}. \tag{5.22}$$

The above total internal forces $f_x^{A_x}$ and $f_y^{A_y}$ have to be appropriately translated into forces per unit volume $\tilde{\boldsymbol{f}}$. This yields the following relationship

$$\tilde{f}_x^{A_x} = \frac{1}{\Delta x \Delta y} f_x^{A_x}, \quad \tilde{f}_y^{A_y} = \frac{1}{\Delta x \Delta y} f_y^{A_y}. \tag{5.23}$$

Within the ISPM, where the fluid is also regarded as incompressible and hence the velocity field is divergence free $\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0$, the pressure parameter needed to enforce such a kinematic constraint is already obtained as part of the fractional step algorithm used to solve the Navier-Stokes equations. Therefore, only deviatoric stress components need to be incorporated as part of the fluid-structure interaction forces in Eq. (5.20 or 5.21). By applying the Kirchhoff stress and Cauchy stress relation $\boldsymbol{\tau} = J\boldsymbol{\sigma}$ yielding

$$\begin{aligned} f_x^{A_x} &= \int_{\Omega^s} \boldsymbol{\sigma}_x^s \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^s)\, dv \simeq \sum_{a_p} W^{a_p} \boldsymbol{\sigma}_x^{s,a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}) \\ f_y^{A_y} &= \int_{\Omega_s} \boldsymbol{\sigma}_y^s \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^s)\, dv \simeq \sum_{a_p} W^{a_p} \boldsymbol{\sigma}_y^{s,a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}) \end{aligned} \tag{5.24}$$

as the Jacobian of the deformation remains one for the entire deformation.

5.1.5 Comparison with ISPM

In the previous sections, we have written the discretised immersed structure forces, from the Cauchy stress tensor or directly from a energy potential. However, in the original ISPM, the following two additional forces are added while in the proposed method the following forces are not required.

The ISPM solves the Navier-Stokes equations for the overall domain. A Newtonian viscous stress contribution is considered implicitly within the structure domain

through the viscous term $\mu\boldsymbol{\Delta u}$. This term must clearly be removed as follows,

$$\begin{aligned} f_x^{flu,A_x} &= -\int_{\Omega_s^0} \boldsymbol{\tau}'_x \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^s)\, dV = -\sum_{a_p} W^{a_p} \boldsymbol{\tau}_x'^{a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}^{a_p}) \\ f_y^{flu,A_y} &= -\int_{\Omega_s^0} \boldsymbol{\tau}_y'^{s} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^s),\ dV = -\sum_{a_p} W^{a_p} \boldsymbol{\tau}_y'^{a_p} \cdot \boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}). \end{aligned} \quad (5.25)$$

In the proposed 'one-phase' formulation, this is not required because the fluid viscous part is only evaluated at the fluid domain.

In the ISPM, when studying real transient interaction phenomena, where the density of the fluid phase ρ and the solid phase ρ_s differ, inertia forces must also be incorporated into the analysis in order to account for such a discrepancy. Thus, a new inertia fluid structure interaction force term $\boldsymbol{f}^{ine} = \left[f_x^{ine,A_x}\ f_y^{ine,A_y}\right]^{\mathrm{T}}$ should be added as follows,

$$\begin{aligned} f_x^{ine,A_x} &= \int_{\Omega_s^0} (\rho_s - \rho)\left(g_x - \frac{\mathrm{d}u^s}{\mathrm{d}t}\right)\hat{\delta}^{A_x}(\boldsymbol{x}^s)\, dV \\ f_y^{ine,A_y} &= \int_{\Omega_s^0} (\rho_s - \rho)\left(g_y - \frac{\mathrm{d}v^s}{\mathrm{d}t}\right)\hat{\delta}^{A_y}(\boldsymbol{x}^s)\, dV \end{aligned} \quad (5.26)$$

where $\boldsymbol{g} = [g_x\ g_y]^{\mathrm{T}}$ represents a possible body force field per unit of undeformed volume (i.e. gravity effects). In the proposed 'one-phase' formulation, the inertia is treated implicitly by leaving the inertial contribution to be dealt with in an Eulerian manner by the fluid solver. Thus, the proposed method is able to deal with large density ratios.

5.2 Computational aspects

5.2.1 Spline-based kernels

Since the inception of the IBM, it has been crucial to construct an approximation to the Dirac delta distribution. In the original IBM, Peskin introduced a cosine kernel [3] for simplicity. Then introduced Peskin-Roma kernel [122], and revisit to define some properties to construct the kernel [61], which includes the requirement for continuity, limited compact support, discrete reproducibility. The meshless reproducing kernel particle method was applied for improve the continuity [117]. Some numerical approximation a singular terms is analysed by Tornberg [123]. Recent a high order spline-based kernel is introduced for continuous in high derivative [5]. Different forms have been proposed in the literature, as shown in Figure 5.4.

There are two options to build a higher dimensional kernel. One is through $r = ||\boldsymbol{x} - \boldsymbol{X}||_2$, another is by the tensor product. It is argued by the authors [5] that the tensorised high dimensional kernels owns better convergence property. Given

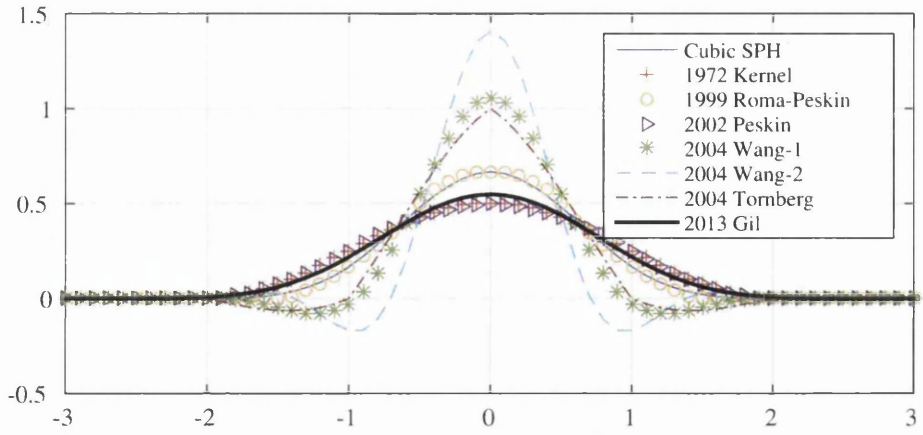


Figure 5.4: Illustration of different kernels used in the Immersed Boundary Method and its variants

the arrangement of the structured fluid mesh, a tensor product approximation to the delta distribution has been considered

$$\hat{\delta}(\boldsymbol{x}) = \prod_{i=1}^{i=n} \delta^1_{\Delta x_i}(x_i) \tag{5.27}$$

where Δx_i is the mesh spacing in the i-th coordinate direction, n is the number of space dimensions and the one-dimensional approximation δ^1_h has been chosen as

$$\delta^1_h(x) = \frac{1}{h}\Psi\left(\frac{x}{h}\right), \tag{5.28}$$

where Ψ is a continuous function which can be given by any of the possible formulas

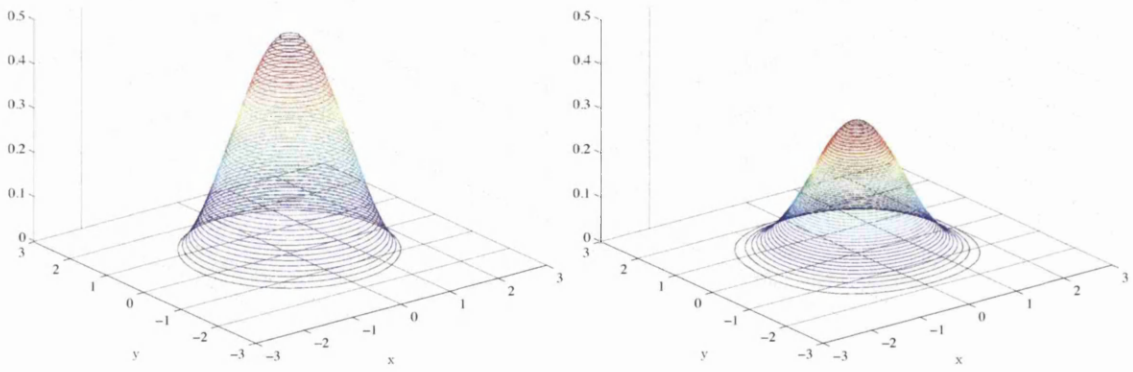


Figure 5.5: Comparison the techniques of constructing kernels from one dimension to higher dimensions, shown in 2D. Left: Peskin's kernel by radius. Right: kernel by tensor product.

below [3, 5, 57, 122]

- Peskin's original kernel [3]

$$\Psi(r) = \begin{cases} \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 4(3 - |r|)|r|}), & 1 \leq |r| \leq 2, \\ \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4(1 - |r|)|r|}), & |r| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.29}$$

- Approximation to Peskin's original kernel [57]

$$\Psi(r) = \begin{cases} \frac{1}{4}(1 + \cos\frac{\pi r}{2}), & |r| \leq 2, \\ 0, & \text{otherwise} \end{cases} \tag{5.30}$$

- Roma-Peskin's kernel [122]

$$\Psi(r) = \begin{cases} \frac{1}{6}(5 - 3|r| - \sqrt{1 - 3(1 - |r|)^2}, & 1 \leq |2r| \leq 3, \\ \frac{1}{3}(1 + \sqrt{1 - 3r^2}), & |2r| < 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.31}$$

- Splined based kernel [5]

$$\Psi(r) = \begin{cases} \frac{11}{756}r^7 - \frac{1}{18}r^6 + \frac{7}{36}r^4 - \frac{29}{60}r^2 + \frac{691}{1260} & 0 < |r| \leq 1 \\ -\frac{17}{1512}r^7 + \frac{1}{8}r^6 - \frac{13}{24}r^5 + \frac{79}{72}r^4 - \frac{65}{72}r^3 + \frac{7}{120}r^2 - \frac{13}{72}r + \frac{1447}{2520} & 1 < |r| \leq 2 \\ \frac{29}{7560}r^7 - \frac{5}{72}r^6 + \frac{21}{40}r^5 - \frac{17}{8}r^4 + \frac{39}{8}r^3 - \frac{243}{40}r^2 + \frac{27}{8}r - \frac{81}{280} & 2 < |r| \leq 3 \\ 0 & 3 < |r| \end{cases} \tag{5.32}$$

The main advantage of the splined based kernel is its higher differentiability (smoothness) whilst maintaining the reproducibility of first order [5]. The use of splined based kernel is beneficial when having to evaluate the stress which requires of the evaluation of the derivative of the kernel. Comparison of original kernel and the splined based kernel is shown in Figure 5.6. It is obvious that the splined kernel is smooth in the derivative whilst the original kernel of Peskin is not.

The splined based will remove the spurious oscillations present in the structure stress field in comparison with existing IBM kernels available in the literature [5]. These spurious oscillations emerge from the artificial lack of regularity introduced by the use of the standard IBM kernels.

5.2.2 Deformation gradient tensor

A vital of the ingredient of the ISPM is the need to evaluate the deformation gradient tensor $\boldsymbol{F}$ at any particle a_p, namely $\boldsymbol{F}^{a_p}$. In conventional immersed methodologies [117,120], the deformation gradient tensor $\boldsymbol{F}$ is obtained upon material differentiation of the spatial coordinates of the structure nodes, which leads to lower spatial resolution of structural stresses and the non-compliance with the kinematic

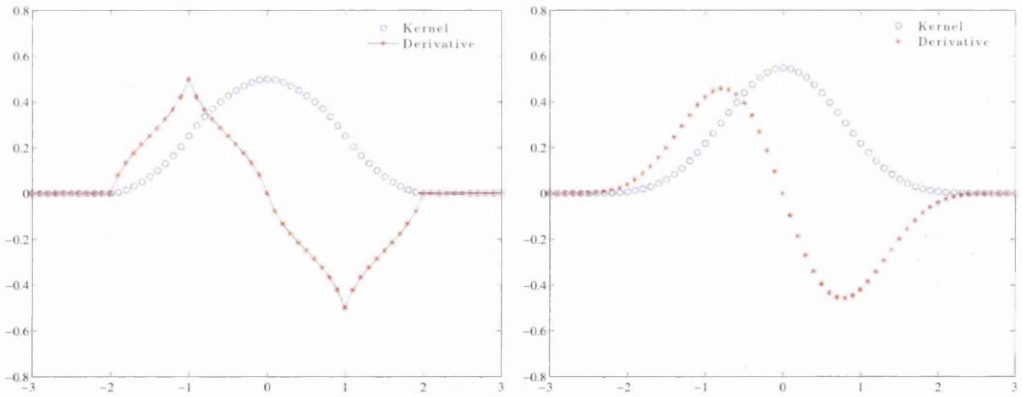


Figure 5.6: Comparison of Peskin's kernel and high order spline kernel and their derivative. Left, Peskin's orignal kernel; right, spline-based kernel.

constraints, such as incompressibility behaviour, namely $J = 1$ [5]. In order to overcome these shortcomings, the approach introduced in [5] is followed, where the following tensor system of kinematic differential equations is solved,

$$\dot{\boldsymbol{F}} = \boldsymbol{l}\boldsymbol{F} = (\boldsymbol{d} + \boldsymbol{w})\boldsymbol{F}, \tag{5.33}$$

is time-integrated to obtain the deformation gradient tensor, where the velocity gradient tensor $\boldsymbol{l}$ is decomposed into its symmetric and skew-symmetric components, $\boldsymbol{d}$ and $\boldsymbol{w}$, respectively. If the above system had a constant coefficient tensor $\boldsymbol{l}$, then the exact solution would be

$$\boldsymbol{F} = e^{\boldsymbol{l}(t-t_0)}\boldsymbol{F}_0, \tag{5.34}$$

$\boldsymbol{F}_0$ being the deformation gradient tensor at time t_0. This motivates us to consider the explicit time integration scheme,

$$\boldsymbol{F}^{n+1} = e^{\Delta t \boldsymbol{l}^n}\boldsymbol{F}^n, \quad n = 0, 1, 2, \ldots \tag{5.35}$$

for the solution of (5.33), where the superscript n is taken as evaluation at time t^n. Clearly, the problem that remains is the computation of $e^{\Delta t \boldsymbol{l}^n}$.

Now let us introduce $\boldsymbol{d}^n = \frac{1}{2}\left(\boldsymbol{l}^n + (\boldsymbol{l}^n)^{\mathrm{T}}\right)$ and $\boldsymbol{w}^n = \frac{1}{2}\left(\boldsymbol{l}^n - (\boldsymbol{l}^n)^{\mathrm{T}}\right)$. It can be noted that $\boldsymbol{d}^n$ and $\boldsymbol{w}^n$ commute iff $\boldsymbol{l}^n$ is normal, which is not true in general, and hence the spectral decomposition theorem cannot be applied to $\boldsymbol{l}^n$ [5].

In the ISPM, the authors introduced a computationally inexpensive method that computes an approximation to this term and has favourable conservation properties. The solution of the tensor system (5.33) can be summarised as follows,

$$\boldsymbol{F}^{n+1} = \left(I - \frac{1}{2}\Delta t \bar{\boldsymbol{w}}^n\right)^{-1}\left(I + \frac{1}{2}\Delta t \bar{\boldsymbol{w}}^n\right)\left(\sum_{i=1}^{3} e^{\bar{\lambda}_i \Delta t}\bar{\boldsymbol{v}}_i \otimes \bar{\boldsymbol{v}}_i\right)\boldsymbol{F}^n, \quad n = 0, 1, 2, \ldots \tag{5.36}$$

where λ_i and $\boldsymbol{v}_i$ are the eigenvalues and eigenvectors of $\boldsymbol{d}^n$, respectively. Note

that the computed approximation to $e^{\Delta t \boldsymbol{l}^n}$ has the property of maintaining $J = 1$ exactly. Nonetheless, it is also interesting to notice that if the selected time integration operator in Eq. (5.33) does not respect the incompressible character of the deformation process and yields a Jacobian not equal to one, only the deviatoric component of the deformation $\widehat{\boldsymbol{F}} = J^{-1/3}\boldsymbol{F}$ must be employed when computing the structural stresses.

For a structure particle a_p, the velocity gradient tensor is obtained by means of consistent interpolation from the background fluid mesh as follows,

$$\boldsymbol{l}^{a_p} = \left[\boldsymbol{l}_x^{a_p}\ \boldsymbol{l}_y^{a_p}\right]^{\mathrm{T}};\quad \boldsymbol{l}_x^{a_p} = \sum_{A_x \in I_x^{a_p}} u^{A_x} \nabla \hat{\delta}^{A_x}(\boldsymbol{x}^{a_p});\quad \boldsymbol{l}_y^{a_p} = \sum_{A_y \in I_y^{a_p}} v^{A_y} \nabla \hat{\delta}^{A_y}(\boldsymbol{x}^{a_p}), \tag{5.37}$$

where the gradient of the kernel functions in both Cartesian directions, $\{\nabla\hat{\delta}^{A_x}, \nabla\hat{\delta}^{A_y}\}$, can be explicitly computed. The formula of $\hat{\delta}^{A_x}$ has been given in Eq. (5.28).

5.2.3 Indicator function

Applying Eq. (5.5a), the indicator function χ can be constructed from spreading a constant function $f = 1$ from the integration point $\boldsymbol{x}^{a_p}$ to the grid

$$\chi(\boldsymbol{x}) = \mathcal{S}(f = 1) \simeq \sum_{a_p} \frac{W^{a_p}}{|\Omega|} \hat{\delta}(\boldsymbol{x} - \boldsymbol{x}^{a_p}) \tag{5.38}$$

Consider the same examples in Section 4.2.1, the whole domain $\Omega = [-1.2, 1.2] \times [-1.2, 1.2]$ and a solid domain Ω_s, which has a shape of sphere, centred in $[0, 0]$ with a radius $r = 1$ immersed in Ω. The Indicator function can be constructed as shown in Figure 5.7 The $|\Omega|$ remain constant because the fixed Eulerian grid do not move.

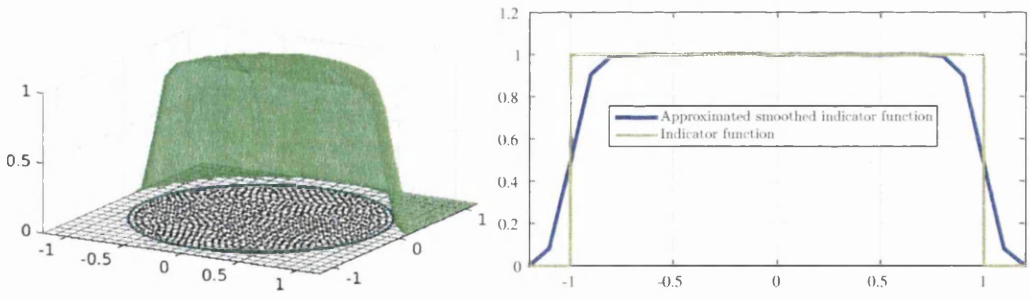


Figure 5.7: The approximation of the indicator function by the integration points. Up: Dual mesh, weight and integration point. Down: Cross section of the approximated Indicator function.

The W^{a_p} does not change with time, because of the incompressible condition.

The indicator function $\chi(\boldsymbol{x})$ is advected with the velocity, because the integration point is moved passively with the background velocity in Eq. (3.27), which can be

discretised as

$$\boldsymbol{x}^{a_p,n+1} = \boldsymbol{x}^{a_p,n} + \frac{1}{2}\Delta t \left(\boldsymbol{u}^{a_p,n+1} + \boldsymbol{u}^{a_p,n}\right) \tag{5.39}$$

As a result, the indicator function updates every time step according to the $\boldsymbol{x}^{a_p,n+1}$.

5.3 Algorithm

In this section, a flowchart of the overall fluid-structure interaction algorithm described in this chapter is included to illustrate the various steps of the methodology presented. By using a simple fixed point iterative scheme, the coupled fluid-structure interaction algorithm can be resolved to advance from time step n to $n+1$ in an iterative fashion to ensure the complete coupling of the fluid and solid equations. By computing a residual norm based upon the difference between the fluid-structure interaction forces in two successive iterations k and $k+1$, namely $\|\boldsymbol{f}_{k+1}^{n+1} - \boldsymbol{f}_{k}^{n+1}\|/\|\boldsymbol{f}^{n}\|$, a convergence criterion can be easily established in order to progress to the following iteration $k+2$ (if not yet converged) or time step $n+2$ (if convergence is accomplished).

For a given time step $n+1$ and provided a starting solution at iteration k, the following block of actions must be carried within iteration $k+1$:

Algorithm 5.1 : IMMERSED FLUID-STRUCTURE INTERACTION SOLVER

- Fixed point iteration, while $\|\boldsymbol{f}_{k+1}^{n+1} - \boldsymbol{f}_{k}^{n+1}\|/\|\boldsymbol{f}^{n}\| > tolerance$

  - Compute solid immersed forces. Loop over the integration points

    1. Interpolate the velocity from the Eulerian fluid mesh to the integration points a_p, namely $\boldsymbol{u}_{k+1}^{a_p,n+1} = \left[u_{k+1}^{a_p,n+1}\ ,\ v_{k+1}^{a_p,n+1}\right]^{\mathrm{T}}$ as

$$u_{k+1}^{a_p,n+1} = \mathcal{I}_{x,k}^{n+1}\left(u_k^{A_x,n+1}\right) = \sum_{A_x \in I_x^{a_p}} u_k^{A_x,n+1}\hat{\delta}^{A_x}(\boldsymbol{x}_k^{a_p,n+1})$$
$$v_{k+1}^{a_p,n+1} = \mathcal{I}_{y,k}^{n+1}\left(v_k^{A_y,n+1}\right) = \sum_{A_y \in I_y^{a_p}} v_k^{A_y,n+1}\hat{\delta}^{A_y}(\boldsymbol{x}_k^{a_p,n+1})$$

    2. Interpolate the velocity gradient tensor at each interpolation point a_p, namely $\boldsymbol{l}_{k+1}^{a_p,n+1} = \left[\boldsymbol{l}_{x,k+1}^{a_p,n+1}\ \boldsymbol{l}_{y,k+1}^{a_p,n+1}\right]^{\mathrm{T}}$ as,

$$\boldsymbol{l}_{x,k+1}^{a_p,n+1} = \boldsymbol{\nabla}\mathcal{I}_{x,k}^{n+1}\left(u_k^{A_x,n+1}\right) = \sum_{A_x \in I_x^{a_p}} u_k^{A_x,n+1}\boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}_k^{a_p,n+1})$$
$$\boldsymbol{l}_{y,k+1}^{a_p,n+1} = \boldsymbol{\nabla}\mathcal{I}_{y,k}^{n+1}\left(v_k^{A_y,n+1}\right) = \sum_{A_y \in I_y^{a_p}} v_k^{A_y,n+1}\boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}_k^{a_p,n+1})$$

    3. Compute at each particle a_p the deformation gradient tensor $\boldsymbol{F}_{k+1}^{a_p,n+1}$ by suitable integration of the velocity gradient tensor $\boldsymbol{l}_{k+1}^{a_p,n+1}$,

$$\boldsymbol{F}_{k+1}^{a_p,n+1} = e^{\left(\Delta t \boldsymbol{l}_{k+1}^{a_p,n+1}\right)}\boldsymbol{F}^{a_p,n}$$

    4. Evaluate the deviatoric component of the Cauchy stress tensor $\boldsymbol{\sigma}_{k+1}^{'s,a_p,n+1}$ by using an appropriate constitutive law $\mathcal{F}$, which can be summarised as,

$$\boldsymbol{\sigma}_{k+1}^{'s,a_p,n+1} = \mathcal{F}(\boldsymbol{F}_{k+1}^{a_p,n+1}, \boldsymbol{l}_{k+1}^{a_p,n+1})$$

    5. Spreading of the Cauchy stress tensor $\boldsymbol{\sigma}_{k+1}^{'s,a_p,n+1}$ to obtain the equivalent set of immersed structure force field $\boldsymbol{f}_{k+1}^{n+1} = \{f_{x,k+1}^{A_x,n+1}\ ,\ f_{y,k+1}^{A_y,n+1}\}$ at the Eulerian fluid mesh

$$f_{x,k+1}^{A_x,n+1} = \mathcal{S}_{x,k}^{n+1}\left(\boldsymbol{\sigma}_{x,k+1}^{'s,a_p,n+1}\right) = \textstyle\sum_{a_p} W^{a_p}\left(\boldsymbol{\sigma}_{x,k+1}^{'s,a_p,n+1}\right)\cdot\boldsymbol{\nabla}\hat{\delta}^{A_x}(\boldsymbol{x}_k^{a_p,n+1})$$
$$f_{y,k+1}^{A_y,n+1} = \mathcal{S}_{y,k}^{n+1}\left(\boldsymbol{\sigma}_{y,k+1}^{'s,a_p,n+1}\right) = \textstyle\sum_{a_p} W^{a_p}\left(\boldsymbol{\sigma}_{y,k+1}^{'s,a_p,n+1}\right)\cdot\boldsymbol{\nabla}\hat{\delta}^{A_y}(\boldsymbol{x}_k^{a_p,n+1})$$

    6. Update the spatial position of the material integration points a_p

$$\boldsymbol{x}_{k+1}^{a_p,n+1} = \boldsymbol{x}^{a_p,n} + \frac{1}{2}\Delta t\left(\boldsymbol{u}_{k+1}^{a_p,n+1} + \boldsymbol{u}^{a_p,n}\right)$$

  - Solve the Navier-Stokes equations, including the a fluid-structure interaction forcing term $\boldsymbol{f}_{k+1}^{n+1} = \{f_{x,k+1}^{A_x,n+1}\ ,\ f_{y,k+1}^{A_y,n+1}\}$

Chapter 6

Immersed rigid body

As discussed in chapter 2, the rigid continuum Cauchy stress tensor is strongly coupled with the velocity, which can only be solved with an implicit method. However, by introducing an projection operator $\mathsf{P}(\boldsymbol{u})$, the rigid immersed force has a explicit expression as

$$\boldsymbol{f} = \rho \left[\left(\frac{\partial \mathsf{P}(\boldsymbol{u})}{\partial t} \right) + (\boldsymbol{\nabla} \mathsf{P}(\boldsymbol{u})) \mathsf{P}(\boldsymbol{u}) \right] + \boldsymbol{\nabla} p - \rho \boldsymbol{g}.$$

Furthermore, an iterative procedure, similar to that of the immersed deformable solid force, can be introduced for solving the coupled system. The immersed rigid body force, equivalent to the divergence of the 'rigid' Cauchy stress tensor, can be interpreted as the constraint required to ensure that the continuum moves according to the motion of a rigid body.

In the proposed methodology, the most complex part is the numerical approximation of the rigid velocity projection operator $\mathsf{P}(\boldsymbol{u})$ which takes an arbitrary vector field and projects it into a Helicoidal Vector Field (HVF). A HVF is an affine vector field [21] with respect to spatial coordinates $\boldsymbol{x}$. This character inspires a linear weighted least squares projection method for the approximation. The weight in the unified 'one-phase' formulation is its corresponding approximated indicator function, which varies from 0 to 1.

For ease of understanding, a complete algorithm describing the methodology is provided in section 6.3. Compared with the multiphase flow solver presented in Chapter 4, the additional cost lies in having to solve a least squares problem, which is relatively inexpensive in terms of computational cost. By using this method, the considered immersed rigid body motion can be adopt a variety of possibilities, namely fixed in space, only allowed rotation, or free rigid body motion.

6.1 Iterative procedure for the immersed rigid body force

Let us consider a single-phase rigid continuum problem for simplicity. Recall Eq. (2.72), rigid continuum mass and linear momentum equations are formulated as

$$\rho\left[\frac{\partial \boldsymbol{u}}{\partial t}+(\boldsymbol{\nabla u})\boldsymbol{u}\right]=-\boldsymbol{\nabla}p+\boldsymbol{f}+\rho\boldsymbol{g} \tag{6.1a}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u}=0 \tag{6.1b}$$

$$\boldsymbol{f}=\rho\left[\left(\frac{\partial \mathsf{P}(\boldsymbol{u})}{\partial t}\right)+(\boldsymbol{\nabla}\mathsf{P}(\boldsymbol{u}))\mathsf{P}(\boldsymbol{u})\right]+\boldsymbol{\nabla}p-\rho\boldsymbol{g} \tag{6.1c}$$

A fixed-point iteration procedure is proposed for solving Eq. (6.1) as follows

- **Velocity update.** Eq. (6.1) is highly nonlinear. Its integration in time with a forward Euler method can remove some non-linearity leading to an update as

$$\boldsymbol{u}_{k+1}^{n+1}=\boldsymbol{u}^{n}-\frac{\Delta t}{\rho}(\rho(\boldsymbol{\nabla u}^{n})\boldsymbol{u}^{n}+\boldsymbol{\nabla}p^{n}-\rho\boldsymbol{g}-\boldsymbol{f}_{k}^{n+1}). \tag{6.2}$$

  Note that it is critical that $\boldsymbol{f}$ is calculated at $n+1$ in order to enforce correctly the rigid body motion.

- **Force update.** In order to update the immersed force vector note first that, assuming $\mathsf{P}(\boldsymbol{u}^{n})=\boldsymbol{u}^{n}$, after some algebra, the immersed rigid body force becomes

$$\begin{aligned}
\boldsymbol{f}_{k+1}^{n+1}&=\rho\left[\frac{\mathsf{P}(\boldsymbol{u}_{k+1}^{n+1})-\mathsf{P}(\boldsymbol{u}^{n})}{\Delta t}\right]+\boldsymbol{\nabla}p^{n}+(\boldsymbol{\nabla}\mathsf{P}(\boldsymbol{u}^{n}))\mathsf{P}(\boldsymbol{u}^{n})-\rho\boldsymbol{g}\\
&=\rho\left[\frac{\mathsf{P}(\boldsymbol{u}_{k+1}^{n+1})-\boldsymbol{u}^{n}}{\Delta t}+\frac{1}{\rho}\boldsymbol{\nabla}p^{n}+\frac{1}{\rho}(\boldsymbol{\nabla u}^{n})\boldsymbol{u}^{n}-\boldsymbol{g}-\frac{1}{\rho}\boldsymbol{f}_{k}^{n+1}+\frac{1}{\rho}\boldsymbol{f}_{k}^{n+1}\right]\\
&=\frac{\rho}{\Delta t}\left[\mathsf{P}(\boldsymbol{u}_{k+1}^{n+1})-\underbrace{(\boldsymbol{u}^{n}-\frac{\Delta t}{\rho}(\rho(\boldsymbol{\nabla u}^{n})\boldsymbol{u}^{n}-\rho\boldsymbol{g}+\boldsymbol{\nabla}p^{n}-\boldsymbol{f}_{k}^{n+1}))}_{\boldsymbol{u}_{k+1}^{n+1}}+\frac{\Delta t}{\rho}\boldsymbol{f}_{k}^{n+1}\right]\\
&=\boldsymbol{f}_{k}^{n+1}+\frac{\rho}{\Delta t}(\mathsf{P}(\boldsymbol{u}_{k+1}^{n+1})-\boldsymbol{u}_{k+1}^{n+1})
\end{aligned}$$

Hence, on the proposed semi-discretised form, the rigidity force is updated iteratively according to the term $(\mathsf{P}(\boldsymbol{u}_{k+1}^{n+1})-\boldsymbol{u}_{k+1}^{n+1})$, which is very similar to the artificial compressibility treatment to the pressure [124]. The iteration will finish when $||\boldsymbol{f}_{k+1}^{n+1}-\boldsymbol{f}_{k}^{n+1}||/||\boldsymbol{f}_{k}^{n}||<$ tolerance.

A similar formulation proposed by Patankar can be found in [72, 76, 77, 77, 90], which can be viewed as a special case of projection operator $\mathsf{P}(\boldsymbol{u})$. Patankar argued that the projection requires a conservation of linear and angular momentum. This will inspire the projection $\mathsf{P}(\boldsymbol{u}) = \boldsymbol{\omega} \times (\boldsymbol{x} - \boldsymbol{x}_c) + \boldsymbol{u}_c$, where the centre of mass, its velocity and its rotation are obtained by

$$M\boldsymbol{u}_c = \int_{\Omega_r} \rho \boldsymbol{u}_{k+1}^{n+1} dv, \qquad \text{linear momentum} \tag{6.4a}$$

$$\boldsymbol{J}\boldsymbol{\omega} = \int_{\Omega_r} \rho \boldsymbol{u}_{k+1}^{n+1} \times (\boldsymbol{x} - \boldsymbol{x}_c) dv, \qquad \text{angular momentum} \tag{6.4b}$$

where M, $\boldsymbol{J}$, ρ denote the mass, moment of inertia tensor and density of the rigid body. However, this projection operator requires the splitting of the spatial velocity into the velocity of the centre of mass $\boldsymbol{u}_c$ and angular velocity $\boldsymbol{\omega}$, the computation of the integral over the domain Ω_r and the inversion of the inertia tensor $\boldsymbol{J}$.

However, the requirement for the conservation of the linear momentum and angular momentum, in the author's opinion, is imprudent for two reasons. First, the HVF projection is a kinematic operation, which can not involve in mass, inertia or general dynamics. Second, in the case of the forced rigid body motion, the requirement of conservation of momentum can not hold. For example, an forced oscillation rigid body can continuously add momentum/energy into a fluid-rigid coupled system.

In this chapter, a generalised projection operator is proposed for the rigid body motion, purely from a kinematic point of view, avoiding velocity splitting, integration and inversion of the inertia tensor. Moreover, the proposed operator is easier to implement.

6.2 Helicoidal Vector Field (HVF)

In this section, an efficient linear least squares method is proposed for the projection from an arbitrary vector field to HVF. Recall the constraint Eq. (2.23) in expanded form

$$u_1 = -\omega_3 y + \omega_2 z + u_{rb,1};\ u_2 = \omega_3 x - \omega_1 z + u_{rb,2};\ u_3 = -\omega_2 x + \omega_1 y + u_{rb,3} \tag{6.5}$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} u_{rb,1} \\ u_{rb,2} \\ u_{rb,1} \end{bmatrix}.$$

The rigid body velocity field is an affine vector field with respect to the spatial coordinates $\boldsymbol{x}$. This inspires a least squares projection for obtaining $\boldsymbol{\omega}$ and $\boldsymbol{u}_{rb}$ in three dimension. In this thesis, we will focus on two dimensional cases and the unknown variable for rigid body velocity reduces to $\omega_3, u_{rb,1}, u_{rb,2}$.

6.2.1 Matrix representation

In chapters 4 and 5, the computational domain is discretised using the Marker-And-Cell method based on a staggered grid spatial discretisation. In this chapter, the same discretisation is followed. In 2−D a rigid body has three degrees of freedom and Eq. (6.5) can be written as a discretised matrix form

$$\mathsf{U}_{i+j} = \mathsf{X}_{i+j,k}\beta_k, \tag{6.6}$$

where the matrix representation of U, X and $\boldsymbol{\beta}$ are

$$\mathsf{U} = \begin{bmatrix} u_i^{A_x} \\ v_j^{A_y} \end{bmatrix}, \quad \mathsf{X} = \begin{bmatrix} -\chi_i^{A_x} y_i^{A_x} & \chi_i^{A_x} & 0 \\ \chi_j^{A_y} x_j^{A_y} & 0 & \chi_j^{A_y} \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = \begin{bmatrix} \omega_3 \\ u_{rb,1} \\ u_{rb,2} \end{bmatrix} \tag{6.7}$$

and $u_i^{A_x}, v_j^{A_y}$ is the discretised velocity component at position $(x_i^{A_x}, y_i^{A_x})$ and $(x_j^{A_y}, y_j^{A_y})$. $\chi_i^{A_x}$ is the discretised indicator function χ in the rigid body region evaluated at $(x_i^{A_x}, x_j^{A_y})$, as shown in Figure 6.1. The subindex i, j and k represent the degrees of freedom of an arbitrary velocity vector in $x-$direction, $y-$direction and the degrees of freedom for a rigid body, respectively.

Note the indicator function χ acts as a weight to the system and varies from 0 to 1, where the weight is zero outside of rigid domain Ω_s.

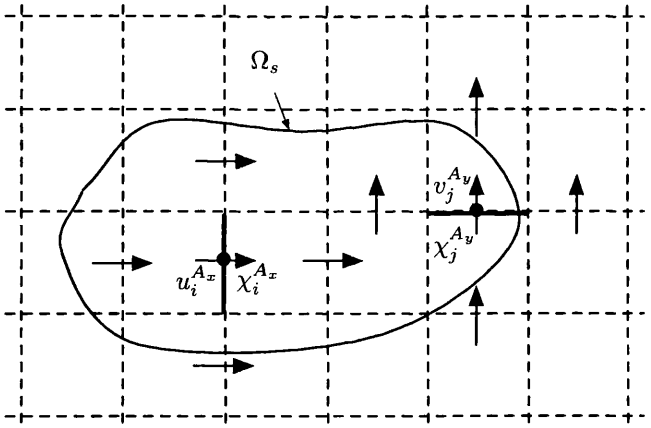


Figure 6.1: The rigid body velocity in the MAC grid.

Inside the rigid body region, where the indicator function is 1, we have

$$\mathsf{X} = \begin{bmatrix} -y_i^{A_x} & 1 & 0 \\ x_j^{A_y} & 0 & 1 \end{bmatrix} \tag{6.8}$$

6.2.2 Linear least squares projection

In Eqs. (6.7), the matrix X is a rectangular matrix with size of $[i+j, 3]$. Assuming there are more than two points to present the rigid region. Then, Eqs. (6.7) becomes an overdetermined system with three unknown coefficients. The unknown $\boldsymbol{\beta}$ can be solved in a linear least squares sense as

$$\mathsf{X}^T\mathsf{X}\boldsymbol{\beta} = \mathsf{X}^T\mathsf{U}. \tag{6.9}$$

Then, the least squares approximation of the rigid velocity field at discrete level U_r will be

$$\mathsf{U}_r = \mathsf{X}\boldsymbol{\beta} = \underbrace{\mathsf{X}(\mathsf{X}^T\mathsf{X})^{-1}\mathsf{X}^T}_{\mathsf{P}}\mathsf{U}, \tag{6.10}$$

where we define $\mathsf{P} = \mathsf{X}(\mathsf{X}^T\mathsf{X})^{-1}\mathsf{X}^T$ to be the projection operator from U to U_r. On the discretised level, the projection operator is given and it is only related to the rigid body position X.

In the case of a fixed rotation around a point (x_0, y_0), the Eqs. (6.7) is reduced to

$$\mathsf{X} = \begin{bmatrix} -\chi_i^{A_x}(y_i^{A_x} - y_0) \\ \chi_j^{A_y}(x_j^{A_y} - x_0) \end{bmatrix}, \quad \boldsymbol{\beta} = \omega_3 \tag{6.11}$$

where $\boldsymbol{\beta}$ becomes a scalar value ω_3, which is the angular velocity. The rigid velocity U_r is calculated in the same way as in Eq.(6.10).

In the case of a given velocity field U_0, the projection velocity U_r is the given velocity as

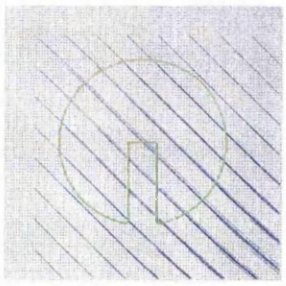
$$\mathsf{U}_r = \mathsf{U}_0. \tag{6.12}$$

The formulation above permits the simulation of several 2-D motions including the fixed in space, only allowed rotation, or free rigid body motion. It can be observed that different rigid body motions leads to different constructions of the matrix X. Moreover, the proposed methodology is extendable for the multibody dynamics.

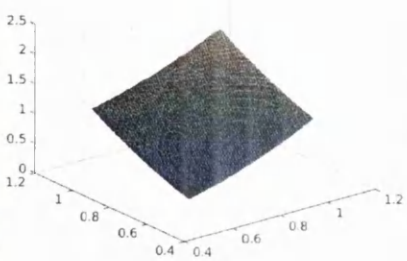
6.2.3 Examples of the HVF projection

Let us consider the case where the rigid body region has the shape of a Zalesak Disk[1] and moves in a medium with velocity given by $\boldsymbol{u}(x, y) = (x^2 + y; -2xy)$ which is a Solenoidal Vector Field (SVF) but not a Helicoidal Velocity Field (HVF). After a linear least square fitting, the resulting velocity field is shown in Figure 6.2. Two cases have been considered, free rigid motion and allowed rotation around the point $(0.75, 0.75)$. The computational domain is $(0, 1) \times (0, 1)$ with a meshsize of 100×100.

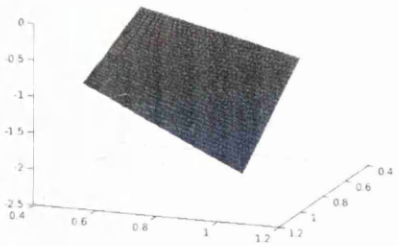
[1]The Zalesak Disk is often used for testing the level set method, which originally introduced by Zalesak [20]



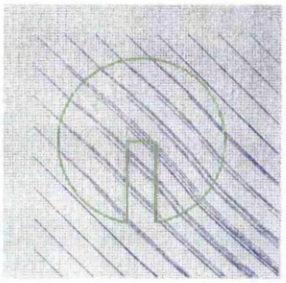
(a) Initial divergence free vector field



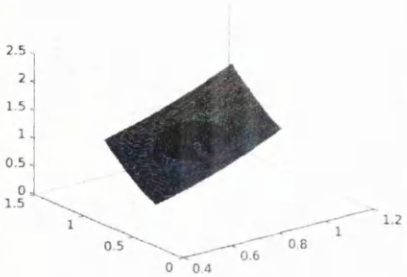
(b) Initial velocity profile in x- diection



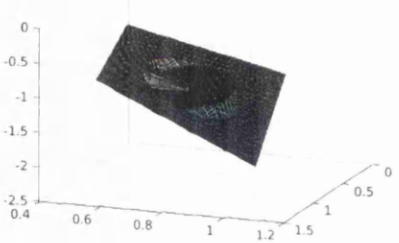
(c) Initial velocity profile in y- direction



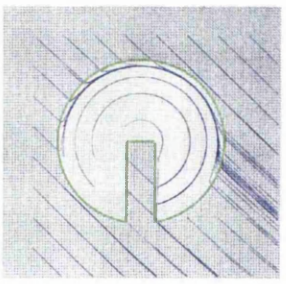
(d) Projected rigid velocity field



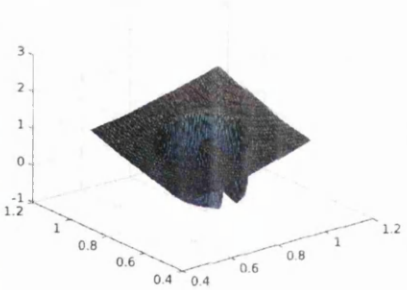
(e) Projected rigid velocity profile in x- direction



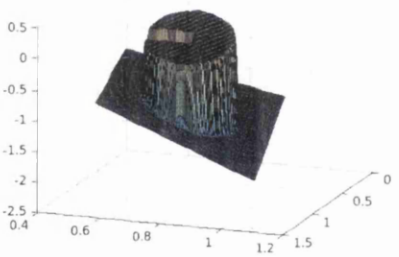
(f) Projected rigid velocity profile in y- direction



(g) Allowed rotation with $(0.75, 0.75)$.



(h) Allowed rotation with $(0.75, 0.75)$. Projected rigid velocity profile in x- direction



(i) Allowed rotation with $(0.75, 0.75)$ Projected rigid velocity profile in y- direction

Figure 6.2: Least squares projection for a rigid body velocity from a divergence free field. The velocity field inside the region body has been changed. From the divergence free field (Solenoidal Vector Field) to a rigid body motion (Helicoidal Velocity Field).

6.2.4 Indicator function

The indicator function can be constructed from the Level Set function in Section (4.2.1) or the integration points in Section (5.2.3). However, we do not use the convection Eq. (3.32) for the convection of the Level Set. Instead, the Level Set is advected in a Lagrangian way. For example, the Level Set function for describing a

sphere in two dimensions can be expressed as

$$\phi(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{x}_0||^2 - r^2 \tag{6.13}$$

where $\boldsymbol{x}_0$ and r are the centre of the sphere and radius of the sphere, respectively. To keep the shape of the rigid body, the centre of mass is moved passively with the velocity rather than the whole Level Set function as

$$\frac{\mathrm{d}\boldsymbol{x}_0}{\mathrm{d}t} = \boldsymbol{u}(\boldsymbol{x}_0) \tag{6.14}$$

At this stage, the Level Set function is limited to a simple geometry, such as a circle or rectangle, which has a theoretical description so that the Level Set function can be advected in a Lagrangian way.

6.3 Algorithm

In this section, a flowchart of the overall fluid rigid body interaction algorithm employed in this thesis is included to illustrate the various steps of the methodology presented here. By using a simple fixed point iterative scheme, the coupled fluid rigid body interaction algorithm can be resolved to advance from time step n to $n+1$ in an iterative fashion to ensure the rigid body kinematics constraint is fully satisfied. By computing the residual norm based upon the difference between the rigid body immersed forces in two successive iteration k and $k+1$, namely $||\boldsymbol{f}^n_{k+1} - \boldsymbol{f}^n_k||/||\boldsymbol{f}^n||$, a convergence criterion can be easily established in order to progress to the following iteration $k+2$ (if not yet converged) or time step $n+2$ (if convergence is accomplished).

Algorithm 6.1 : IMMERSED FLUID AND RIGID BODY INTERACTION SOLVER

- Initialise the fluid velocity $\boldsymbol{u}_0$, indicator function $\boldsymbol{\chi}$ from level set $\boldsymbol{\phi}$ or from integration point $\boldsymbol{x}_{a_p}$ for the rigid body
- Loop over time: while fixed-point iteration $||\boldsymbol{f}_{k+1}^{n+1} - \boldsymbol{f}_k^{n+1}||/||\boldsymbol{f}_k^n|| > tolerance$
  - Compute an intermediate velocity field $\boldsymbol{u}^\star$ by advancing the unified 'one-phase' momentum equation as follows

$$\int_\Omega \frac{\boldsymbol{u}^\star - \boldsymbol{u}^n}{\Delta t}\, dv = -\int_\Omega \left(\mathrm{Conv}(\boldsymbol{u})^{n+\frac{1}{2}} + \frac{\mu(\boldsymbol{\chi})_k^{n+1}}{\rho(\boldsymbol{\chi})_k^{n+1}} \Delta \boldsymbol{u}^n + \boldsymbol{f}_k^{n+1}(\boldsymbol{\chi}) \right) dv$$

    where the convective terms are computed using the Adams-Bashforth formula
  - Compute the new pressure to satisfy the incompressible constraint with Neumann boundary condition

$$\int_\Omega \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla} p_k^{n+1}}{\rho(\boldsymbol{\chi})_k^{n+1}} \right) dv = -\int_\Omega \frac{\boldsymbol{\nabla} \cdot \boldsymbol{u}^\star}{\Delta t}\, dv$$

  - Apply velocity correction

$$\int_\Omega \boldsymbol{u}_{k+1}^{n+1}\, dv = \int_\Omega \boldsymbol{u}^\star\, dv - \int_\Omega \frac{\Delta t}{\rho(\boldsymbol{\chi})_k^{n+1}} \boldsymbol{\nabla} p_k^{n+1}\, dv$$

  - Evaluate the rigidity force $\boldsymbol{f}_{k+1}^{n+1}$
    1. **Projecting** the velocity field into a HVF

$$\boldsymbol{u}_{k+1}^{n+1} = \mathsf{P}(\mathsf{U}_{k+1}^{n+1})$$

    2. **Updating** the immersed rigid body force as

$$\boldsymbol{f}_{k+1}^{n+1} - \boldsymbol{f}_k^{n+1} = \frac{\rho(\boldsymbol{\chi})}{\Delta t} \left(\mathsf{P}(\boldsymbol{u}_{k+1}^{n+1}) - \boldsymbol{u}_{k+1}^{n+1} \right)$$

  - Evaluate indicator function by convecting level sets/particles using $\boldsymbol{u}_{k+1}^{n+1}$.
  - Update the rigid body location, the indicator function $\boldsymbol{\chi}_{k+1}^{n+1}$, density $\rho(\boldsymbol{\chi})_{k+1}^{n+1}$
- Update velocity $\boldsymbol{u}^{n+2}$, density $\rho(\boldsymbol{\chi})^{n+2}$ to the next time step

Part IV

Numerical Examples

Chapter 7

Multiphase fluid

In this chapter, a list of benchmark problems will be presented in order to demonstrate the robustness and applicability of the multiphase flow solver. These problems include the prototypical dam break problem, some bubble dynamics simulations, the three-phase water mixing problem and the splash of a droplet.

7.1 Classical dam break problem

The dam problem is a well documented problem [15, 125–129] which simulates the sudden collapse of a rectangular column of water onto a horizontal surface. The main characteristic of the problem is the simulation of the transient flow of two fluids separated by a sharp interface, where surface tension effects are disregarded. Since the fluid solver is immersed based, the physical properties of the two fluids across the sharp interface are smoothed. This problem has been studied at an experimental [125, 126] and numerical [15, 127–129] levels. For the calculations presented herein,

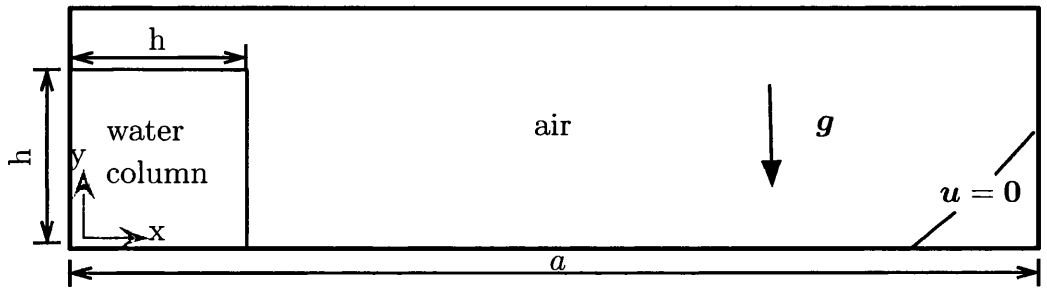
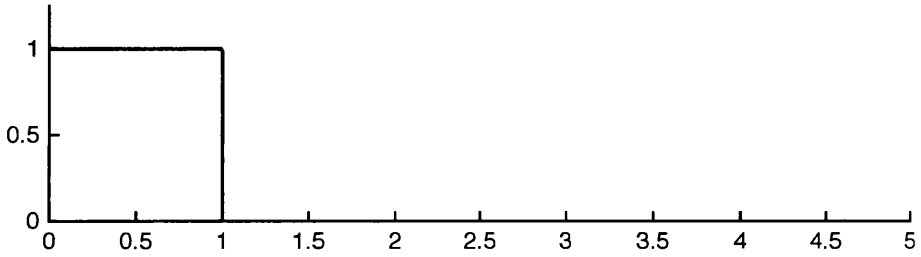


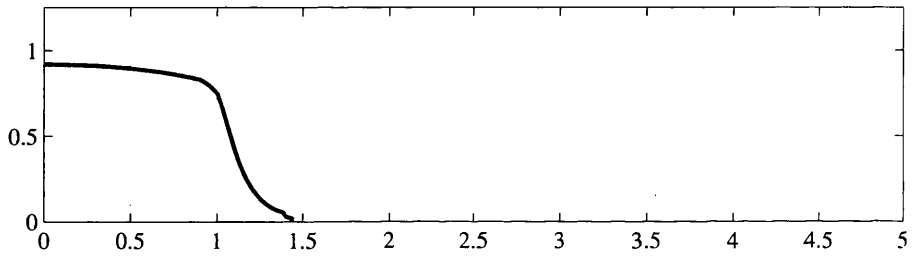
Figure 7.1: Schematic of the initial water column ($t = 0$) for the dam break problem ($h = 1$; $a = 5$).

the initially prescribed height of the water column is $h = 1$ and the base length is $a = 5$. The fluid properties (water and air will be referred by the subscripts w and a, respectively) are listed in Table 7.1 below.

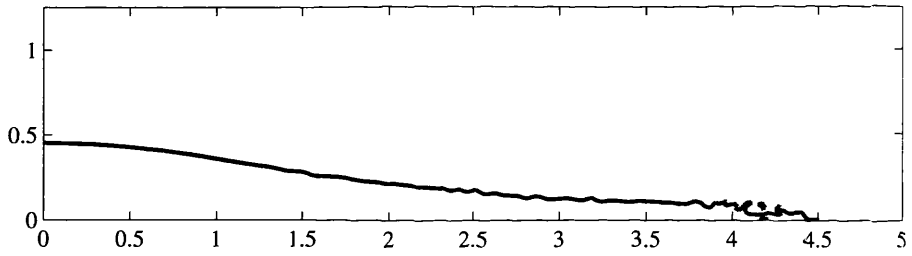
Figure 7.2 displays a series of snapshots of the free surface position as a function of time. The predicted heights and the surge front location of the collapsed water are



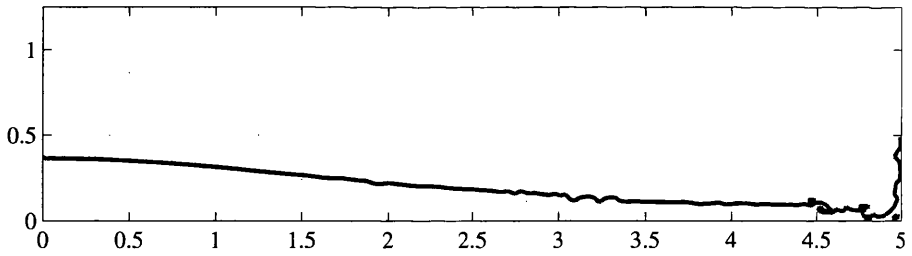
(a) $t = 0.0$



(b) $t = 1.25$



(c) $t = 2.5$



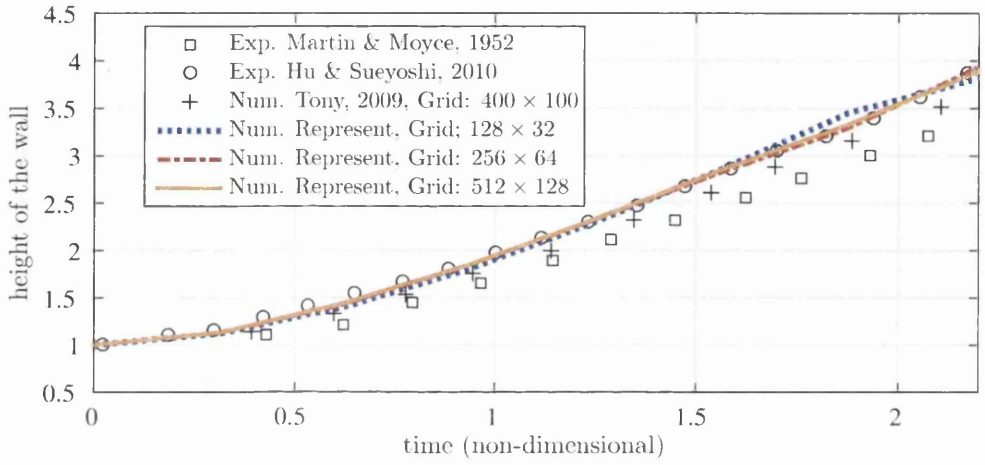
(d) $t = 3.0$

Figure 7.2: Comparison of the predicted free surfaces at different time steps, obtained for a mesh of 512 × 128 cells, (a) t= 0.0; (b) t = 1.25; (c) t = 2.5; (d) t = 3. The material parameters are as follows: water density $\rho_w = 1000$, air density $\rho_a = 1$, water viscosity $\mu_w = 10^{-3}$, air viscosity $\mu_a = 10^{-5}$.

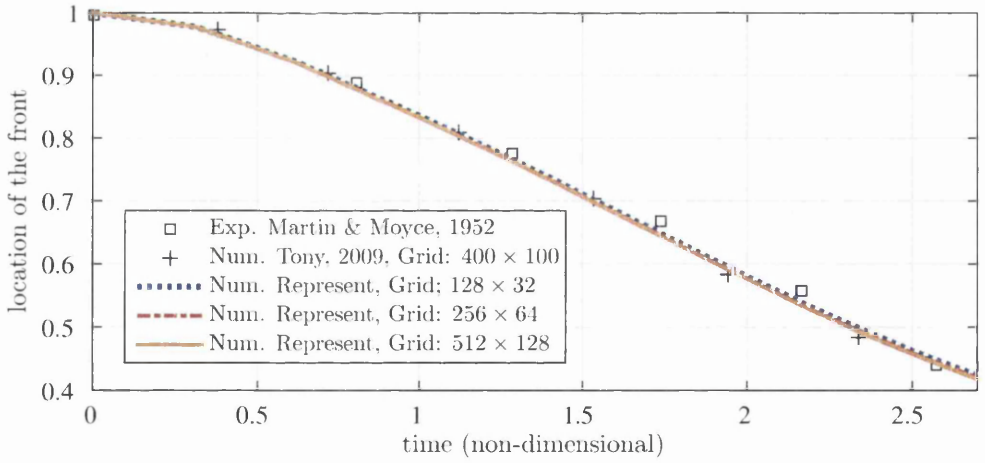
water density	air density	water viscosity	air viscosity
$\rho_w = 1000$	$\rho_a = 1$	$\mu_w = 10^{-3}$	$\mu_a = 10^{-5}$

Table 7.1: Physical properties of water and air for the classical dam break problem.

plotted against the dimensionless time $\tau = t\sqrt{h/g}$ as defined in [125]. In Figure 7.3, very good agreement is displayed between the numerical simulation and the latest experimental result [126] and other numerical results available in the literature. For comparison purposes, an alternative Finite Volume solution (Tony [129]) is shown. With mesh refinement, the presented results converge extremely well to the latest experimental data.



(a): surge front position



(b): water column height

Figure 7.3: Comparison of the surge front location and the water column height with experimental data and numerical results. The material parameters are as follows: water density $\rho_w = 1000$, air density $\rho_a = 1$, water viscosity $\mu_w = 10^{-3}$, air viscosity $\mu_a = 10^{-5}$.

7.2 Bubble rising in a partially filled container

The second problem is a circular bubble rising in a partially filled container. A bubble made up of a fluid phase termed 'fluid 2' rises in a container partially filled with another fluid phase termed 'fluid 1', as shown in Figure 7.4. The two-fluid system has a density ratio of 2 : 1. Initially, the perfectly circular bubble with a diameter of D, is immersed inside a container filled 'fluid 1' with depth equal to $2.5D$, leaving an empty layer of height D above in the container. The centre of the bubble is located at depth D below the free surface. The Reynolds number (Re) and the Weber number (We) are defined in Reference [15] as

$$Re = \frac{\rho_1 \sqrt{g} D^{1.5}}{\nu_1}, \quad We = \frac{\rho_1 g D^2}{\gamma} \tag{7.1}$$

where $\rho_1(\rho_2)$ is the density for of 'fluid 1'('fluid 2'), g is the gravitational acceleration, D is the diameter of the bubble fluid 2, $\nu_1(\nu_2)$ is the kinematic viscosity of 'fluid 1'('fluid 2') and γ is the surface tension coefficient. In the following, two cases are considered. In what follows, case 1 does not consider the surface tension effect whilst case 2 considers the effect of surface tension. The computational domain is defined by a rectangle $\Omega = [0.12 \times 0.14]$. The time sequence of the rising bubble is shown in

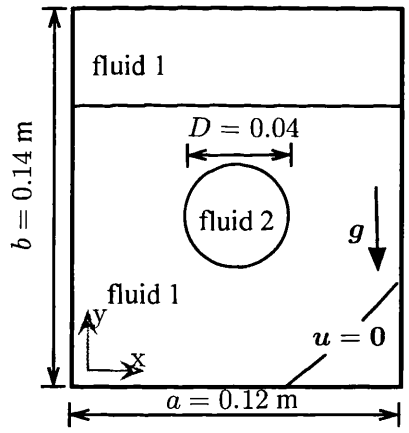


Figure 7.4: Schematic diagram of a two-dimensional container partially filled with fluid 1 and 2.

Test case 1	ρ_1/ρ_2	ν_1/ν_2	Re	We
1	2 : 1	2 : 1	200	0
2	2 : 1	2 : 1	200	10

Table 7.2: Dimensionless physical properties of two fluids for the test case of a bubble rising in a partially filled container.

Figure 7.5 (without surface tension) and Figure 7.6 (with surface tension). As it can be observed, a good agreement is shown with the results predicted by Reference [15].

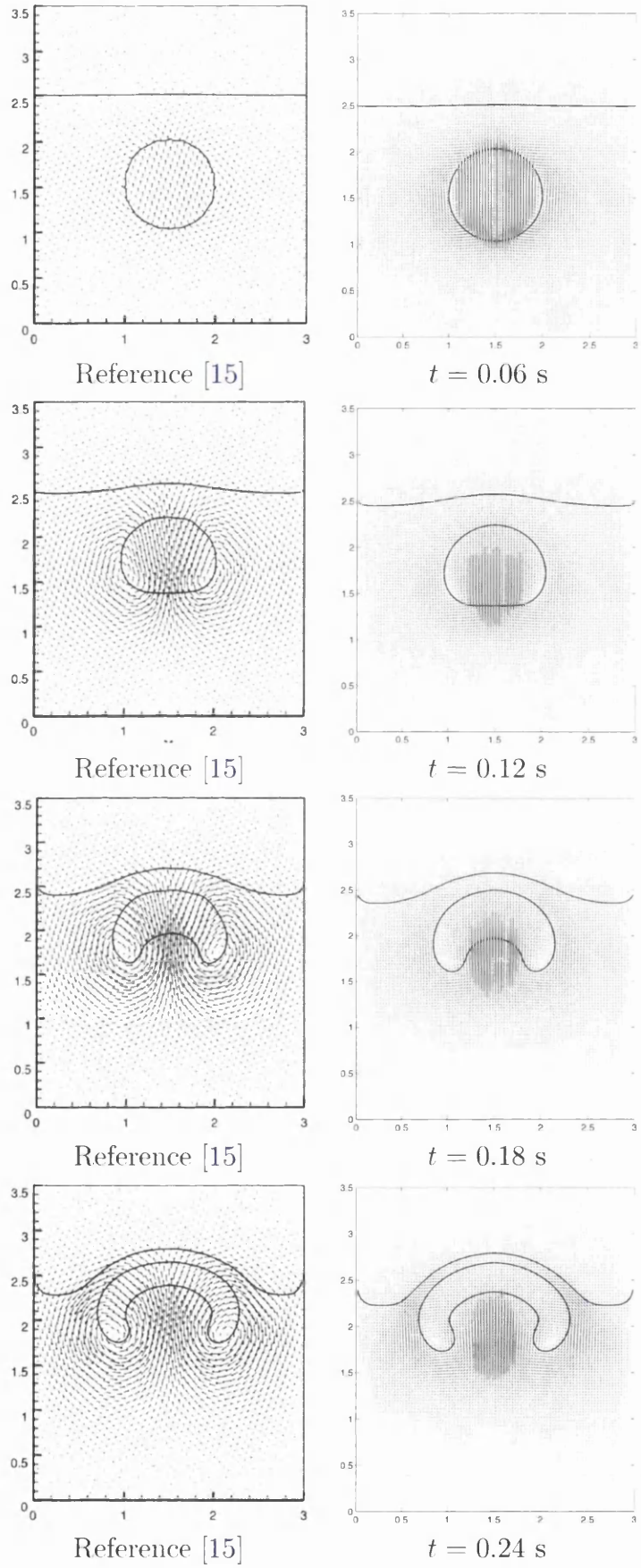


Figure 7.5: Time evolution of fluid 2 rising in a partially filled container. ($Re = 200$, $We = 0$). Left: Reference solution [15], right: proposed method.

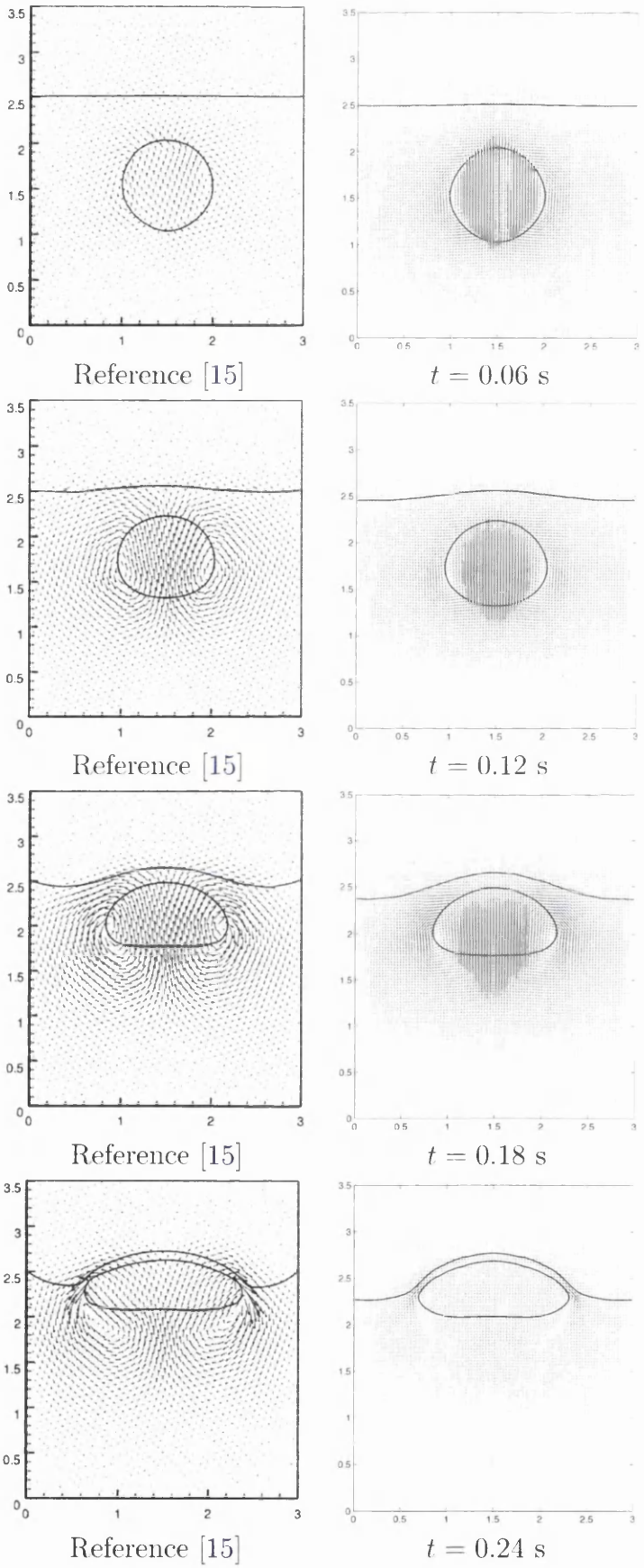


Figure 7.6: Time evolution of fluid 2 rising in a partially filled container. ($Re = 200$, $We = 10$) Left: Reference solution [15], right: proposed method.

7.3 Bubble rising in a fully filled container

In this third example, we consider a bubble rising in a fully filled container. This quantitative validation for a two-phase problem is proposed in [130]. The initial configuration consists of a circular bubble of radius $r = 0.25$ centred at $[0.5, 0.5]$ in a rectangular domain $\Omega = [1 \times 2]$. The density of the bubble is smaller than that of the surrounding fluid ($\rho_2 < \rho_1$). The non-slip boundary condition ($\boldsymbol{u} = \boldsymbol{0}$) is used at the top and bottom boundaries, whereas the free slip condition $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ is imposed on the vertical walls (being $\boldsymbol{n}$ the outward unit normal). Computations were conducted until $t = 3$. In order to benchmark the numerical solution, a series

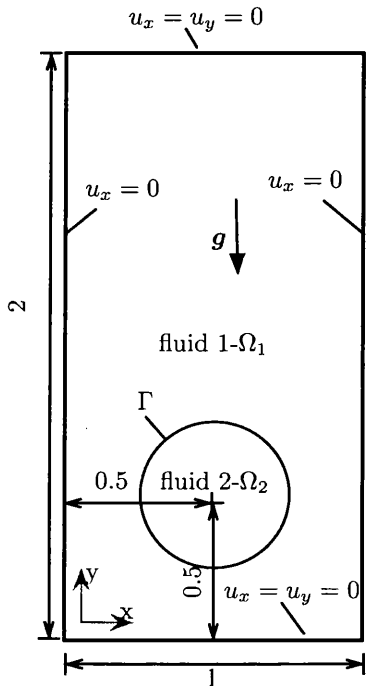


Figure 7.7: Initial configuration and boundary conditions for the test cases.

Test case	ρ_1	ρ_2	μ_1	μ_2	g	γ	Re	E_o	ρ_1/ρ_2	μ_1/μ_2
Case 1	1000	100	10	1	0.98	24.5	35	10	10	10
Case 2	1000	1	10	0.1	0.98	1.96	35	125	1000	100

Table 7.3: Dimensionless physical property of two fluids for the test case of a bubble rising in a fully filled container.

of physical magnitudes of interest are introduced [130], including the centre of mass of the bubble, its mean rising velocity and the so-called circularity of the bubble (to be defined below).

- **Centre of mass**: The position of the mass is used to track the translation of the bubble as a function of time, and is defined as

$$\boldsymbol{x}_c = \frac{\int_{\Omega_2} \boldsymbol{x}\, dv}{\int_{\Omega_2} dv}. \tag{7.2}$$

- **Circularity**: The so-called 'degree of circularity' of the evolving bubble can be defined as

$$C = \frac{P_a}{P_b} = \frac{\text{perimeter of the area-equivalent circle}}{\text{perimeter of the bubble}} = \frac{\pi D_a}{P_b}, \tag{7.3}$$

where P_a denotes the perimeter of a circular bubble with diameter D_a, which has an area equal to that of a deformed bubble with perimeter P_b. For the initial circular bubble, the circularity is equal to one and it then decreases as deformation of the bubble increases.

- **Rising** or **terminal velocity** defined as

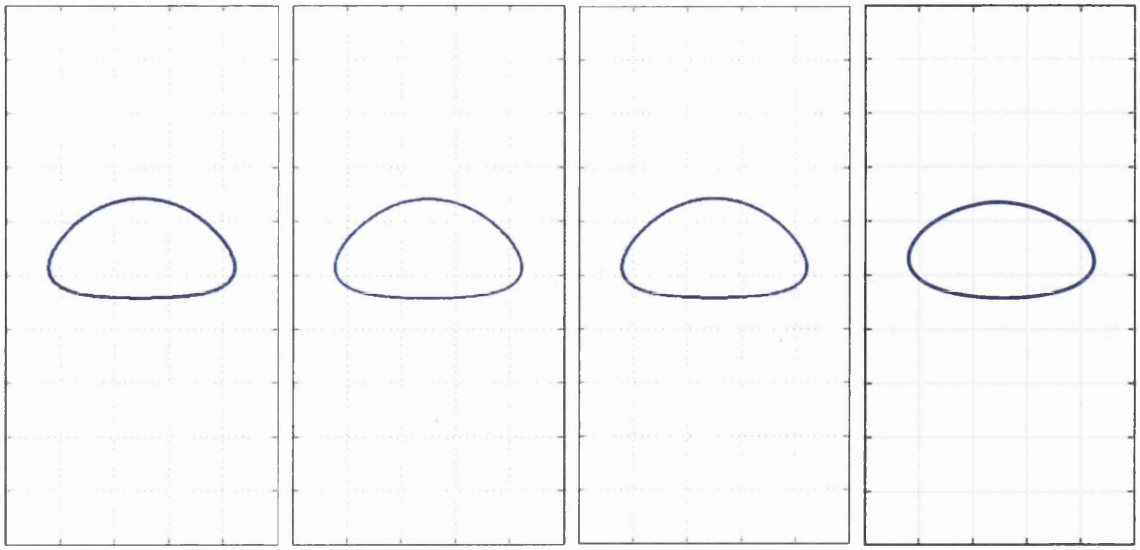
$$\boldsymbol{u}_c = \frac{\int_{\Omega_2} \boldsymbol{u}\, dv}{\int_{\Omega_2} dv}. \tag{7.4}$$

The above is the mean velocity with which a bubble is rising or moving is a particularly interesting parameter because it measures not only the behaviour of the interface tracking algorithm but also the quality of the overall solution.

Three different of codes are compared with the proposed method. The TP2D (Transport Phenomena in 2D) code is a FEM based incompressible flow solver with the Level Set method describing the interface, developed in the Technical University of Dortmund [131]. The FreeLIFE (Free-Surface Library of Finite Element) software is a FEM based incompressible flow solver developed in EPFL Lausanne [130]. The MooNMD (Mathematics and object oriented Numerics in MagDeburg) is a FEM based incompressible solver using the arbitrary Lagrangian-Eulerian (ALE) approach developed in Otto-von-Guericke University Magdeburg [132].

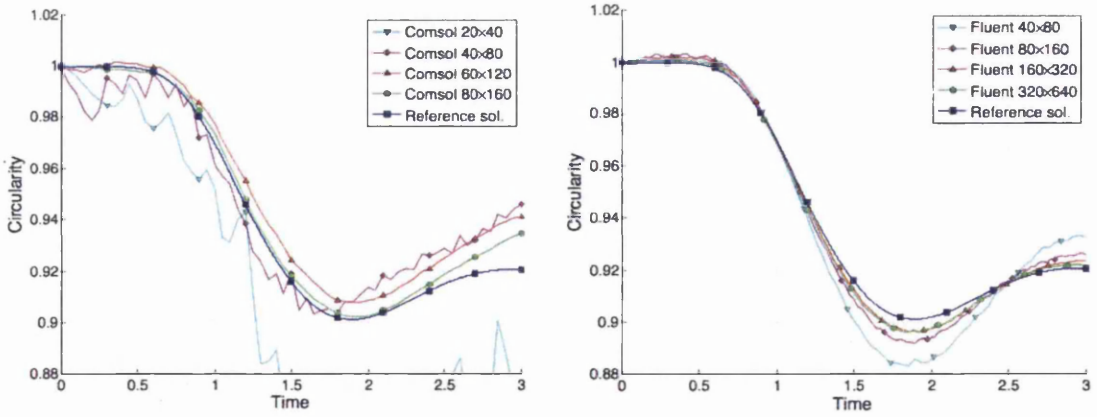
Case 1: The solution for test case 1 in comparison with the other three open-source FEM based softwares (FreeLIFE, MooNMD, TP2d) is presented in Figure 7.8. The bubble, being initially circular, is stretched horizontally and first develops a dimple as it rises, but after some time proceeds to assume a more stable ellipsoidal shape.

No significant differences can be observed from the proposed FV based solver and the other reference FEM solvers. However, Reference [16] stated that commercial software, such as Comsol Multiphysics [133] and Ansys Fluent [134] did not show strong convergence towards the reference solution, especially in the case of measuring the circularity magnitude, as shown in Figure 7.9.



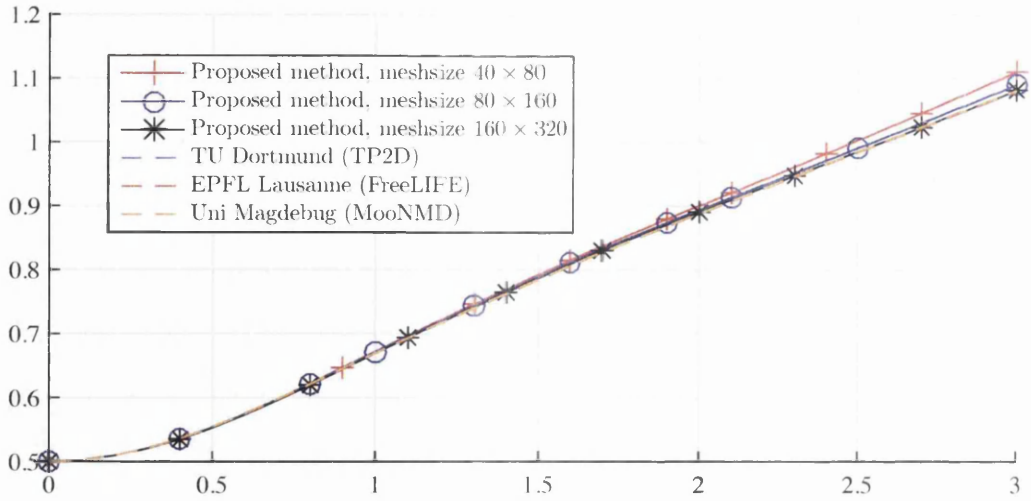
(a) FreeLIFE (b) MooNMD (c) TP2d (d) Proposed method

Figure 7.8: Comparison of the predicted free surfaces, from referenced codes and presented method, with the finest mesh size 160×320.

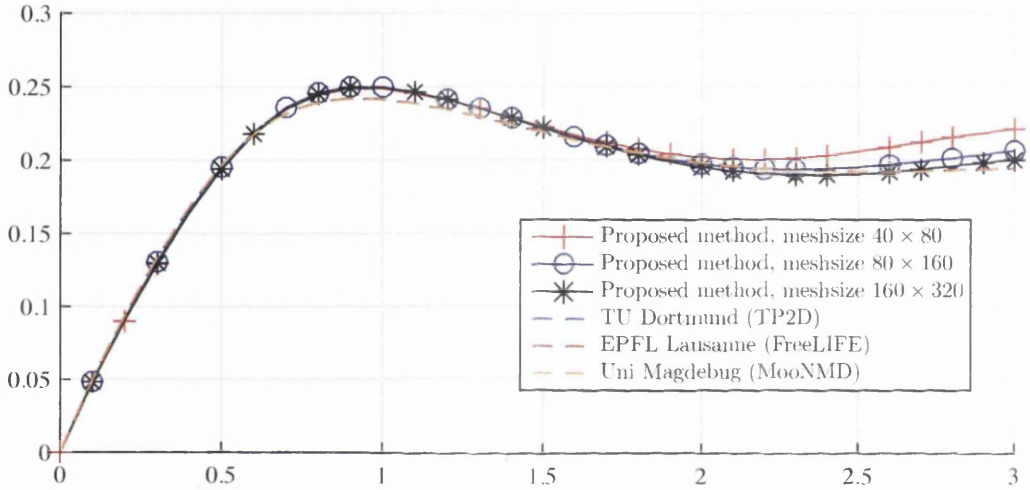


(a) Comsol Multiphysics [133] (b) Ansys Fluent [134]

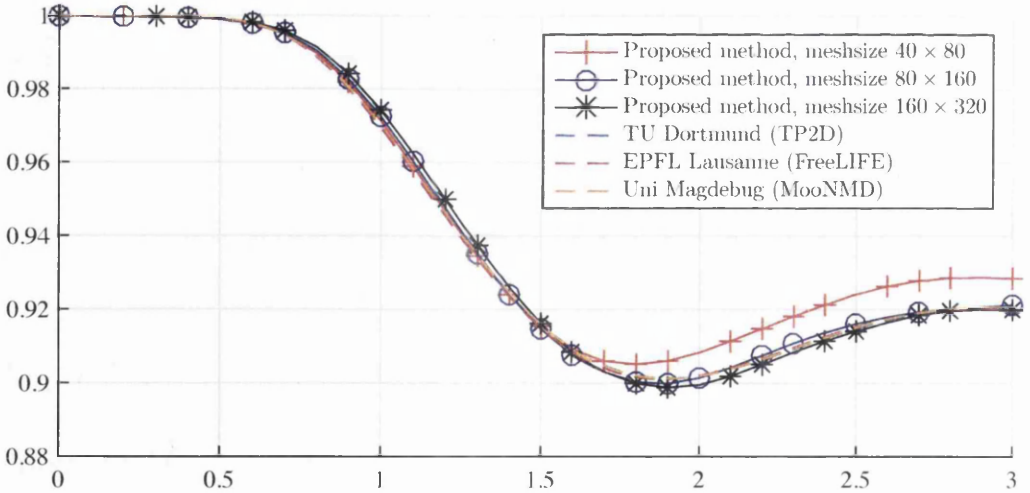
Figure 7.9: Quantitative comparison of the circularity for commercial softwares, Comsol Multiphysics and Ansys Fluent, reproduced from [16]



(a) Centre of the mass



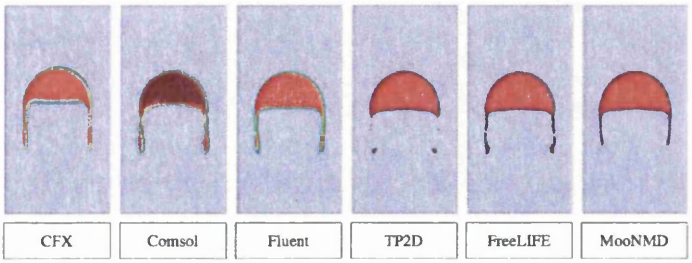
(b) Rising velocity



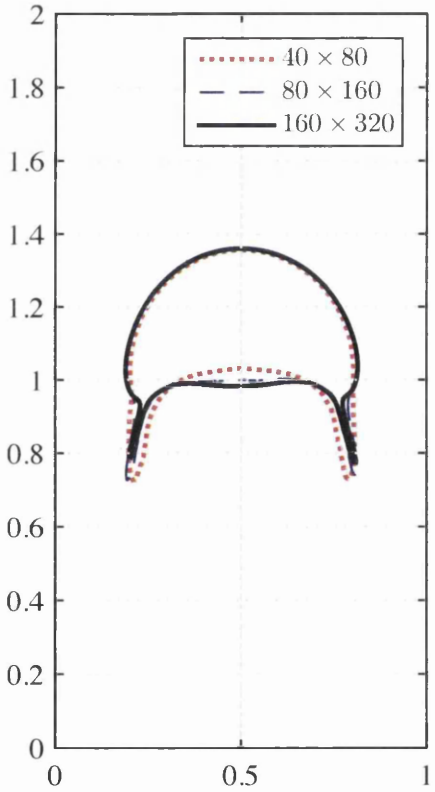
(c) Circularity

Figure 7.10: Quantitative comparison for the centre of mass, circularity and rise velocity for the bubble rising problem of case 1. The simulation is performed with three levels of refinement, 40×80, 80×160 and 160×320.

Case 2: Figure 7.11 shows the final shape of the bubble on a three set of mesh refinement. Although the bubble in both test cases rises with approximately the same speed, the decrease in surface tension causes this bubble to assume a more non-convex shape and develop thin filaments, which might eventually break off. The result of the six different codes[1] is taken from Reference [130], as shown in Figure 7.11. There is no agreement with respect to the thin filamentary regions. The TP2D and FreeLIFE codes show a break up of the bubble while CFX, Comsol, MooNMD and the proposed method show that the long thin trailing filaments remain intact.



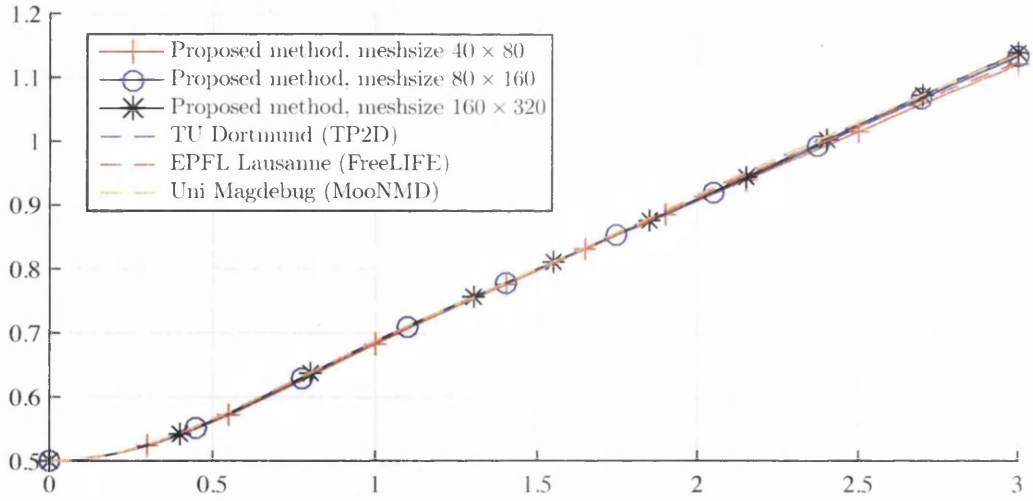
(a) Six different codes, reproduced from [130]



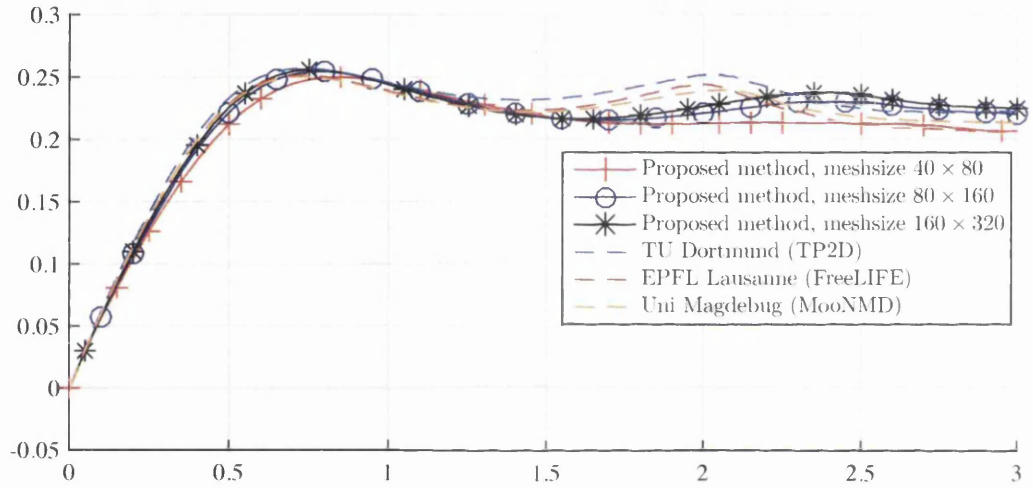
(b) Proposed method

Figure 7.11: Numerical simulations of a two-dimensional rising bubble for six different codes and presented method. The simulation is performed with three levels of refinement: 40×80, 80×160 and 160×320.

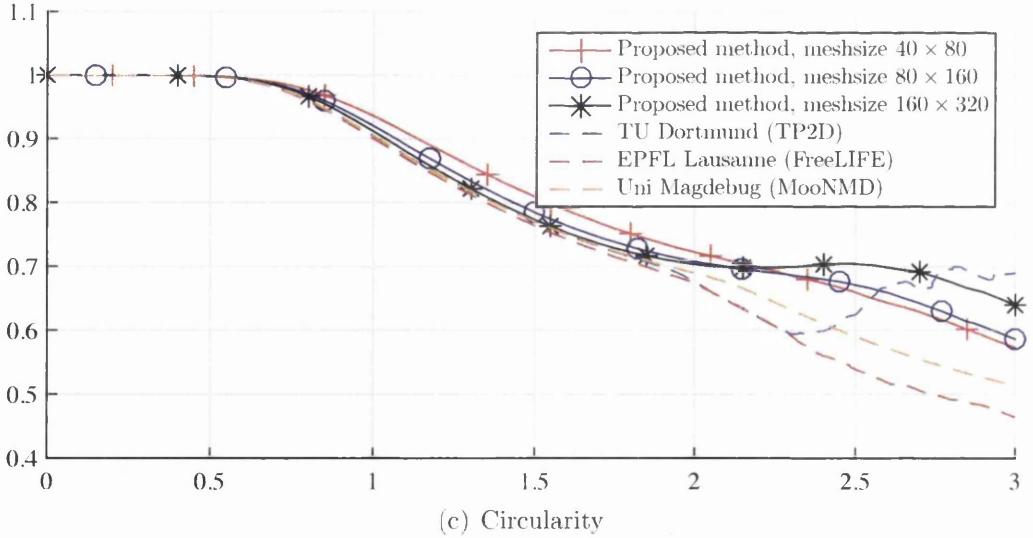
[1]CFX are commercial CFD software



(a) Centre of the mass



(b) Rising velocity



(c) Circularity

Figure 7.12: Quantitative comparison for the centre of mass, circularity and rise velocity for the bubble rising problem of case 2. The simulation is performed with three levels of refinement: 40×80, 80×160 and 160×320.

7.4 Dam break with mixing processes on a web bed

Three phase flow problems are rarely solved using the Level Set method [97], because the classical Level Set method can only represent two phases. The multiple level set technique makes the multi-phase flow simulation possible. The first case considered is a dam break with multiple phases interacting and mixing. This problem has been studied by means of experimental methods [135] and the Smoothed-Particle Hydrodynamics (SPH) method [17, 136–138].

The experimental setup is shown in Figure 7.13, which includes a water tank operated from the downstream channel by a gate. Water with a depth of 0.015m is blocked with a gate. The downstream channel contains a PEO (polyethylene-oxide) solution. According to the experiment [135], the solution's viscosity is 0.9835×10^{-6} m^2/s and has a density close to that of the water. In the experiment, the vertical

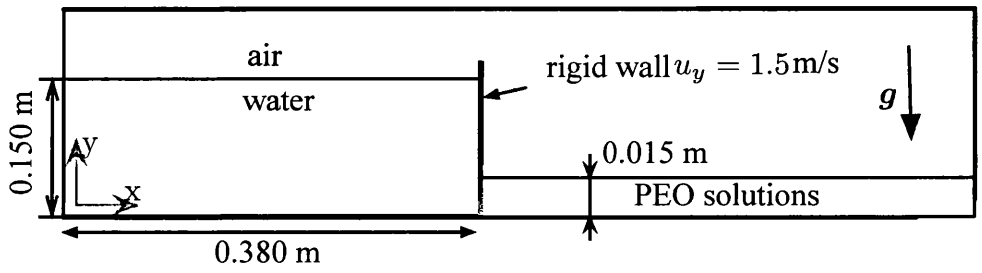


Figure 7.13: Geometry of the three-phase dam break problem.

gate is removed at a constant velocity of $u_y = 1.5$ m/s. The fluid is then free to fall under the gravitational force. The gate must be removed gradually in the numerical simulation, and it has a significant effect on the results [17] .

The rectangular computational domain is set to be $\Omega = [1 \times 0.2]$ with a computational mesh size of 320×64 and time step $\Delta t = 3 \times 10^{-4}$s. Figure 7.14 compares the numerical and experimental configuration after $t = 0.13$ s. The comparisons show that the numerical simulation appropriately illustrates the relevant features of the process at different stages. The gate has been pulled up gradually, similarly to the experiments. As the figure shows, the gradual removal of the gate causes a reserve flow at the crest of the wave. The SPH results are also shown in Figure 7.15 for comparison. These results illustrate the ability of this method to simulate the flow features and the mixing process in a multiphase dam break system.

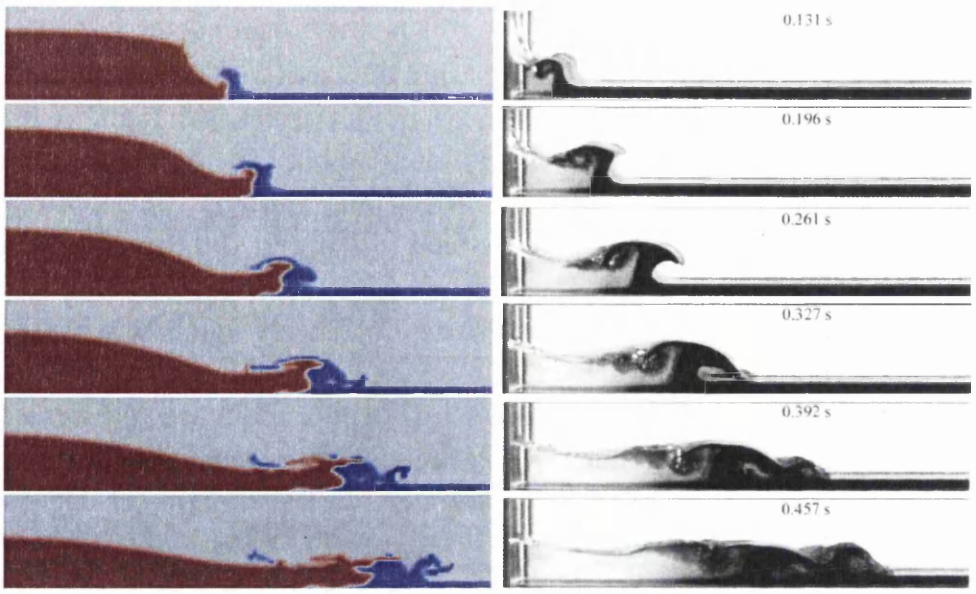


Figure 7.14: Comparison of the proposed numerical results (left) and experimental results for the dam break with mixing processes on a wet bed (right). It involves three phases flows and a rigid wall removed gradually.

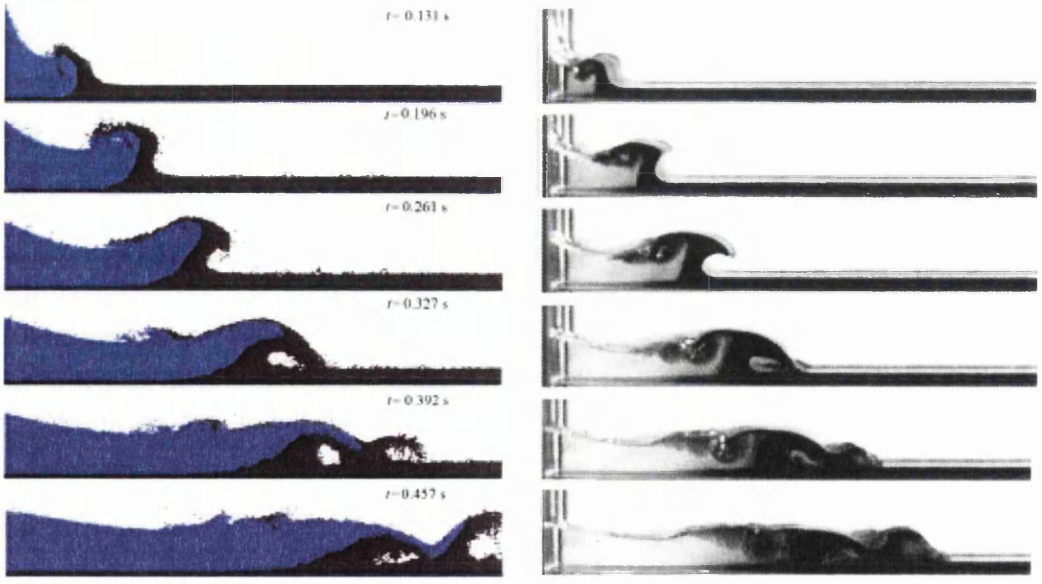


Figure 7.15: Comparison of the numerical and experimental results using SPH [17].

7.5 Impact of a liquid on a thin layer

The liquid impact is a very popular problem involving strong deformation of liquid surfaces. The present example concentrates on the early times of droplet impact on a thin liquid layer. The liquid rises around the impact point through a circular, approximately vertical and expanding sheet, called a 'corolla'. Figure 7.5 shows the initial conditions for the simulation. Liquid and gas are referred by the subscripts

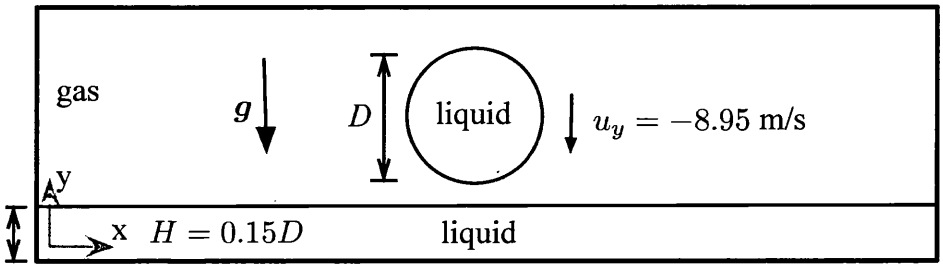


Figure 7.16: The initial conditions for a droplet splashing on a thin liquid layer.

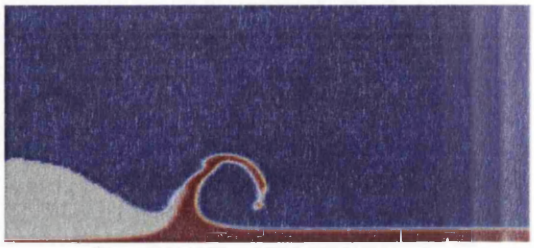
l and s, respectively. $\rho_l/\rho_g = 500$, $\mu_l/\mu_g = 20$. The droplet diameter $D = 6$ mm. Here the Weber and Reynolds numbers for the liquid are [139, 140]

$$We = \frac{\rho_l u_y^2 D}{\gamma}, \quad Re = \frac{\rho_l u_y D}{\mu_L}. \tag{7.5}$$

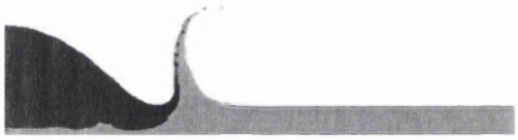
The surface tension Weber number is chosen to be $We = 8000$ where $\gamma = 0.065$ N/m, and the density is $\rho_l = 10^3$ kg/m^3. The drop parameter and thin layers $H/D = 0.15$, with a drop velocity $u_y = -8.95$ m/s.

A similar problem has been studied via experimental methods [141–145] and numerical simulations [139, 140, 146] using the Volume Of Fluid method modelling the two phases. Although it is a two phase flow problem, the liquid drop and the liquid layer on the bottom of the tank is treated as two different fluids with the same material property using the multiple Level Set technique, in order to identify the pseudo interface after merging.

The computational domain is set to be 36 × 12 mm with a mesh grid of 270 × 90. The droplet is launched above the liquid layer with a height of 6 mm. Good agreement with the results shown in Josserand and Zaleski [140] using the Volume Of Fluid are displayed in Figure 7.17.



(a) Present results, for $Re = 1000$



(b) Josserand and Zaleski [140], for $Re = 1000$

(c) Present results, for $Re = 100$

(d) Josserand and Zaleski [140], for $Re = 100$

(e) Present results, for $Re = 40$

(f) Josserand and Zaleski [140], for $Re = 40$

Figure 7.17: Comparison result of the liquid impact on a thin layer, with three-phases flow and simulation mesh size for 270×90.

Chapter 8

Flexible solid

The aim of this chapter is to present a series of benchmark problems to test the unified 'one-phase' framework for FSI problems. The single-phase FSI problem has already been handled by classical Immersed Boundary Methods and, therefore, the examples in this chapter will be chosen for different density ratio and hydrodynamic multiphase coupled problems, in order to show the strength of the proposed methodology.

8.1 One flapping membrane

Let the channel be an idealised two-dimensional channel $\Omega = [4 \times 1.61]$, which we will consider to be filled with an incompressible Newtonian viscous fluid with viscosity $\mu = 1$ and density $\rho_f = 100$. A leaflet is inserted into the channel, as seen in Figure 8.1. The top and bottom boundaries of the channel are fixed, a pulsatile non-reversible inflow is applied at the left hand boundary using a time-varying amplitude $A(t) = 5(\sin(2\pi t + 1.1))$. The constitutive model of the leaflet is an incompressible neo-Hookean material model with shear modulus of $G = 2.0 \times 10^7$. Three cases with different structure densities $\rho_s = 50, 100, 200$ will be simulated. Because there are no inertial or buoyancy effects, the results of different cases are fairly similar. The

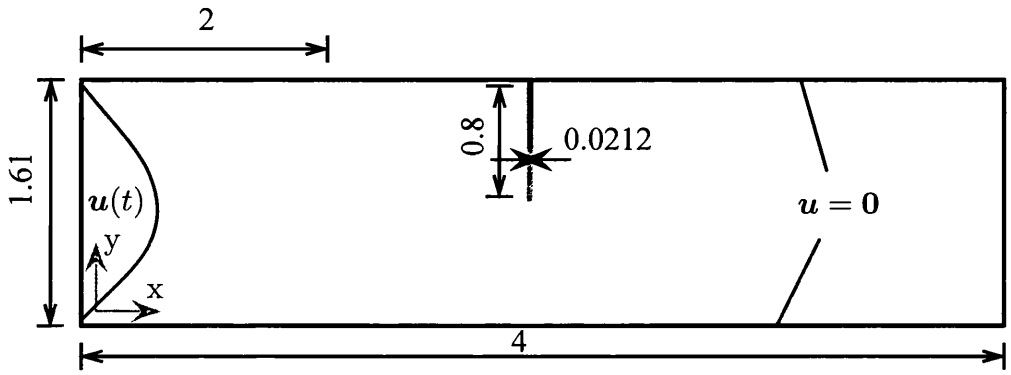


Figure 8.1: Geometry and boundary conditions for an idealised leaflet valve

series of diagrams in Figure 8.2 show the time evolution for the pulsatile flow and the deformation of the membrane using a 160×80 mesh discretisation for the fluid and 4025 solid integration points for the membrane. These results show that this method is able to successfully model the inclusion of highly deformable structures into the fluid with large density ratio without the need to resort to an expensive implicit time-stepping algorithm. The present methodology allows for a very simple and robust treatment of the structure.

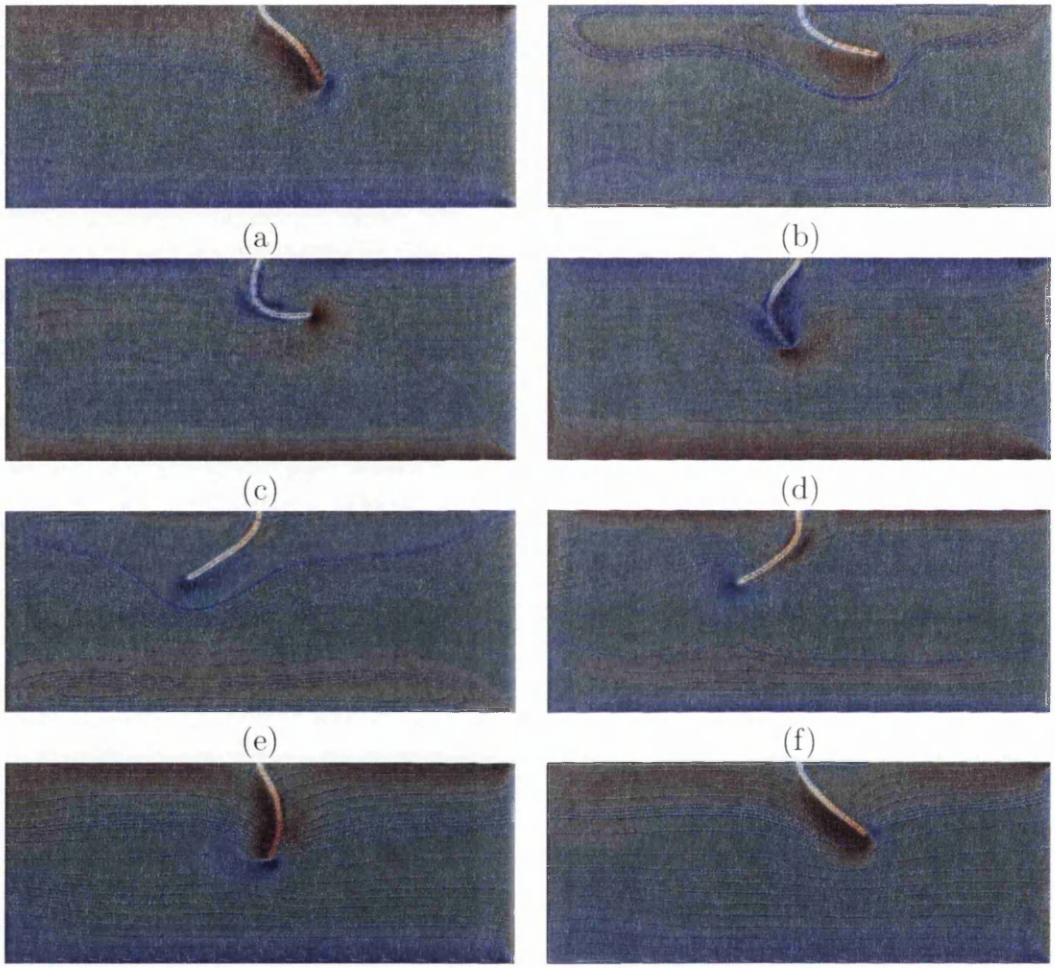


Figure 8.2: Time evolution of a flexible membrane under pulsatile flow and streamlines of the fluid. Fluid viscosity $\mu = 1$, shear modulus of $G = 2.0 \times 10^7$, 4025 solid integration points, mesh size 160×80.

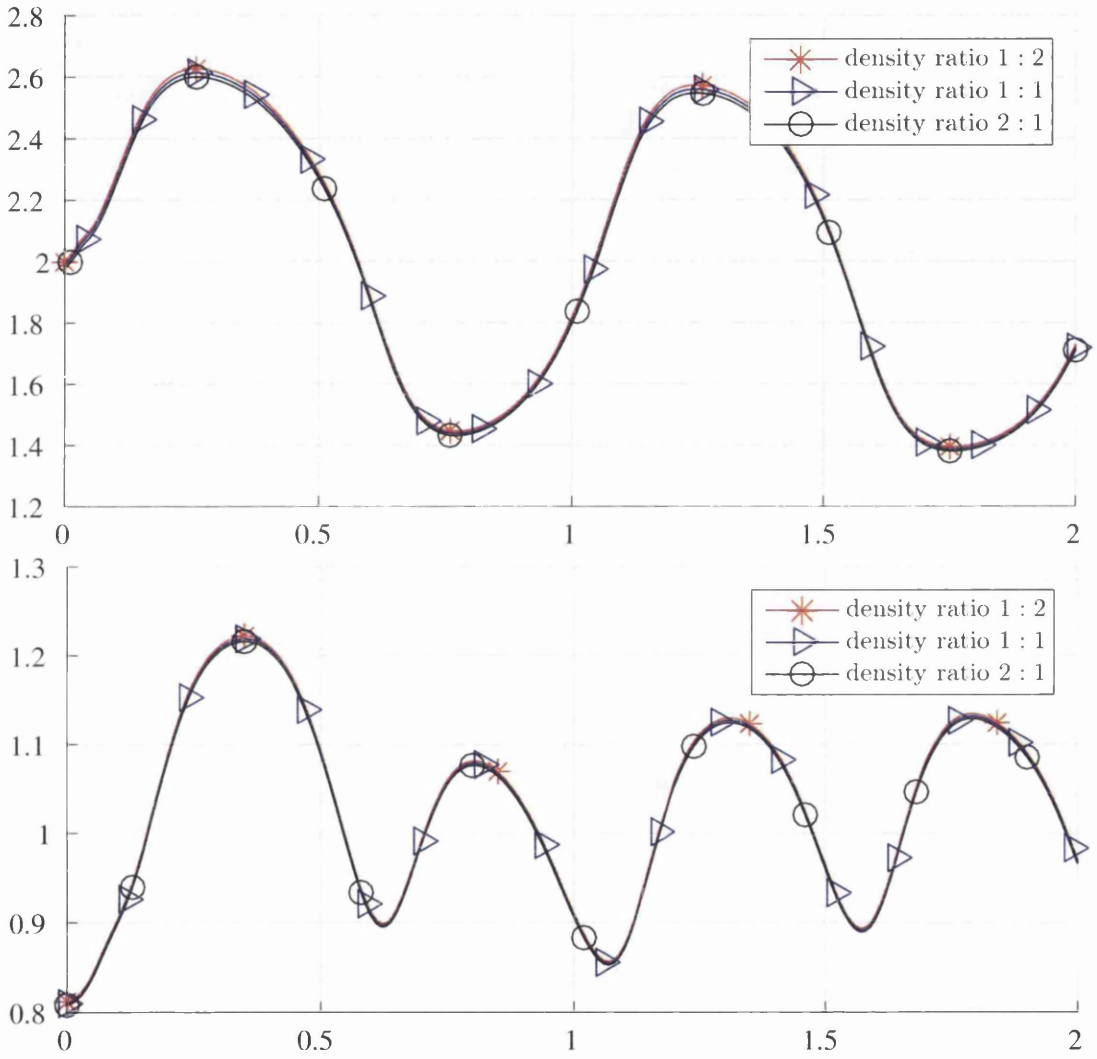


Figure 8.3: $x-$ and $y-$ components of the bottom left point of the membrane (cm) with different density ratio $1:2$, $1:1$ and $2:1$. Fluid viscosity $\mu = 1$, shear modulus of $G = 2.0 \times 10^7$, 4025 solid integration points, mesh size 160×80.

8.2 Dam break with elastic beam

A water column of width $a = 14.6$ cm and height $2a$ is placed in the left corner of a tank of size $4a \times 1.5a$. A rectangular incompressible obstacle, with width $b = 1.2$ cm, height $\frac{20}{3}b$, Young's modulus $E = 10^7$ g/cm/s^2, $\rho_s = 2.5$ g/cm^2 and Poisson ratio 0.5, is fixed at the bottom of the tank. There is no current experimental results available for this problem, but a similar problem with a free surface flow has been investigated using a monolithic FSI model [6], partitioned FSI method [147], Particle Finite Element Method (PFEM) [148] and Smoothed-Particle Hydrodynamics (SPH) model [149]. The time history of the displacement of the beam is compared

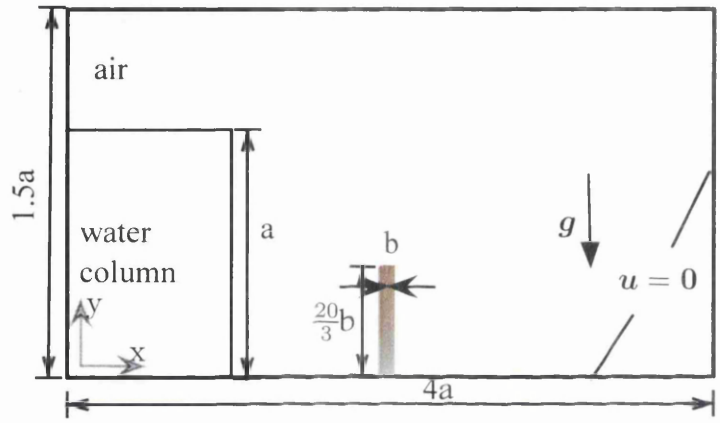


Figure 8.4: Schematic of the dam break with elastic beam

in Figure 8.5 for the various methods. When the fluid hits the obstacle, the left upper corner of the beam first deflects to the left and while the water rises, it moves to the right. It obtains its maximum deflection when the water passes the top and is fully attached to the left side of the structure. The impact of the fluid causes the beam to oscillate. This simulation is run for a mesh size 256×160. It is shown that

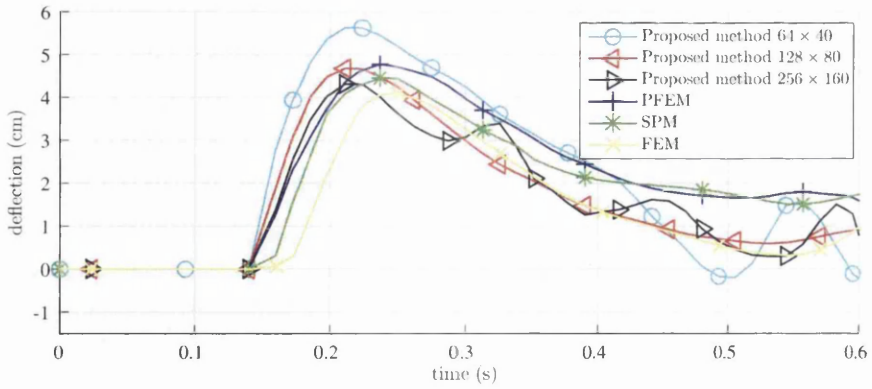
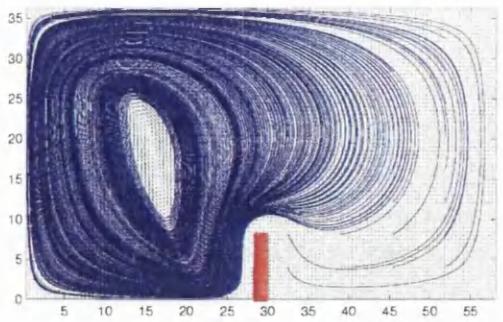
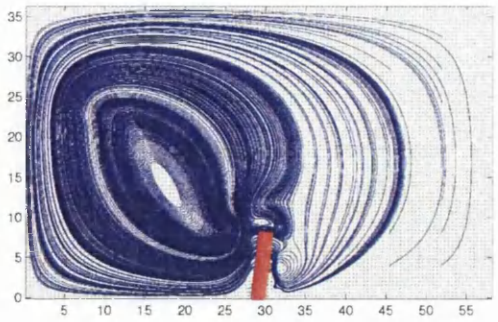


Figure 8.5: History of the $x-$displacement of the upper left corner of the elastic beam.

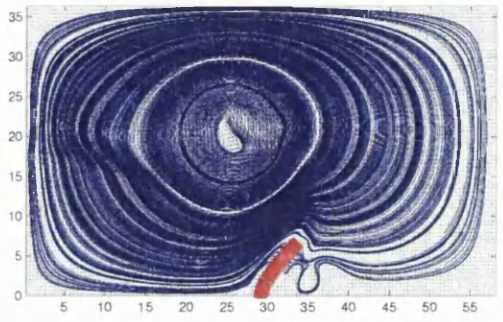
around $t = 0.2$ s, the beam reaches its maximum deflection. The snapshots of the simulation are shown in Figure 8.6 and 8.7.



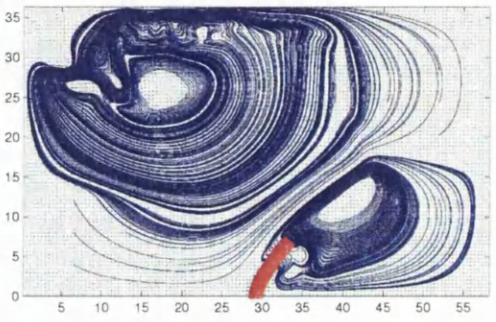
(a): $t = 0.1$ s



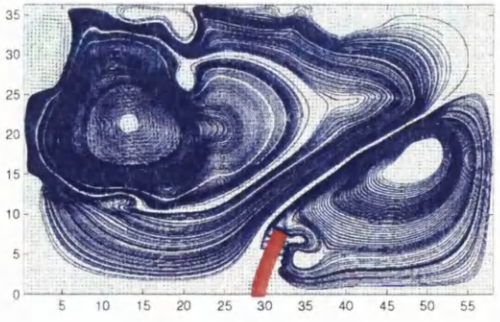
(b): $t = 0.15$ s



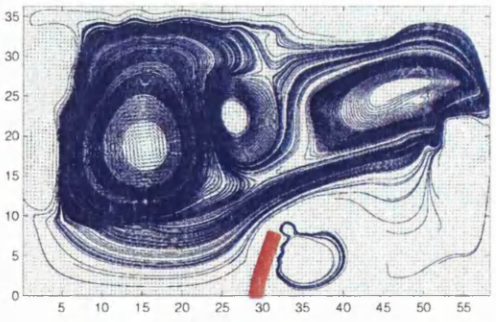
(c): $t = 0.2$ s



(d): $t = 0.25$ s

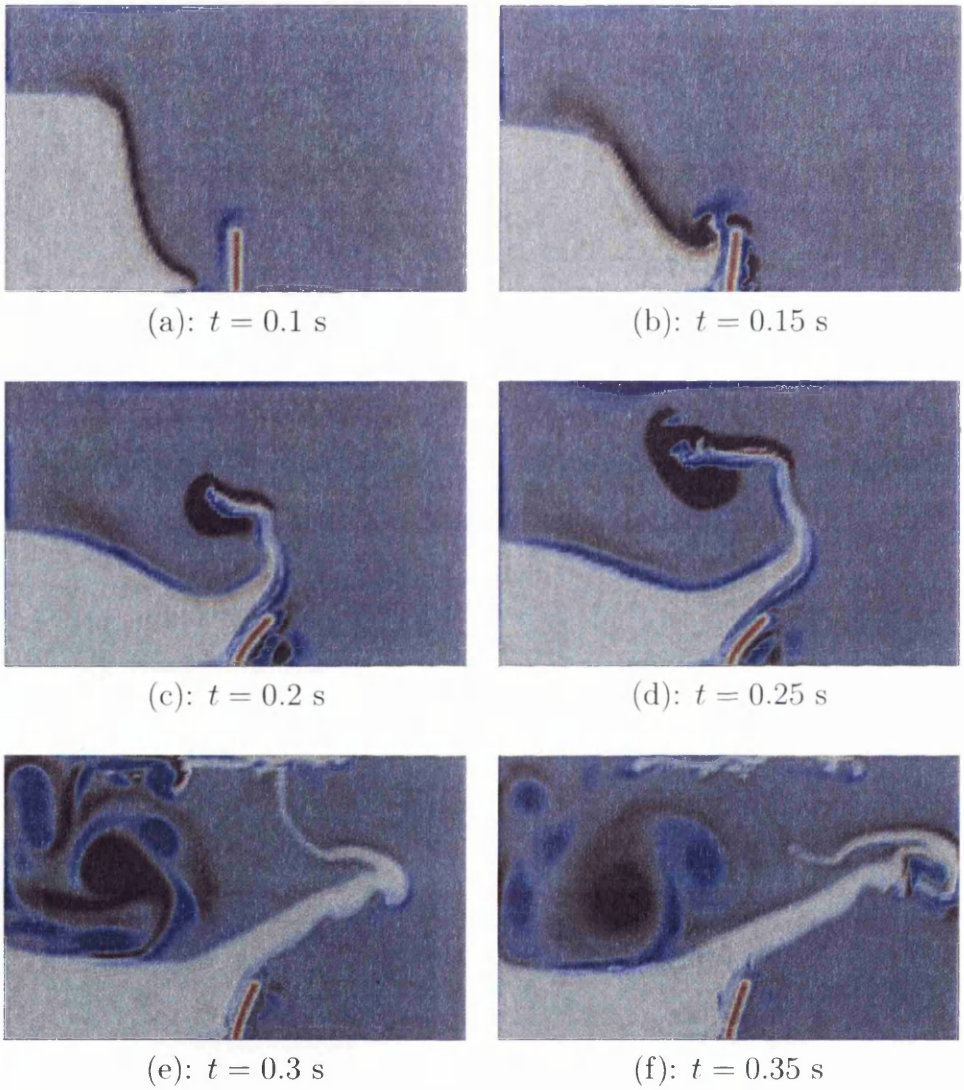


(e): $t = 0.3$ s



(f): $t = 0.35$ s

Figure 8.6: Time evolution of the interaction between a wave hitting an elastic wall and streamlines of the fluid. Water density $\rho_w = 1.0 \times 10^3$ g/cm^2, air density $\rho_a = 1.0 \times 1.0$ g/cm^2, structure density $\rho_s = 2.5 \times 10^3$ g/cm^2, water viscosity $\mu_w = 10^{-2}$ Pa s, air viscosity $\mu_a = 10^{-5}$ Pa s, Young's modulus $E = 10^7$ g/cm/s^2, mesh size 256×160, 4025 solid integration points.



(a): $t = 0.1$ s (b): $t = 0.15$ s

(c): $t = 0.2$ s (d): $t = 0.25$ s

(e): $t = 0.3$ s (f): $t = 0.35$ s

Figure 8.7: Time evolution of the interaction between a wave hitting an elastic wall and vorticity field of the fluid, vorticity contours ($-10 < \omega < 10$). Water density $\rho_w = 1.0 \times 10^3$ g/cm^2, air density $\rho_a = 1.0 \times 1.0$ g/cm^2, structure density $\rho_s = 2.5 \times 10^3$ g/cm^2, water viscosity $\mu_w = 10^{-2}$ Pa s, air viscosity $\mu_a = 10^{-5}$ Pa s, Young's modulus $E = 10^7$ g/cm/s^2, mesh size 256×160, 4025 integration points.

8.3 Elastic beam under time-dependent water pressure effects

This problem is similar to the previous one, and it was first analysed by [150] using available experimental and numerical data. It has been used as a validation FSI test, particularly for SPH simulations. In the experiment, an elastic gate, clamped at the top end and free at the bottom one, interacts with a mass of water initially confined in a free-surface tank behind the gate (see Figure 8.8). The flexible gate, 5 mm thick, is made of rubber with a density fo 1100 kg/m^3 density and an assumed constant 10 MPa Young modulus. The comparison between snapshots from the experimental

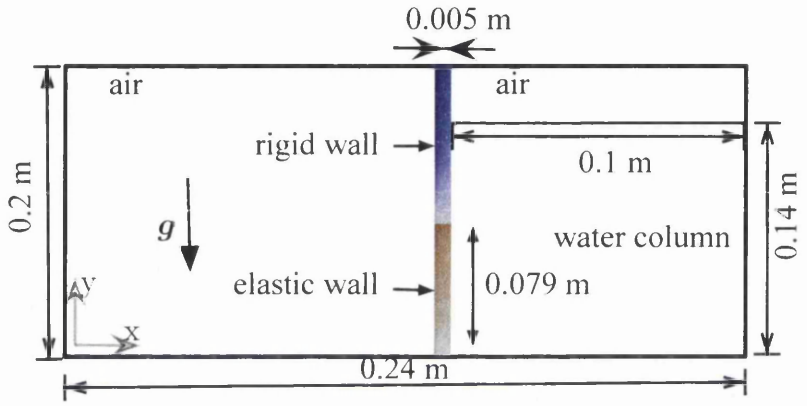


Figure 8.8: Schematic of the tank and of the elastic gate.

results and the present method results are illustrated in Figure 8.10. In Figure 8.9, a qualitative comparison is shown. It can be observed that the maximum deformation occurs near the free end of the gate. The results match well with the experiment by Antoci [150]. The gate reaches its maximum deformation at time $t = 0.15$ s. Small differences which are seen after the water hits the left wall.

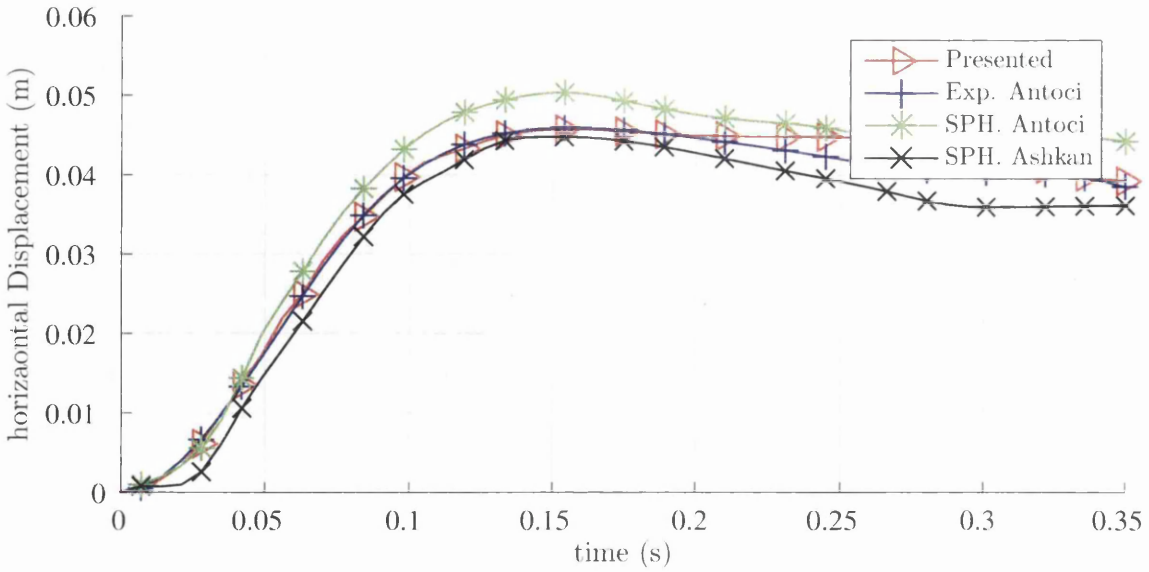


Figure 8.9: Comparison of the horizontal displacement of the free end of the gate. water density $\rho_w = 1.0 \times 10^3$ kg/m^3, air density $\rho_a = 1 \times 1.0$ kg/m^3, structure density $\rho_s = 1.1 \times 10^3$kg/m^3, water viscosity $\mu_w = 10^{-3}$Pa s, air viscosity $\mu_a = 10^{-5}$Pa s, Young's modulus $E = 10$ MPa, mesh side 288 × 160, 4025 solid particles.

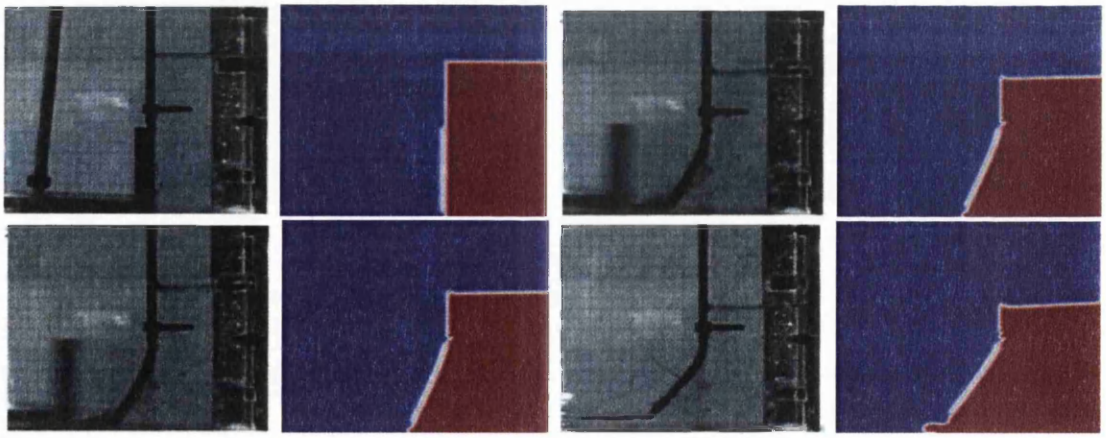


Figure 8.10: Comparison of the numerical and experimental results, every 0.08 sec. water density $\rho_w = 1.0 \times 10^3$ kg/m, air density $\rho_a = 1 \times 1.0$ kg/m^3, structure density $\rho_s = 1.1 \times 10^3$ kg/m^3, water viscosity $\mu_w = 10^{-3}$ Pa s, air viscosity $\mu_a = 10^{-5}$ Pa s, Young's modulus $E = 10$ MPa, mesh side 288 × 160, 4025 solid integration points.

Chapter 9

Rigid body

The interaction of a rigid body with water waves is a very interesting hydrodynamic problem. A series of numerical solutions for the coupling of a rigid body with a one-phase fluid and a multiphase fluid is presented to show the capabilities of the computational methodology proposed in this thesis.

Two numerical examples are illustrated for a fixed rigid body, flow over a cylinder and entry/exit of a cylinder into water. The rigid velocity is given. Three examples are presented for rigid motion without constraints. The first problem is the terminal velocity of a free falling rigid cylinder. This case is similar to flow over a cylinder, where the terminal velocity depends on the correct calculation of the drag forces. The second problem is the fluttering of a rigid body in the fluid. Unlike the previous laminar flow case, rotations and translations can be observed. The last case shown is the release of a rigid ball into water.

9.1 Flow over cylinder

The problem of the flow over cylinder has been numerically studied by [18]. Here the same parameters as those in [18] are employed. The underlying geometry is a pipe shown in Figure 9.1. The kinematic viscosity is taken as $\nu = 0.001$ with a density $\rho = 1$. The parabolic inflow profile is prescribed as

$$\boldsymbol{u} = \left(\frac{4Uy(0.41 - y)}{0.41^2}, 0 \right)^T \tag{9.1}$$

The maximum velocity is chosen to be $U = 1.5$, so that the parabolic profile results in a mean velocity $U_{mean} = 1$. The Reynolds number can be evaluated as

$$Re = \frac{U_{mean} L}{\nu} = 100$$

The fluid is discretised using the Marker-and-Cell method in a sequence of meshes $110 \cdot 2^i \times 41 \cdot 2^i$ cells for i-th run, $i = 0, 1, 2, 3$. The finest computational mesh is set to be 880×328, where $\Delta x = 2.5 \times 10^{-3}$, $\Delta y = 1.3 \times 10^{-3}$ and $\Delta t = 3.125 \times 10^{-4}$

The drag and lift forces are easily calculated using the immersed rigid body force

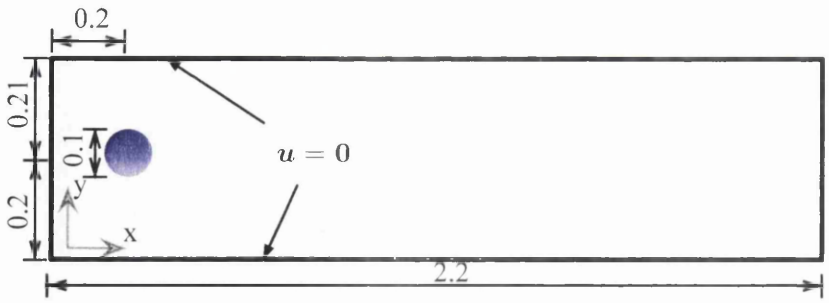


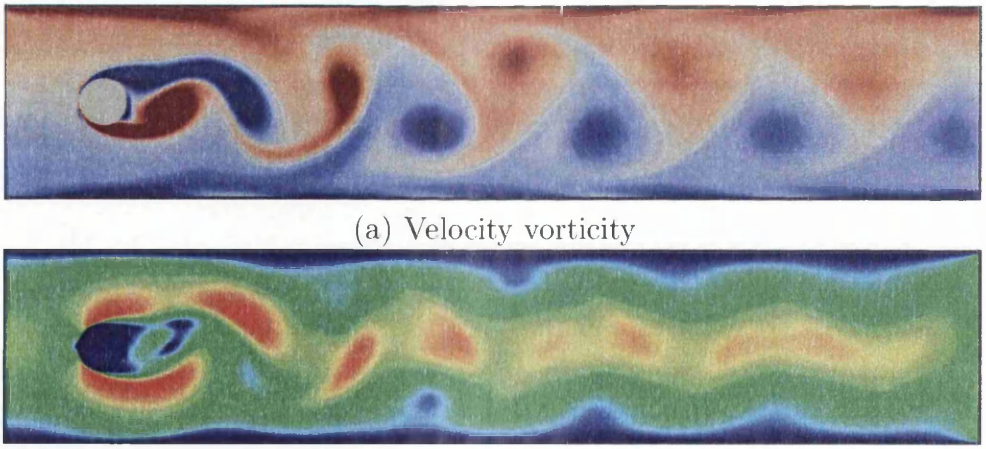
Figure 9.1: Geometry and boundary conditions flow around cylinder.

$$F_D = -\int_\Omega \boldsymbol{f} \cdot \boldsymbol{e}_x \, dv; \quad F_L = -\int_\Omega \boldsymbol{f} \cdot \boldsymbol{e}_y \, dv, \tag{9.2}$$

where $\boldsymbol{f}$ is the immersed rigid body force and $\boldsymbol{e}_x = [1,0]^T$, $\boldsymbol{e}_y = [0,1]^T$. The dimensionless drag and lift coefficients are calculated as

$$C_D = \frac{2}{U_{mean}^2 L} F_D; \quad C_L = \frac{2}{U_{mean}^2 L} F_L. \tag{9.3}$$

For Reynolds number $Re = 100$, the flow turns into a time-periodic behaviour with vortex shedding behind the cylinder. The flow vector field is visualised in Figure 9.2 with vorticity in $Z-$ direction given by $\left(\frac{\partial u_y}{\partial y} - \frac{\partial u_x}{\partial y}\right)$ and a velocity magnitude of $\sqrt{u_x^2 + u_y^2}$, showing the periodic Karman vortex shedding. Table 9.1 and Figure



(a) Velocity vorticity

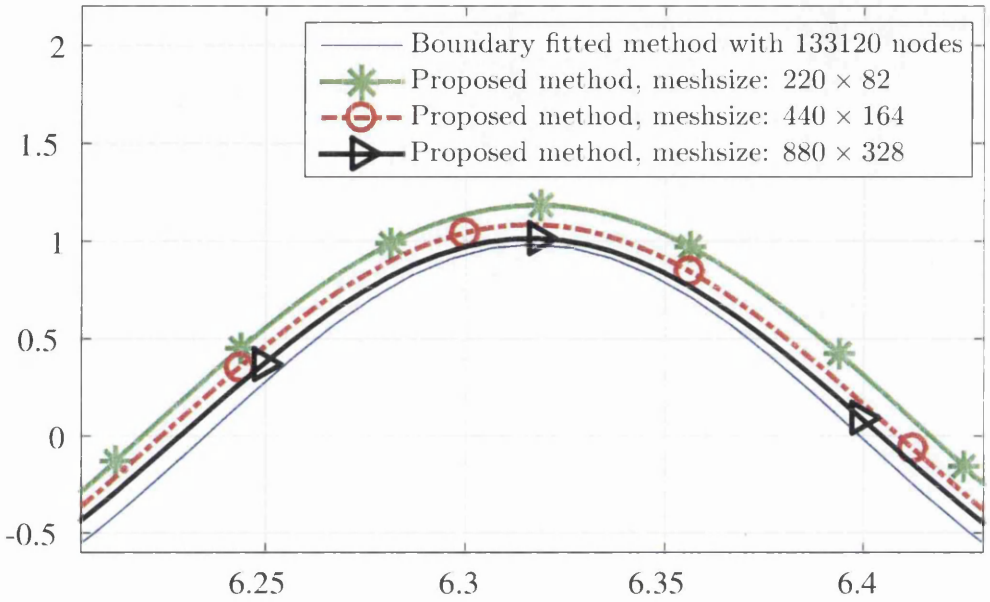
(b) Velocity magnitude

Figure 9.2: Velocity profile for $Re = 100$, mesh size 800×312. The cylinder is described by a Level Set function.

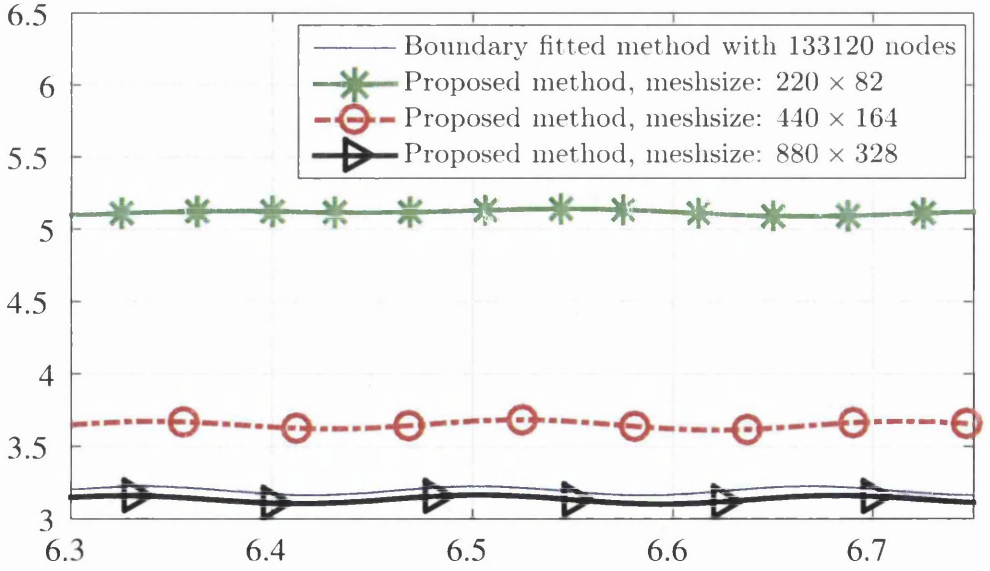
9.3 show the information about the drag and lift coefficients. The computation on the coarsest mesh $(110, 41)$ does not show a periodic Karman vortex because of the numerical diffusion, which will be listed for the comparison.

run	Δt	Meshsize	Geometry description	C_D	C_L
1	1.3×10^{-3}	(220,82)	Level Set	5.1168	1.1842
2	6.25×10^{-4}	(440,164)	Level Set	3.6496	1.0864
3	3.125×10^{-4}	(880,328)	Level Set	3.1337	1.0138
Ref.		133120 nodes	Boundary fitted mesh	3.1884	0.9830

Table 9.1: Comparison of the Lift coefficient C_L and drag coefficient C_D with the Boundary Fitted Method [18], where C_D is the average drag coefficient and C_L is the maximum lift coefficient.



(a) Lift coefficient C_L



(b) Drag coefficient C_D

Figure 9.3: Lift coefficient C_L and drag coefficient C_D with respect to time t, compared with the Boundary Fitted Method [18] with 133120 nodes. The cylinder is described by a Level Set function.

9.2 Entry/exit of a cylinder in water

The problem of a cylinder impacting on water has been numerically studied in [151–153]. In this example, the same parameters as those in [152, 153] are used: a circular cylinder of radius $R = 1$ is placed in air and the distance of its centre to the air-water interface is $H = 1.25$. The gravity's acceleration is set to be $\boldsymbol{g} = (0, -1)^T$ and the cylinder is given a constant downward velocity $u_y = -1$ at the $T = u_y t/H = 0$ with t the time in the calculation. The computational domain is set to be $\Omega = [40R \times 24R]$. For the exit from water problem, the same parameters are used as those for the entry problem, except the cylinder starts its upward motion from a height of $H = -1.25$ below the stationary water surface with a constant upward velocity $u_y = 0.39$. The mesh grid is set to 400×240 for the whole domain

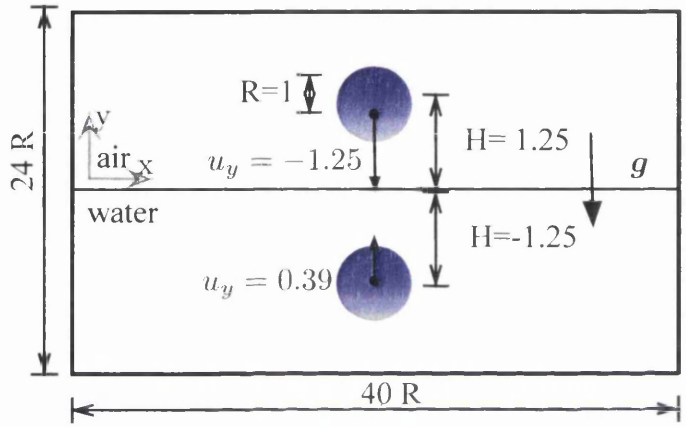
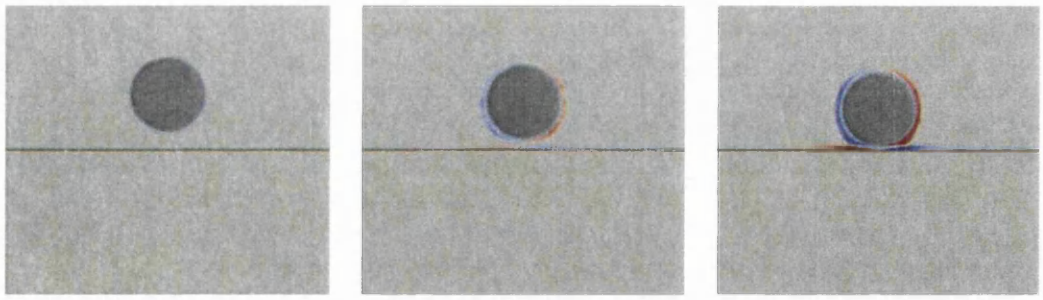


Figure 9.4: Geometry description for the cylinder water entry/exit problems

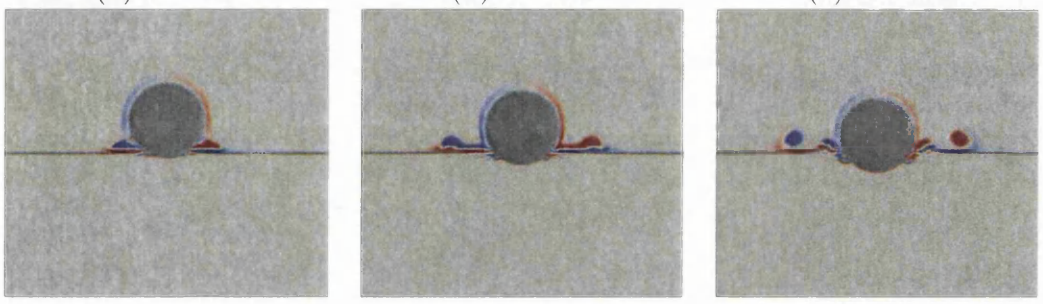
and the rigid cylinder is described by a Level Set function where the centre of the cylinder moves with the prescribed speed. A series of snapshots are shown in Figure 9.5. For the entry problem impacts the air-water interface, two jets are generated along the left and right side of the cylinder. Several vortices shed from the shear layers along the surface of the cylinder interact with the air-water interface. The results are very similar to [152, 153].

9.3 Terminal velocity of a free falling rigid cylinder

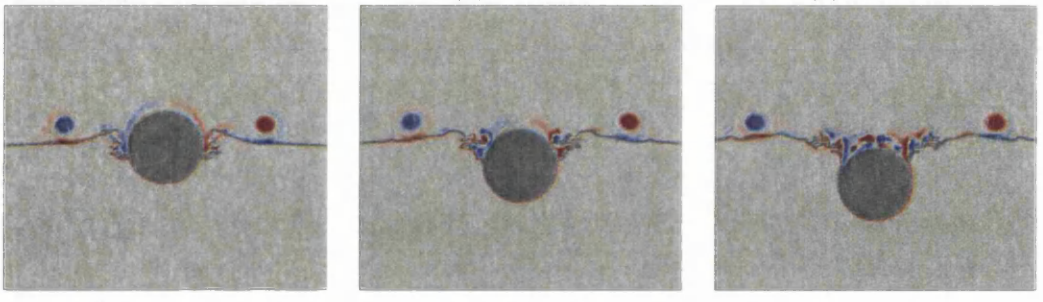
A series of numerical examples of a rigid cylinder or disks with diameter $2a$ falling in a channel with dimensions $8L \times 2L$ filled with an incompressible Newtonian viscous fluid under the action of a horizontal body force are presented. See Figure 9.7 for a description of the geometry. The physical parameters for this set of examples are as follows: gravitational acceleration $g = 9.81$ m/s^2; fluid dynamic viscosity $\mu = 1$ dyne/cm^2(0.1 Pa s); fluid density $\rho_f = 1$ g/cm^3; $L = 2$ cm; $a = 1/4$ cm. These



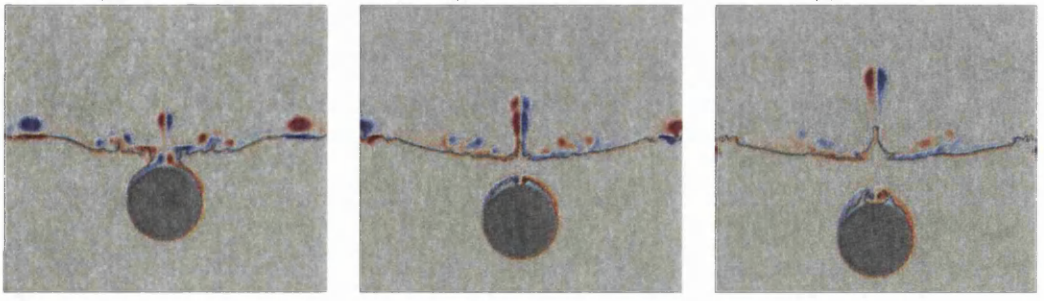
(a) $t = 0.0$ (b) $t = 0.2$ (c) $t = 0.4$



(d) $t = 0.6$ (e) $t = 0.8$ (f) $t = 1.0$

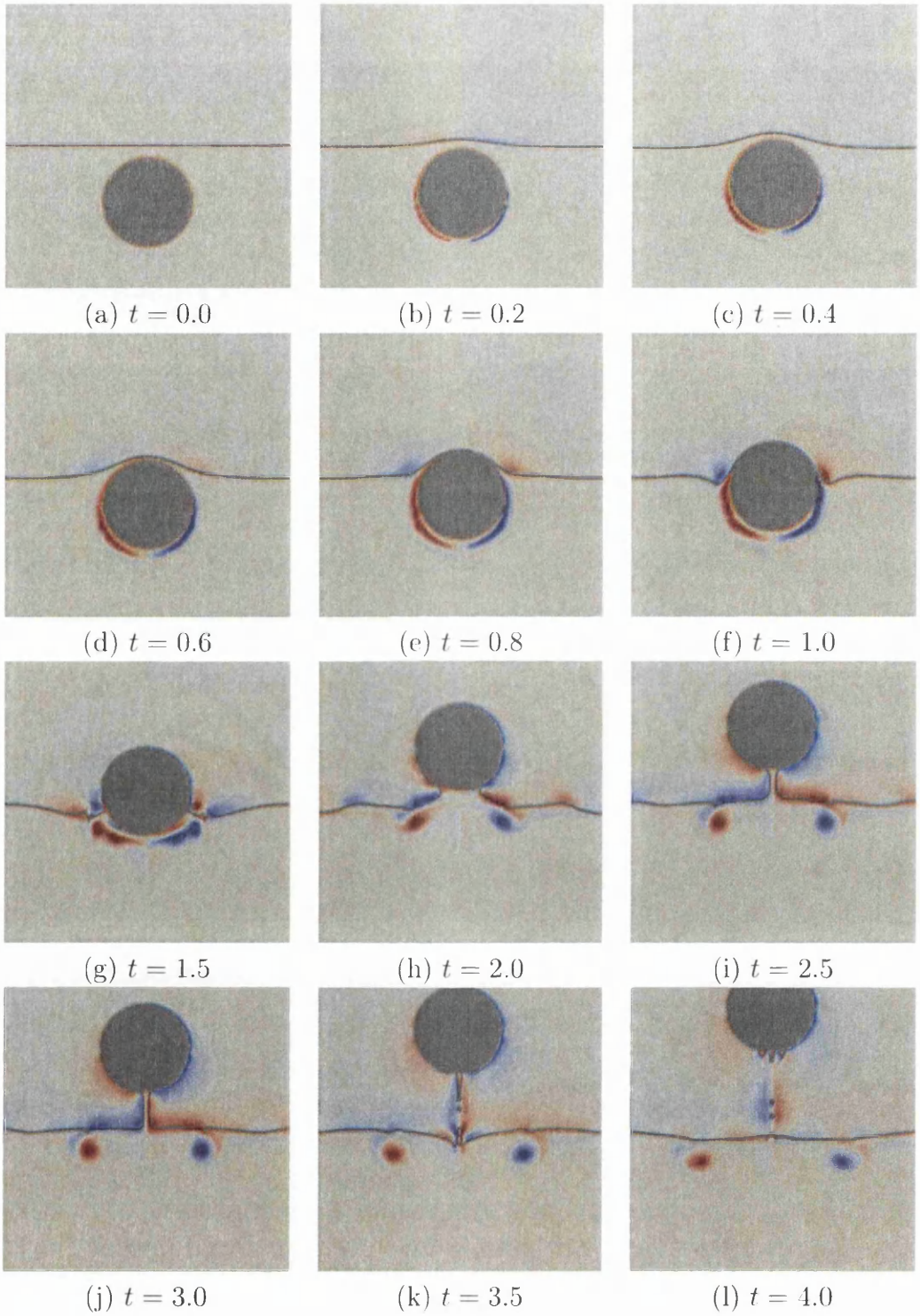


(g) $t = 1.5$ (h) $t = 2.0$ (i) $t = 2.5$



(j) $t = 3.0$ (k) $t = 3.5$ (l) $t = 4.0$

Figure 9.5: Water impact problem: the air-water interface position (solid black line) and vorticity contours ($-20 < \omega < 20$), $g = -1$, constant velocity $\boldsymbol{u}_y = -1$, meshgrid 400×240, the rigid cylinder is described by the Level Set $r = 1$.



(a) $t = 0.0$ (b) $t = 0.2$ (c) $t = 0.4$

(d) $t = 0.6$ (e) $t = 0.8$ (f) $t = 1.0$

(g) $t = 1.5$ (h) $t = 2.0$ (i) $t = 2.5$

(j) $t = 3.0$ (k) $t = 3.5$ (l) $t = 4.0$

Figure 9.6: Water exit problem: the air-water interface position (solid black line) and vorticity contours ($-10 < \omega < 10$), $g = -1$, constant velocity $\boldsymbol{u}_y = 0.39$, meshgrid 400×240, the rigid cylinder is described by the Level Set with a radius $r = 1$.

run	Δt	(nx,ny)	Rigid indicator
1	1.8×10^{-3}	(18,24)	Level Set
2	4.4×10^{-4}	(36,48)	Level Set
3	2.2×10^{-4}	(72,96)	Level Set
4	1.1×10^{-4}	(192,144)	Level Set
5	2.76×10^{-5}	(384,288)	Level Set

Table 9.2: A list of simulation cases for the simulation of free falling rigid cylinder.

values lead to a Reynolds number $Re = 14.6$. The rigid solid is initially placed at the geometric centre of the channel and is modelled using a Level Sets. The cylinder has a density of $\rho_s = 1200$ g/cm^3.

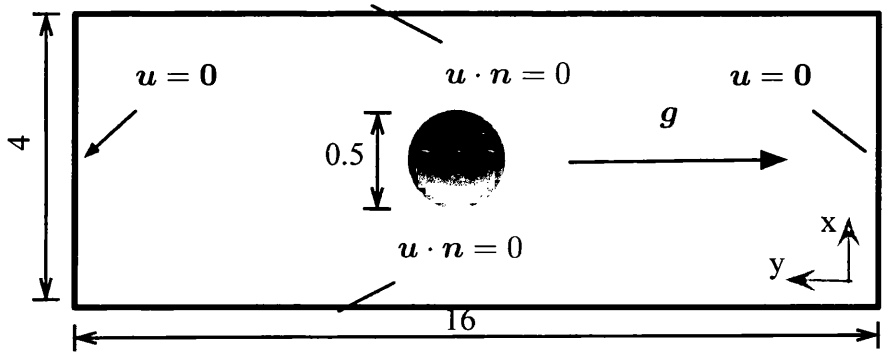


Figure 9.7: Geometry and boundary conditions for a rigid falling cylinder in a channel (dimensions in cm).

For a rigid cylinder, the solution for the terminal velocity U in the direction of the gravitational force is given by [154]:

$$U = \frac{(\rho_s - \rho_f)\, g a^2}{4\mu} \left[\ln\left(\frac{L}{a}\right) - 0.9157 + 1.7244 \left(\frac{a}{L}\right)^2 - 1.7302 \left(\frac{a}{L}\right)^4 \right] \tag{9.4}$$

In contrast with other researchers [5, 66], the rigid body is modelled using a Level Set function. The effect of gravity is applied to both fluid and the rigid body, properly capturing the hydrostatic pressure profile. The fluid is discretised using the Marker-and-Cell method in a sequence of meshes $12 \cdot 2^i \times 9 \cdot 2^i$ cells for i-th run, $i = 1, 2, 3, 4, 5$.

The evolution of the velocity of the cylinder can be observed in Figure 9.8. Oscillation can be observed for the coarsen meshes and due to the numerical errors of constructing the indicator function. Considered another case, the cylinder forced to move with the constant speed $\boldsymbol{u} = (0, -0.0365)$ in order to study the conservation of the area over time. On average the mass of the coarse grid is overestimated. As a result, the terminal velocity is overestimated as well.

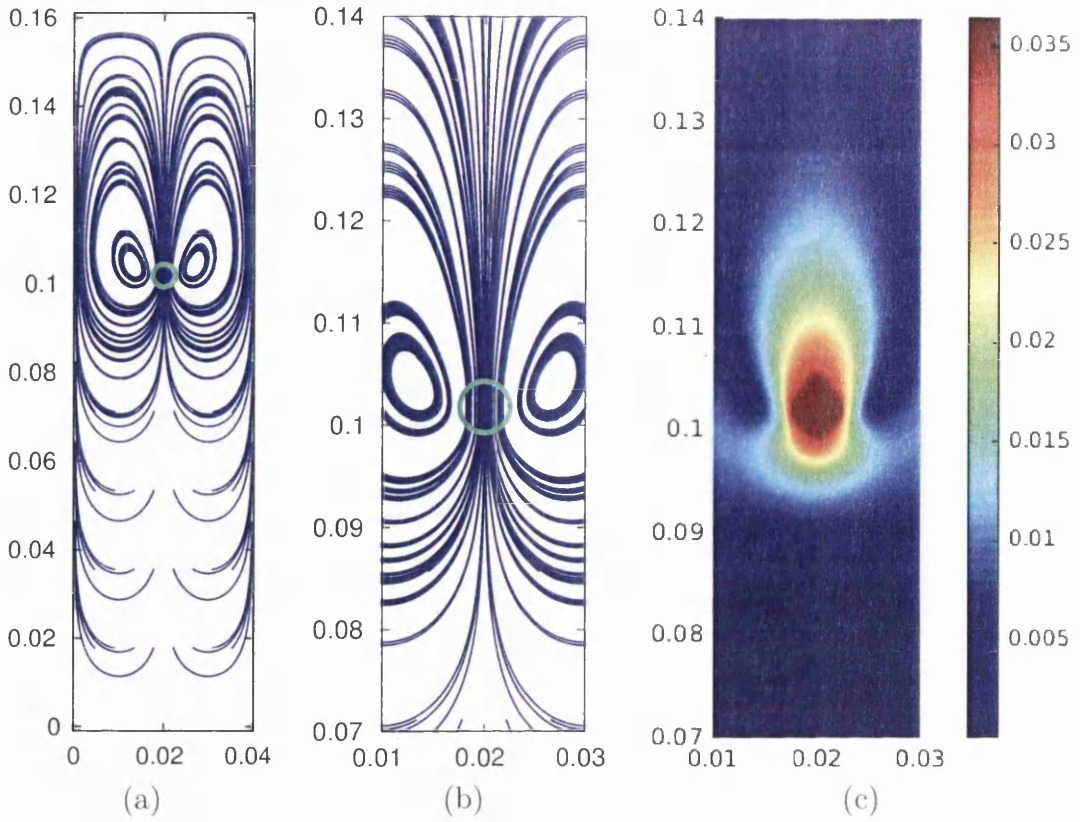


Figure 9.8: Free falling rigid cylinder. From left to right: (a) streamlines of the velocity field. (b) close view of the streamlines. (c) velocity contour. Mesh size 384×288, the cylinder is described by the Level Set.

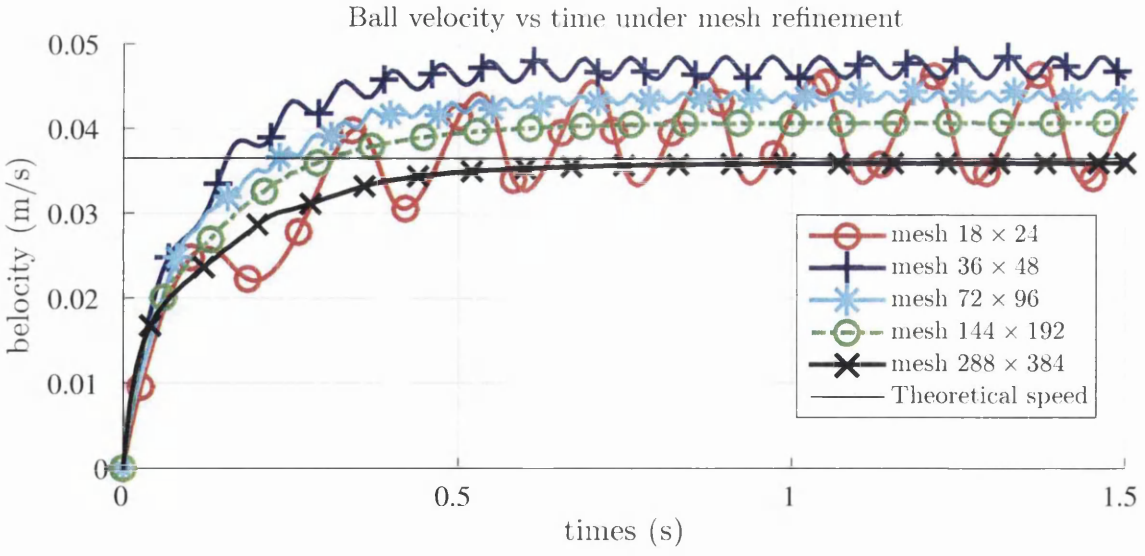


Figure 9.9: Velocity history of a free falling rigid cylinder with a diameter ratio $\frac{a}{L} = \frac{1}{8}$ ($d = 2a$, $r = \frac{1}{4}$, $L = 2$).

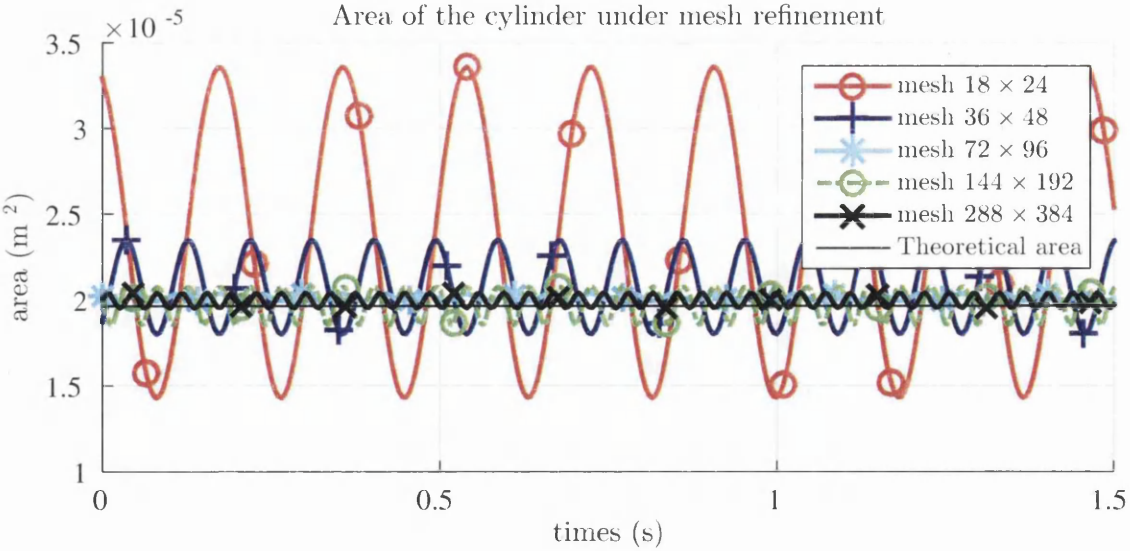


Figure 9.10: Area of the cylinder if moving the cylinder passively wtih its theoretical terminal speed, the theoretical area $\pi r^2 = 1.9635 \times 10^{-5}$

9.4 Rigid body fluttering

In this example, we consider a rectangular domain 15×40, discretised with a series of Cartesian meshes, the finest of which is composed of 240×640 cells, filled with a Newtonian viscous fluid of viscosity $\mu = 10^{-5}$ and density $\rho = 10^3$. A rigid rectangle of size 5×0.5 is rotated clockwise an angle of $\pi/3$ and translated such that its geometrical centre is at position $(3.4665, 35.96)$ with respect to the bottom left corner of the fluid domain (see Figure 9.11). The rigid solid has a density 1.5 times that of the fluid and is discretised using 5760 integration points. The total run-time for the case with the finest mesh is 1 hour using a 2.4 GHz Intel Core 2 Duo CPU. In Figure 9.12a and 9.12b we can observe convergence of the evolution over time of the position and velocity of the bottom left corner of the rectangle for a series of discretisation. In Figure 9.13, in a series of snapshots of the solution, it can be observed how the rigid solid sinks and flutters as it creates vortices in its wake.

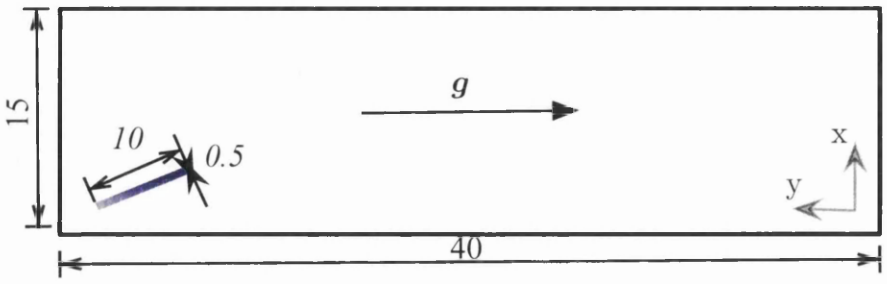


Figure 9.11: Geometry of the rigid body fluttering problem.

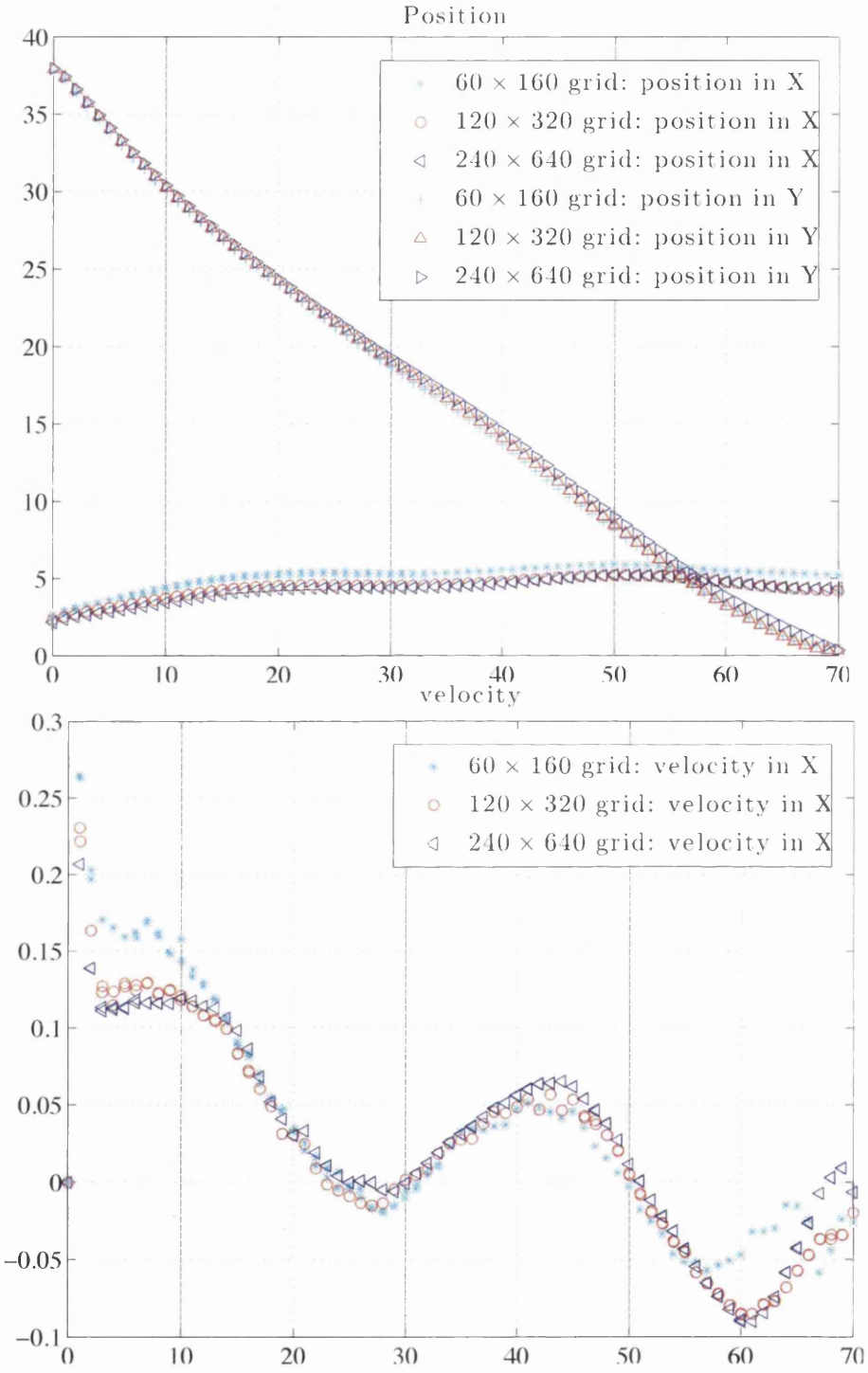
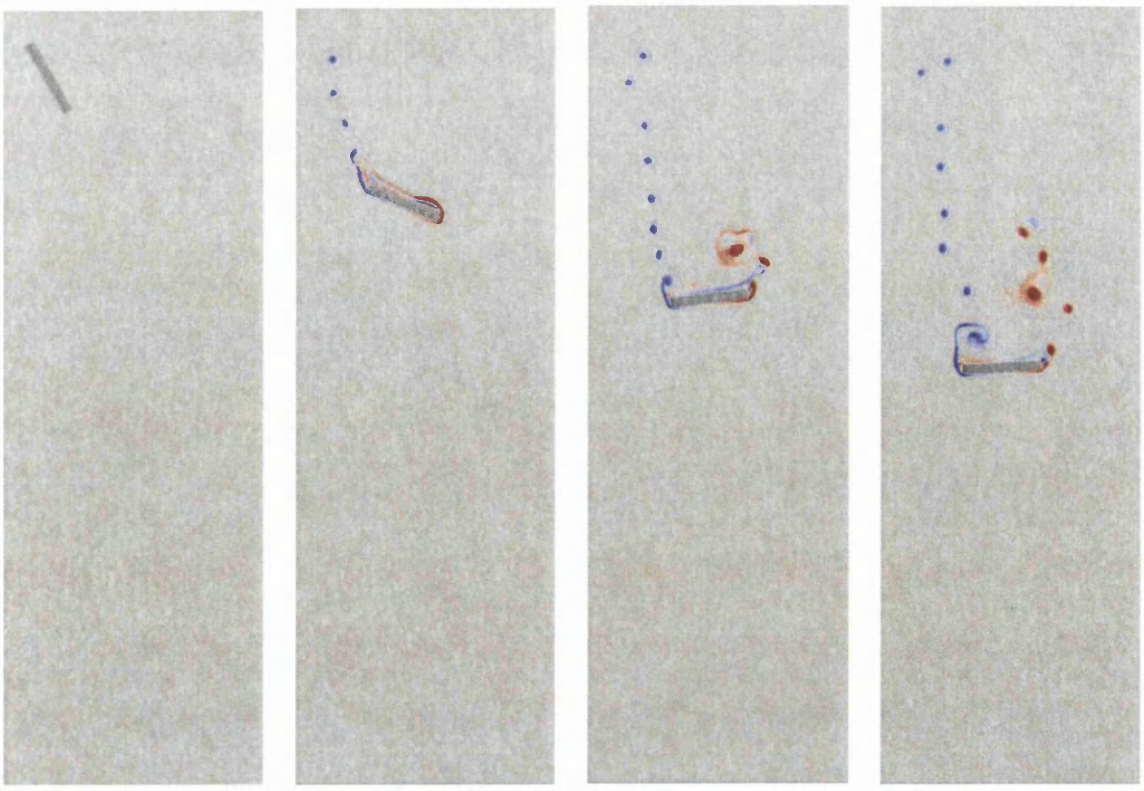
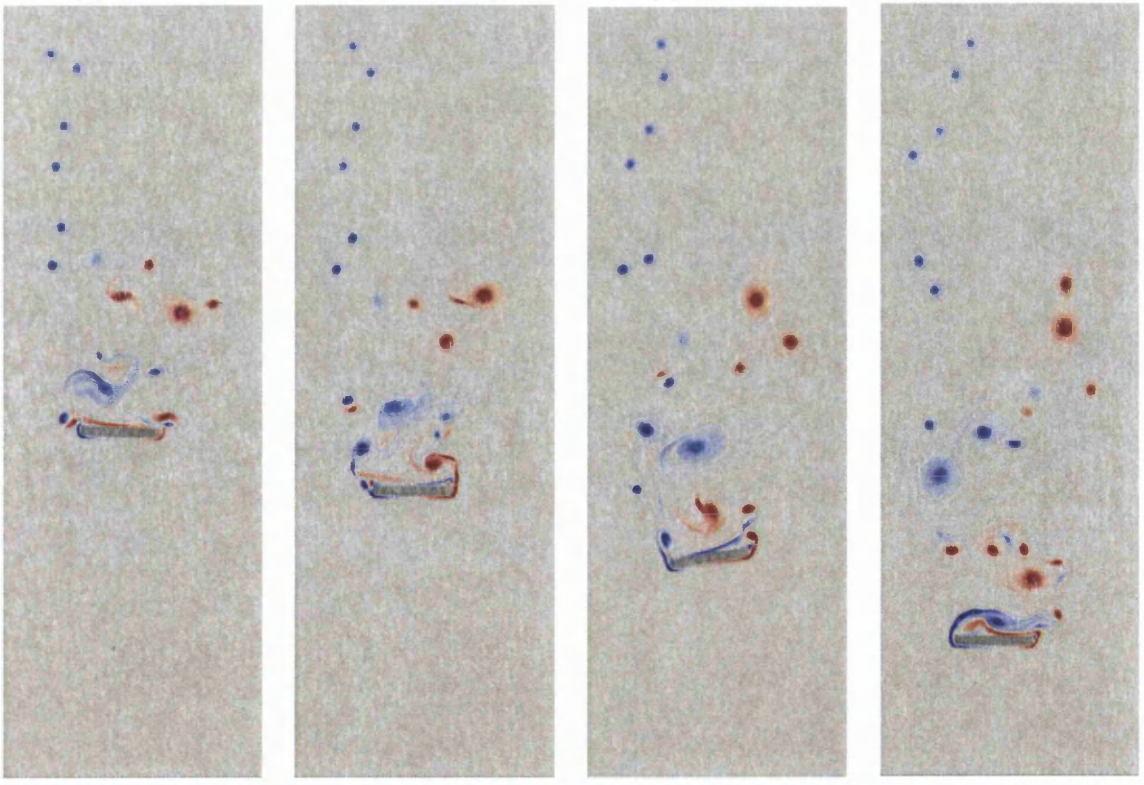


Figure 9.12: (a) Evolution with respect to time of the x and y position of the bottom left corner of the rigid rectangle for a series of mesh discretisation; (b) Corresponding x-velocity of the same point with respect to time for a series of meshes.



(a) $t = 0$; (b) $t = 10$; (c) $t = 20$; (d) $t = 30$

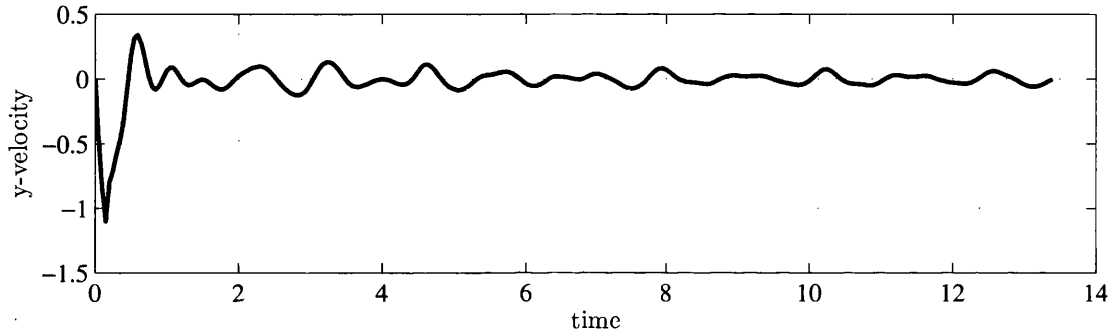


(e) $t = 40$; (f) $t = 50$; (g) $t = 60$; (h) $t = 70$

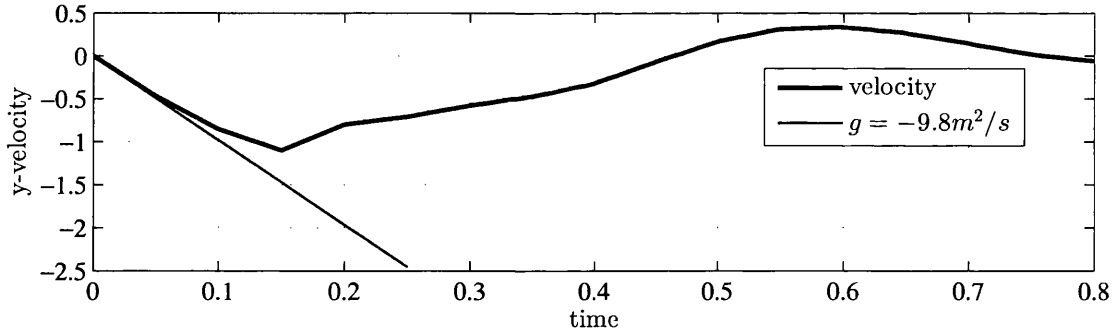
Figure 9.13: Snapshots of the solution for the problem of a rigid solid sinking and tumbling in a viscous fluid and vorticity contours ($-10 < \omega < 10$), mesh size 240×640, 5760 particles, fluid viscosity $\mu = 10^{-5}$, fluid density 10^3, rigid body density 1.5×10^3

9.5 Impact of a free falling cylinder on water

This problem is similar to the entry/exit problem of a cylinder in water, but in this case the cylinder is driven by its own gravitational force. A rigid cylinder with radius of 0.1 m is dropped into a rectangular container of dimensions of 2×1.2 m half filled with water. The left and right walls are set to be slip boundary, while the top and bottom walls are set be non-slip boundary. The gravitational acceleration $g = 9.8$ m/s^2. The rigid is initially placed at the height of 0.2 m above the stationary water surface, and released at time $t = 0$. The total simulation time is 13 s. The cylinder first accelerates due to gravity and is slowed down by the drag caused by air first and then with water. The maximum downward velocity reached is -1.0999 m/s at time 0.15 s. Upon impact, the cylinder bounces back up with a maximum upward velocity of 0.3370 m/s, as it can be observed in Figure 9.14.

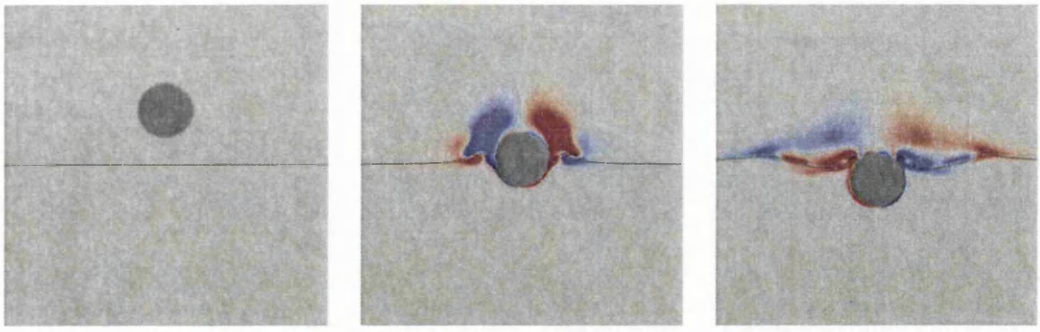


(a): whole view

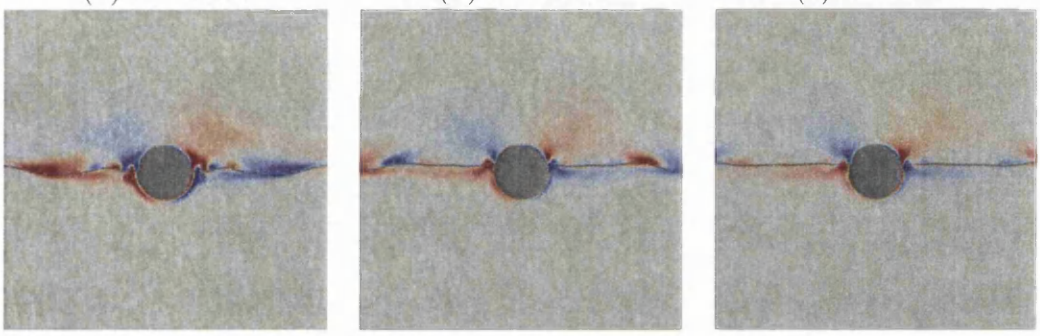


(b): close up view

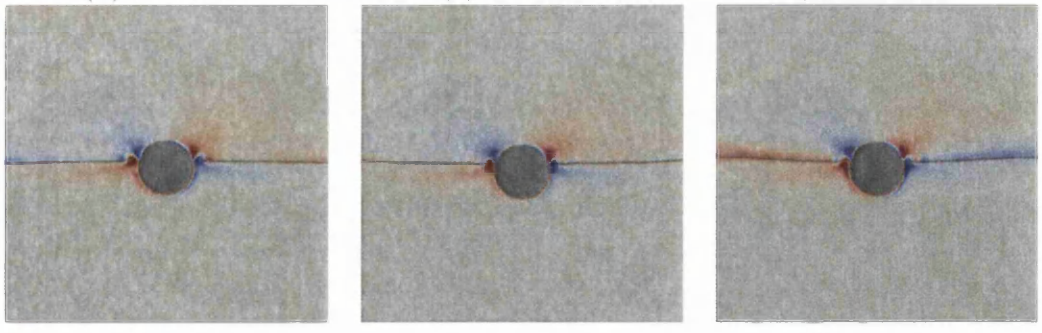
Figure 9.14: Evolution of the vertical component of the velocity with time, of a free-falling rigid cylinder of radius 0.1 m into a half filled tank of dimensions of 2×1.2 m.



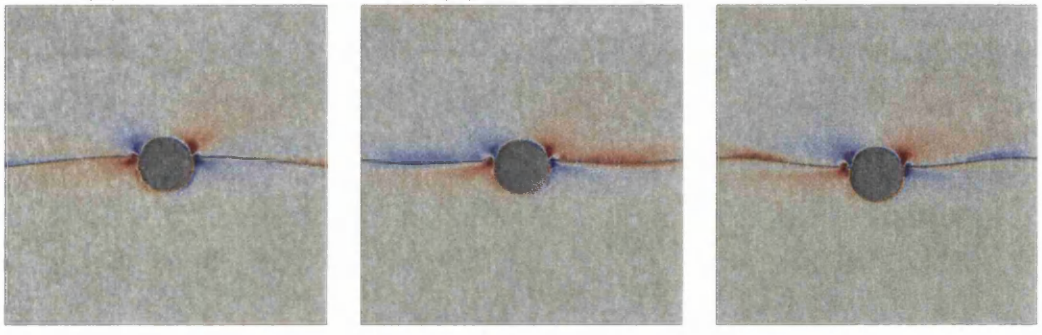
(a) $t = 0.0$ s (b) $t = 0.2$ s (c) $t = 0.4$ s



(d) $t = 0.6$ s (e) $t = 0.8$ s (f) $t = 1.0$ s



(g) $t = 1.5$ s (h) $t = 2.0$ s (i) $t = 2.5$ s



(j) $t = 3.0$ s (k) $t = 3.5$ s (l) $t = 4.0$ s

Figure 9.15: Impact of a free-falling cylinder in water. The air-water interface position (solid black line) and vorticity contours ($-10 < \omega < 10$)

Part V

Conclusions

Chapter 10

Conclusions and future work

10.1 Conclusions

This thesis has presented a computational framework for the efficient computation of incompressible multiphase flows in conjunction with rigid body and flexible body interaction problems. This novel framework provides great flexibility and enables the modelling of multiple deformable and rigid bodies interacting with the fluid into a single-phase formulation.

10.1.1 On the mathematical modelling

From a continuum modelling point of view, the proposed 'one-phase' framework has been shown to be equivalent to boundary fitted methods. Other immersed methodologies, such as the 'one-fluid' multiphase flow or the classical immersed boundary method, can be viewed as a special case of the proposed framework. Specifically, two main contributions on the mathematical modelling side have been made.

The first contribution is the consideration of the rigid body as a continuum, namely 'a rigid continuum', where the differential and variational forms of the governing equations have been presented. By making use of any of the four options (section 2.1.5) used to describe the rigid kinematics constraints, the system of governing equations is closed. The kinematic constraints are enforced via Lagrange multipliers that mimic the behaviour of a stress tensor, in a similar manner to the treatment of the incompressible kinematics constraint. The formulation for the description of the rigid continuum dynamics is presented without the explicit computation the Lagrange multipliers. One of the disadvantages of the rigid continuum approach is that the mass and the moment of inertia tensor are known in engineering purposed calculations, rather than the body's shape and its mass distribution.

The second contribution is the development of a multiple phase interaction computational framework by using a 'one-phase' fully Eulerian description with the help of indicator functions in order to describe the multiple phases, rather than by using

the more classical Dirac delta distribution approach [4]. The different regions of the continuum are differentiated by an indicator function, where the fluid, the deformable solid and the rigid body phases differ only in the definition of the Cauchy stress tensor.

10.1.2 On the numerical methodology

From the computational point of view, the framework offers a great flexibility with four key contributions.

First, it hybrids an Eulerian Level Set method with a Lagrangian material particle based method, and presents a computational algorithm for the representation, evolution and compatibility of the multiple phases.

Second, the inertia term is treated implicitly, which allows for the consideration of large density ratio problems by solving an non-constant diffusion Poisson equation, often encountered in hydrodynamics problems.

Third, the immersed deformable structure force field is derived from the spatial integration of the structure's deviatoric Cauchy stress tensor excluding volumetric effects.

Finally, an iterative procedure has been proposed for the modelling of a rigid body phase, where the immersed rigid force is updated dynamically within the residual of the set of discrete equations. A linear least square projection procedure is proposed for the rigid velocity projection. This method is capable of dealing with arbitrary rigid motions without resorting to the use of the classical Lagrangian mechanics description of the rigid body.

10.1.3 On the computer implementation

From the computer implementation standpoint, the critical computing routines have been implemented in FORTRAN and the overall structure of the code is organised in MatLab. The anisotropic Poisson solver is speeded up by using the geometric multigrid solver included in the Hypre library [106].

10.1.4 On the engineering applications

In terms of applicability and reliability, a wide range of benchmark and applications have been presented in the examples chapters, including bubble dynamics, hydrodynamics, three-phase flows, haemodynamics, fluid-structure interaction with free surfaces, aerodynamics and complex fluid-rigid coupling with free surfaces. These problems are included in order to prove the robustness and flexibility of the proposed computational methodology.

10.2 Future work

Future research lines have been opened-up after the developments presented in this thesis. A few directions are pointed out:

- **Extension into three-dimensional simulations.** For realistic engineering applications, the framework should be extended to three dimensional scenarios. There is nothing in the methodology that precludes a straightforward extension to three-dimensional simulations, the only consideration being an efficient implementation and the corresponding computational cost. The computational speed of the extended algorithm will certainly benefit from the use of a fast Marker-And-Cell discretisation.

- **Fully Eulerian based formulation.** At the moment, the background mesh computations are Eulerian. The multiphase flow interfaces are captured by a Level Set method and the rigid body can be modelled either by a Level Set or a Lagrangian based method. The flexible body described in a Lagrangian fashion can be also modelled in an Eulerian manner. The inverse of the deformation gradient can be transported in an Eulerian way as follows

$$\frac{\partial(\nabla \boldsymbol{X})}{\partial t} + \boldsymbol{u} \cdot \nabla(\nabla \boldsymbol{X}) = \boldsymbol{0} \tag{10.1}$$

- **Parallelisation.** Due to the semi-explicit nature of the scheme, the existing computational programme is ideal for parallelisation (the implicit Poisson solver, in fact, has been coded in parallel). Computation of interaction forces is also a candidate for massive parallelisation using advanced architectures, such as GPGPU.

- **Fully explicit formulation.** The method is semi-explicit due to the implicit resolution of the incompressibility constraint. The most time consuming part is the solution of the anisotropic Poisson equation within each sub-iteration. Alternatively, a fully explicit formulation by means of an artificial compressibility approach can be developed in the case of small density ratio problems, of particular interest in the field of biomechanics.

- **Fully implicit formulation.** The fluid Cauchy stress is evaluated explicitly and the structure Cauchy stress is calculated implicitly by a fixed point iteration. Both stresses in the fluid and the structure can be evaluated implicitly via a Newton's method.

- **Multi-body dynamics.** The proposed fluid-rigid body interaction is limited to rigid bodies without contact. By introducing new constraints to describe the interaction of multiple rigid bodies, multi-body dynamics with fluid interaction can be performed. Typical applications include wave-power generators.

- **Adaptive mesh refinement with an Octree structure.** An octree is a data structure in which each internal node has exactly eight children. The partition three-dimensional space could be used for adaptive mesh refinement.

- **Constitutive models.** So far just a few constitutive models have been used within the present formulation, including an isotropic Neo-Hookean model. Other material models can also be considered, including visco-elasticity and visco-plasticity.

- **Free surface model.** In some coastal engineering simulations involving air, water and a structure, the influence of the air can be regarded as a constant atmospheric pressure on the free surface. In these cases, a free surface model can be implemented into the fluid solver, which will largely reduce the computational domain and the condition number of the global Poisson problem.

Part VI

Appendices

Appendix A

Linear algebra

This appendix reviews some fundamentals of linear algebra which are employed in this thesis. The use of a Cartesian coordinate system is sufficient for applications in this thesis. A more in-depth explanation can be found in [28].

A.1 Vectors

Let $\boldsymbol{e}_1$, $\boldsymbol{e}_2$ and $\boldsymbol{e}_3$ denote the three base vectors in the Cartesian space, any given vector $\boldsymbol{u}$ can be expressed as a linear combination of these vectors as

$$\boldsymbol{u} = \sum_{i=1}^{3} u_i \boldsymbol{e}_i, \quad \boldsymbol{u} = [u_1, u_2, u_3]^T. \tag{A.1}$$

If Einstein convention is used, whereby repetition of an index automatically implies its summation, the above equation can be written as

$$\boldsymbol{u} = u_i \boldsymbol{e}_i. \tag{A.2}$$

A.1.1 Inner product, norm, orthogonality and cross product

The inner, dot or scalar product of two arbitrary vectors is defined by

$$\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{u}^T \boldsymbol{v} = u_i v_i. \tag{A.3}$$

Then,

$$\|\boldsymbol{u}\| = \sqrt{\boldsymbol{u} \cdot \boldsymbol{u}} \tag{A.4}$$

defines the Euclidean norm, magnitude or length of a vector $\boldsymbol{u}$ and, moreover $\boldsymbol{u}$ is said to be a unit vector if

$$\|\boldsymbol{u}\| = 1. \tag{A.5}$$

A vector $\boldsymbol{u}$ is said to be orthogonal to a vector $\boldsymbol{v}$ if

$$\boldsymbol{u} \cdot \boldsymbol{v} = 0. \tag{A.6}$$

The vector or cross product of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is a new vector $\boldsymbol{u} \times \boldsymbol{v}$ which is defined as

$$\boldsymbol{u} \times \boldsymbol{v} = \begin{bmatrix} u_2v_3 - u_3v_2 \\ u_3v_1 - u_1v_3 \\ u_1v_2 - u_2v_1 \end{bmatrix} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}. \tag{A.7}$$

This equation can be written as

$$\boldsymbol{u} \times \boldsymbol{v} = \tilde{\boldsymbol{u}}\boldsymbol{v}, \tag{A.8}$$

where $\tilde{\boldsymbol{u}}$ is the skew-symmetric matrix associated with the vector $\boldsymbol{u}$ and is defined as

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix}. \tag{A.9}$$

A.2 Second order tensors

Any second order tensor is a linear transformation from $\mathbb{U}$ into $\mathbb{U}$; that is, $\boldsymbol{L} : \mathbb{U} \longrightarrow \mathbb{U}$ where $\mathbb{U}$ denotes a vector space. It maps a vector $\boldsymbol{u}$ into another vector $\boldsymbol{v}$ such as

$$\boldsymbol{v} = \boldsymbol{L}\boldsymbol{u}, \tag{A.10}$$

In particular,

$$\boldsymbol{0}\boldsymbol{u} = \boldsymbol{0}; \qquad \boldsymbol{I}\boldsymbol{u} = \boldsymbol{u}, \tag{A.11}$$

where $\boldsymbol{0}$ represents the zero tensor and $\boldsymbol{I}$ denotes the identity tensor.

A.2.1 Symmetric and skew symmetric tensors

An arbitrary second order tensor $\boldsymbol{L}$ can be conveniently decomposed into

$$\boldsymbol{L} = \text{sym}(\boldsymbol{L}) + \text{skew}(\boldsymbol{L}). \tag{A.12}$$

Their symmetric and skew symmetric parts are denoted by

$$\text{sym}(\boldsymbol{L}) = \frac{1}{2}(\boldsymbol{L} + \boldsymbol{L}^T); \qquad \text{skew}(\boldsymbol{L}) = \frac{1}{2}(\boldsymbol{L} - \boldsymbol{L}^T). \tag{A.13}$$

A.2.2 Tensor product

The tensor product of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted by $\boldsymbol{u} \otimes \boldsymbol{v}$ and is defined as

$$\boldsymbol{u} \otimes \boldsymbol{v} = \boldsymbol{u}\boldsymbol{v}^T. \tag{A.14}$$

Such resulting second order tensor maps an arbitrary vector $\boldsymbol{w}$ into the vector $(\boldsymbol{w} \cdot \boldsymbol{v})\boldsymbol{u}$ as

$$(\boldsymbol{u} \otimes \boldsymbol{v})\boldsymbol{w} = (\boldsymbol{w} \cdot \boldsymbol{v})\boldsymbol{u}. \tag{A.15}$$

This is also known as a dyadic process.

A.2.3 Trace, Inner Product and Euclidean Norm

The trace of the tensor $\boldsymbol{u} \otimes \boldsymbol{v}$ is defined by

$$\mathrm{tr}(\boldsymbol{u} \otimes \boldsymbol{v}) = \boldsymbol{u} \cdot \boldsymbol{v}. \tag{A.16}$$

For a generic second order tensor $\boldsymbol{L}$, its trace is defined as the summation of the diagonal terms of its matrix representation, that is

$$\mathrm{tr}\boldsymbol{L} = L_{ii}. \tag{A.17}$$

The inner product of two second order tensors, $\boldsymbol{T}\colon \boldsymbol{L}$, is described as

$$\boldsymbol{T}\colon \boldsymbol{L} = \mathrm{tr}(\boldsymbol{T}^T\boldsymbol{L}) = \mathrm{tr}(\boldsymbol{T}\boldsymbol{L}^T) = T_{ij}L_{ij}. \tag{A.18}$$

The Euclidean or Frobenius norm of a second order tensor $\boldsymbol{L}$ is then defined by

$$\|\boldsymbol{L}\| = \sqrt{\boldsymbol{L}\colon \boldsymbol{L}} = \left(\sum_{i,j=1}^{3} L_{ij}^2\right)^{1/2}. \tag{A.19}$$

A.2.4 Spectral Decomposition

Given a second order tensor $\boldsymbol{L}$, a non-zero vector $\boldsymbol{n}$ is said to be an eigenvector of $\boldsymbol{L}$ associated with the eigenvalue ω if

$$\boldsymbol{L}\boldsymbol{n} = \omega\boldsymbol{n}. \tag{A.20}$$

Let $\boldsymbol{L}$ be a symmetric second order tensor, then it admits the representation

$$\boldsymbol{L} = \sum_{\alpha=1}^{3} \omega_\alpha \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha. \tag{A.21}$$

Here, $\{\boldsymbol{n}_1, \boldsymbol{n}_2, \boldsymbol{n}_3\}$ are eigenvectors of $\boldsymbol{L}$, which can also be used as an alternative Cartesian base and $\{\omega_1, \omega_2, \omega_3\}$ denotes the set of corresponding eigenvalues. The

above expression is called the spectral decomposition of $\boldsymbol{L}$.

Appendix B

Comparison of convection schemes

This Appendix addresses some of the methods available for solving transient convection problems, typically for the Level Set Eq. (3.33). The Level Set equation is, in general, a nonlinear advection equation, as the velocity field $\boldsymbol{u}$ and the level set ϕ are interrelated. If $\boldsymbol{u}$ is independent of ϕ, then the Level Set equation is a linear hyperbolic equation with time-space-varying velocity field. A brief introduction of Finite Volume based methods [82] as well as a characteristic line based method [155] is presented in this Appendix.

B.1 Spatial and temporal discretisation

B.1.1 Runge-Kutta time integrator for temporal discretisation

In the case of transient problems, the temporal discretisation involves the integration of the time-derivative terms over a time step Δt. The spatial domain can initially be discretised to produce a semi-discrete form as follows

$$\frac{\partial \phi(t)}{\partial t} = F(\phi(t)). \tag{B.1}$$

Runge-Kutta methods are multistage methods that make use of the solution ϕ^n at time t^n to compute the solution ϕ^{n+1} at t^{n+1}. This is achieved by computing a number of intermediate values of the time derivative of the unknown ϕ, within the interval $\Delta t = t^{n+1} - t^n$. If we define the forward Euler operator $\mathcal{L}_h$ as $\mathcal{L}_h\phi^n = \phi^n + \Delta t F(\phi^n)$, the two-stage Runge-Kutta scheme would be described as follows refer to Figure B.1.

- Step 1: Solve forward for $\phi^1 = \mathcal{L}_h\phi^n$
- Step 2: Solve forward for $\phi^{n+1} = \frac{1}{2}\phi^n + \frac{1}{2}(\mathcal{L}_h\phi^1)$

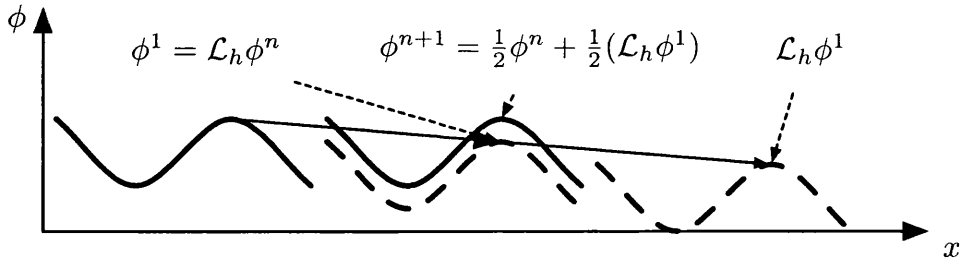


Figure B.1: Two-stage Runge-Kutta method

B.1.2 Finite Volume for spatial discretisation

To derive the Finite Volume spatial discretisation of the Level Set equation, we first re-write the Level Set equation in an integral form:

$$\int_\Omega \frac{\partial \phi}{\partial t}\, dv + \int_\Omega \nabla \cdot (\boldsymbol{u}\phi)\, dv = 0. \tag{B.2}$$

Appplication of the Gauss divergence theorem yields

$$\int_\Omega \frac{\partial \phi}{\partial t}\, dv + \int_{\partial\Omega} F_\phi\, da = 0, \tag{B.3}$$

which represents the conservation of a scalar ϕ transport equation, where $F_\phi = \phi \boldsymbol{u}\cdot\boldsymbol{n}$ is the flux of the Level Set function.

Within the framework of the Finite Volume scheme, Ω can be regarded as a control volume. The convective components of the numerical fluxes, are obtained as a first order upwind scheme or via the second order QUICK scheme. As time and space are discretised independently, this approach is called 'method of lines' [156].

First order upwind

In the particular one-dimensional setting, and for the linear advection equation (B.3), the so-called numerical upwind flux as stated by the Godunov method [82] is defined as

$$-\frac{u}{\Delta x}(\phi_E - \phi_P), \tag{B.4}$$

where ϕ_P and ϕ_E is a face value and its east neighbour values, respectively.

Quadratic Upstream Interpolation for Convective Kinematics

The Quadratic Upstream Interpolation for Convective Kinematics (QUICK) [100] uses a three-point upstream-weighted quadratic interpolation for the numerical flux cell face values. The face value of ϕ is obtained from a quadratic function passing through two bracketing nodes (on each side of the face) and a node on the upstream side.

For face e in figure B.2 and assuming the flow is from left to right, the face value ϕ_e can be written as :

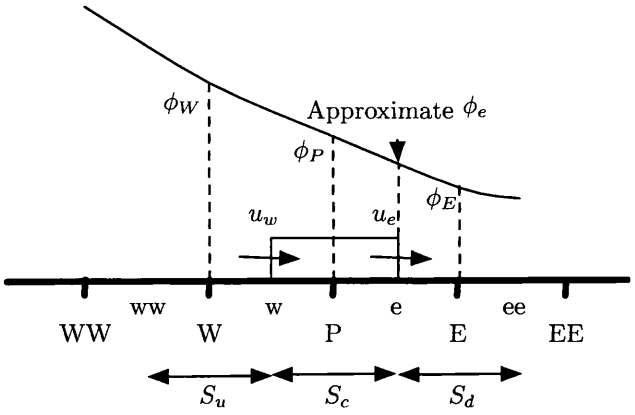


Figure B.2: The Quadratic upstream interpolation

$$\phi_e = \psi \left[\frac{S_d}{S_c + S_d}\phi_P + \frac{S_c}{S_c + S_d}\phi_E \right] + (1 - \psi) \left[\frac{S_u + 2S_c}{S_u + S_c}\phi_P + \frac{S_c}{S_u + S_c}\phi_W \right], \quad \text{(B.5)}$$

where $\psi = 1$ in the above equation results in a central second-order interpolation while $\psi = 0$ yields a second-order upwind scheme. The traditional QUICK scheme is obtained by setting $\psi = 1/8$ [82].

B.2 Semi-Lagrangian Method

Equation (3.33) can alternatively be solved in space-time using a numerical method inspired in the method of characteristics [156]. This method is called 'semi-Lagrangian' [157] because once the characteristic curve is identified, the path-integration is carried in a Lagrangian manner (following the same particle). This scheme is known for allowing large time stepping.

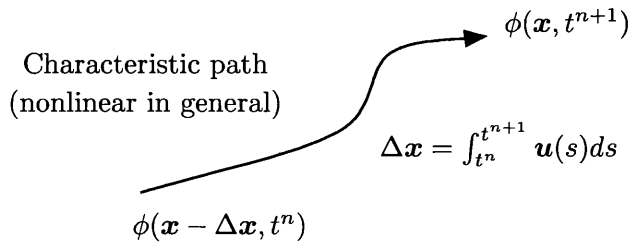


Figure B.3: Illustrate of the characteristic line

As the advection equation is homogeneous,

$$\phi(\boldsymbol{x}, t^{n+1}) = \phi(\boldsymbol{x} - \Delta \boldsymbol{x}, t^n). \quad \text{(B.6)}$$

Notice that once $\Delta \boldsymbol{x}$ is computed, $\phi(\boldsymbol{x} - \Delta \boldsymbol{x}, t_n)$ requires evaluation. In general,

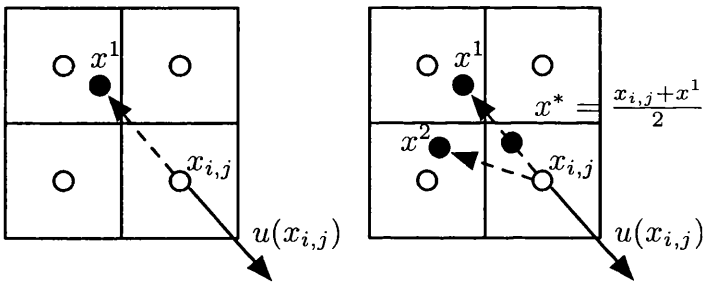
spatial interpolation is required to obtain $\phi(\boldsymbol{x} - \Delta\boldsymbol{x}, t_n)$.

Equation (B.6) can also be written as

$$\phi(\boldsymbol{x}, t^{n+1}) = \phi\left(\left(\boldsymbol{x} - \int_{t^*}^{t^{n+1}} \boldsymbol{u}(s)\, ds\right), t^* \in [t^n, t^{n+1}]\right). \tag{B.7}$$

B.2.1 First order and second order semi-Lagrangian method

The first order semi-Lagrangian method is the simplest of them all and it is unconditional stable. That is the reason why the semi-Lagrangian method is widely used in numerical weather prediction for large CFL numbers [155]. It updates the value of ϕ at location $\boldsymbol{x}$ by sampling ϕ where the fluid at $\boldsymbol{x}$ was located in the previous time step. The first order semi-Lagrangian method is comprised of the following



(a) 1st order (b) 2nd order

Figure B.4: First order and second order semi-Lagrangian method

steps:

- Find the interpolation point by $\boldsymbol{x}^{(1)} = \boldsymbol{x}_{i,j} - \Delta t \boldsymbol{u}(\boldsymbol{x}_{i,j}, t^{n+1})$.
- Then interpolation $\phi_{i,j}^{n+1} = \phi^n(\boldsymbol{x}^1)$.

Unfortunately, the first order semi-Lagrangian introduces undesired numerical dissipation. To achieve second order accuracy in time, the linear backward tracing scheme is replaced by a more accurate second-order Runge-Kutta scheme, as described with the following steps

- Find the first interpolation point by $\boldsymbol{x}^{(1)} = \boldsymbol{x}_{i,j} - \Delta t \boldsymbol{u}(\boldsymbol{x}_{i,j}, t^{n+1})$.
- Find the intermediate interpolation point $\boldsymbol{x}^* = \frac{\boldsymbol{x}_{i,j}+\boldsymbol{x}^{(1)}}{2}$.
- The 2nd order accurate interpolation point $\boldsymbol{x}^{(2)} = \boldsymbol{x}_{i,j} - \Delta t \boldsymbol{u}(\boldsymbol{x}^*, t^{n+1/2})$.
- Then interpolation $\phi_{i,j}^{n+1} = \phi^n(\boldsymbol{x}^2)$.

B.2.2 Back and Forth Error Compensation and Correction

Here we review the Back and Forth Error Compensation and Correction method (BFECC) by Dupont T, Liu Y [158]. This approach can improve the accuracy of the Level Set resolution, which is proven to be second order accurate both in space and time. The non zero curvature of the interface will result in uneven gradients of the Level Set function which can introduce extra numerical error. The Level Set function ϕ first move forward in time and then backward. The difference between the back and forth Level Set is used to compensate the error.

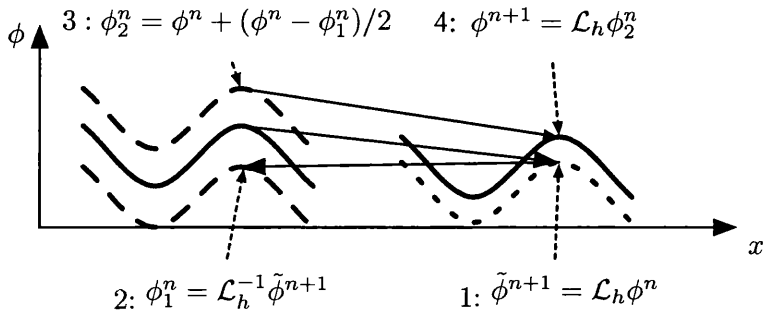


Figure B.5: Back and Forth Error Compensation and Correction (BFECC)

For a given ϕ^n at time level t_n, we have the following algorithm. If the first order semi-Lagrangian scheme is written as $\phi(\boldsymbol{x}, t_{n+1}) = \mathcal{L}(\phi(\boldsymbol{x}, t_n))$ in short, then the BFECC is given by

- Solve forward for $\tilde{\phi}^{n+1} = \mathcal{L}_h \phi^n$.
- Solve backward for $\phi_1^n = \mathcal{L}_h^{-1} \tilde{\phi}^{n+1}$.
- Let $\phi_2^n = \phi^n + (\phi^n - \phi_1^n)/2$.
- Solve forward for ϕ^{n+1} using $\phi^{n+1} = \mathcal{L}_h \phi_2^n$.

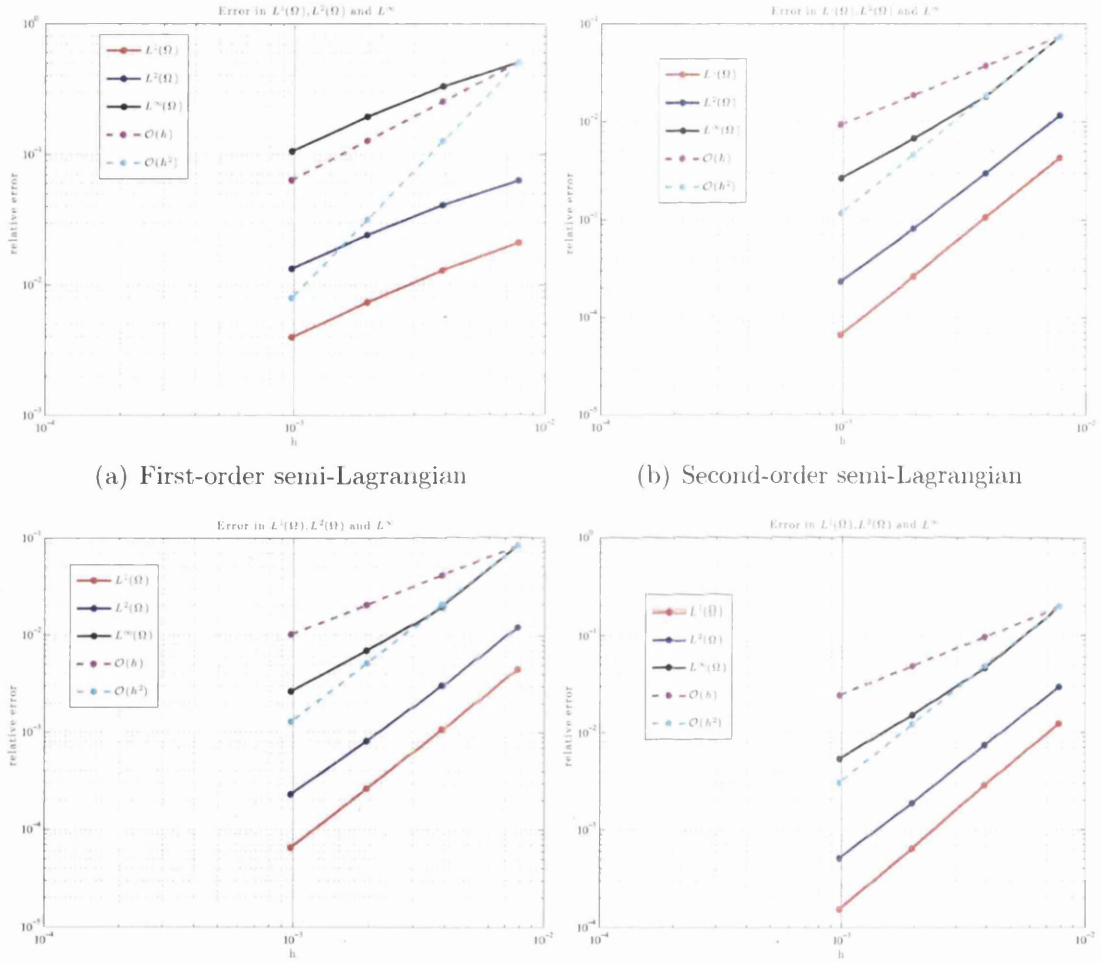
B.3 Numerical study

B.3.1 Convergence tests

In order to illustrate the performance of the different convection schemes, we convect a two-dimensional sine hill defined as

$$\phi = \begin{cases} \frac{1}{4}(1 + \sin(4\pi(x - \frac{3}{8})))(1 + \sin(4\pi(y - \frac{3}{8}))) & 0.25 < x, y < 0.75 \\ 0 & 0 \end{cases} \tag{B.8}$$

using a constant velocity field $\boldsymbol{u} = (4, 3)^T$. The problem is discretised over the domain $[0, 3] \times [0, 2.5]$ and four algorithms are tested for convergence. As shown in Figure B.6, first-order semi-Lagrangian method exhibits a first order convergence, whereas the rest exhibit the expected second order convergence.



(a) First-order semi-Lagrangian

(b) Second-order semi-Lagrangian

(c) BFECC semi-Lagrangian

(d) Two stage Runge-Kutta with QUICK

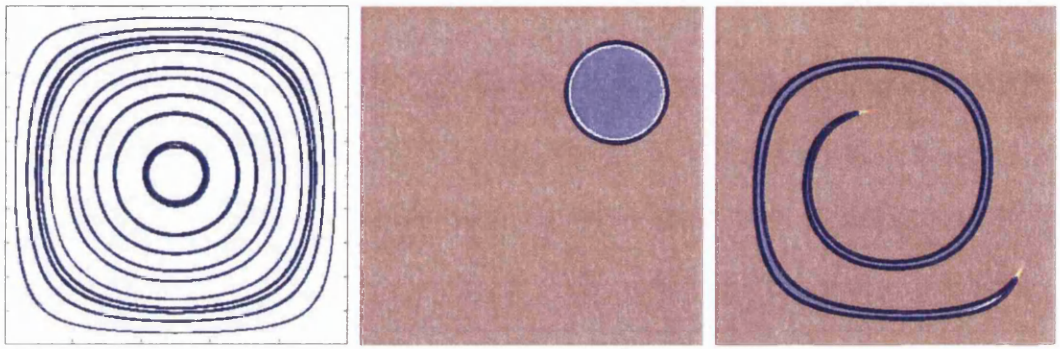
Figure B.6: Comparison of the convergence rate for different convective schemes.

B.3.2 Qualitative tests

To address the performance of Level Set method, several authors have examined the passive advection of a patch of fluid by a given velocity field. Two test have emerged as the most popular ones.

Test 1: Vortical flow

This is the vortical flow introduced by Rider and Kothe [19] and is a common benchmark problem for the Level Set Method, VOF method and particle based methods. In this problem, the velocity field is defined by the steam function: $\phi = 1/\pi \sin^2(\pi x) \sin^2(\pi y)$, which gives an almost vortical flow in an enclosed domain, as shown in Figure B.7.



(a) Velocity Streamline (b) Initial state (c) Final state

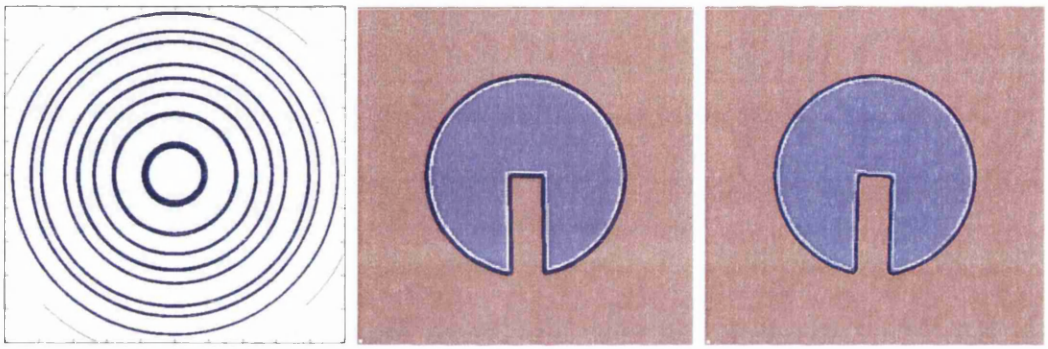
Figure B.7: The advection test case introduced by Rider and Kothe [19]. The circular blob near the top is deformed in a vortical flow field. The domain is given by $\Omega = [0, 1] \times [0, 1]$. The underformed blob is circular, with a radius of 0.15 and is initially located at $0.75, 0.75$, with mesh size 100×100.

Test 2: Rotation of a Zalesak disk

The second test is that of the rotation of disk known as Zalesak's disk, which is introduced in [20]. A circular blob, with a rectangular cutting thorough the circle, is rotated with a fluid velocity (See Figure B.8b). The domain is set to be $\Omega = [01] \times [01]$. The constant vorticity velocity field is given by $u = (\pi/314)(0.5 - y); v = (\pi/314)(x - 0.5)$. So that the disk completes one revolution every 628 time units.

B.4 Conclusions

We have tested four different discretisation schemes for the convection equation of the Level Set. Except for the 1st order semi-Lagrangian method, the rest three schemes (Second order semi-Lagrangian method, BEFCC semi-Lagrangian method and the two stage Runge-kutta with QUICK) all exhibit second order convergence.



(a) Velocity Streamline (b) Initial state (c) Final state

Figure B.8: The advection test case introduced by Zalesak [20]. The disk is initially located at $0.5, 0.5$, with mesh size 100×100.

Appendix C

One-dimensional temporal stability and error analysis

This appendix is devoted to the stability and truncation error analysis of the one-dimensional linear equation by the consideration of different space-time discretisation schemes. In spite of the restriction to a one-dimensional linear analysis, useful insight will be gained, which can be applicable to higher dimensional problems.

C.1 Temporal stability

Let us first consider the study of temporal stability for time-integration methods. Only explicit methods will be considered, all conditionally stable. Let us consider the modal equation

$$\frac{dq}{dt} = \lambda q, \tag{C.1}$$

where λ denotes a typical complex eigenvalue. Eq. C.1 is analysed for three different discretisation techniques, namely, forward Euler, two-stage-Runge-Kutta and backward forward error compensation and correction (BFECC) methods. Applying Taylor expansion to Eq. C.1, we arrive at

$$q^{n+1} = q^n + \lambda \Delta t q^n + \frac{\lambda^2 \Delta t^2}{2} q^n + + \frac{\lambda^3 \Delta t^3}{6} q^n + \mathcal{O}(\Delta t^4) q^n. \tag{C.2}$$

Forward Euler

Application of the forward Euler scheme to Eq. C.1 yields:

$$q^{n+1} = (1 + \Delta t \lambda) q^n, \tag{C.3}$$

where $(1 + \Delta t \lambda)$ is typically known as the numerical amplification factor. In order for the numerical solution to remain bounded, the following condition should be fulfilled

$$|1 + \Delta t \lambda| \leqq 1, \tag{C.4}$$

The complex expantion of $\Delta t\lambda = x + yi$ into above Eq. (C.4) , yields

$$(x+1)^2 + y^2 \leqq 1. \tag{C.5}$$

Two stage Runge-Kutta

A similar procedure as that followed above is performed. For the two stage Runge-Kutta time integration scheme (Figure. B.1), it yields,

$$\begin{aligned} \hat{q}^{n+1} &= (1+\Delta t\lambda)q^n \\ q^{n+1} &= \tfrac{1}{2}(q^n + 1 + \Delta t\lambda)\hat{q}^n \\ q^{n+1} &= \tfrac{1}{2}(q^n + (1+\Delta t\lambda)^2 q^n) = (1 + \Delta t\lambda + \tfrac{1}{2}\Delta t^2\lambda^2)q^n. \end{aligned} \tag{C.6}$$

In order for the numerical solution to remain bounded, the following condition should be fulfilled

$$|(1 + \Delta t\lambda + \frac{1}{2}\Delta t^2\lambda^2)| \leqq 1 \tag{C.7}$$

Substituting $\Delta t\lambda = x + yi$ into above Eq. (C.7), yields

$$\left(1 + x + \frac{1}{2}x^2 - \frac{1}{2}y^2\right)^2 - (xy + y)^2 \leqq 1 \tag{C.8}$$

Back and forth error compensation and correction (BFECC)

For the backward forward error compensation and correction (BFECC) scheme (Figure B.5), we obtain:

$$\begin{aligned} \hat{q}^{n+1} &= (1+\Delta t\lambda)q^n \\ q_1 &= (1-\Delta t\lambda)\hat{q}^{n+1} \\ q_2 &= q^n + (q^n - q_1) \\ q^{n+1} &= (1+\Delta t\lambda)q_2. \end{aligned} \tag{C.9}$$

Thus,

$$q^{n+1} = \left(1 + \Delta t\lambda + \frac{1}{2}\Delta t^2\lambda^2 + \frac{1}{2}\Delta t^3\lambda^3\right) q^n. \tag{C.10}$$

In order for the numerical solution to remain bounded, the following condition should be fulfilled

$$|(1 + \Delta t\lambda + \frac{1}{2}\Delta t^2\lambda^2 + \frac{1}{2}\Delta t^3\lambda^3)| \leqq 1. \tag{C.11}$$

Substituting $\Delta t\lambda = x + yi$ into above Eq. (C.11), yields, yields

$$\left(1 + x + \frac{1}{2}x^2 - \frac{1}{2}y^2 + \frac{1}{2}x^3 - \frac{3}{2}xy^2\right)^2 + \left(xy + y + \frac{3}{2}x^2y - \frac{1}{2}y^3\right)^2 \leqq 1. \tag{C.12}$$

The stability regions of Eqs. (C.5), (C.8) and (C.12) are illustrated in Figure C.1, where the x and y axis represent the real and imaginary parts.

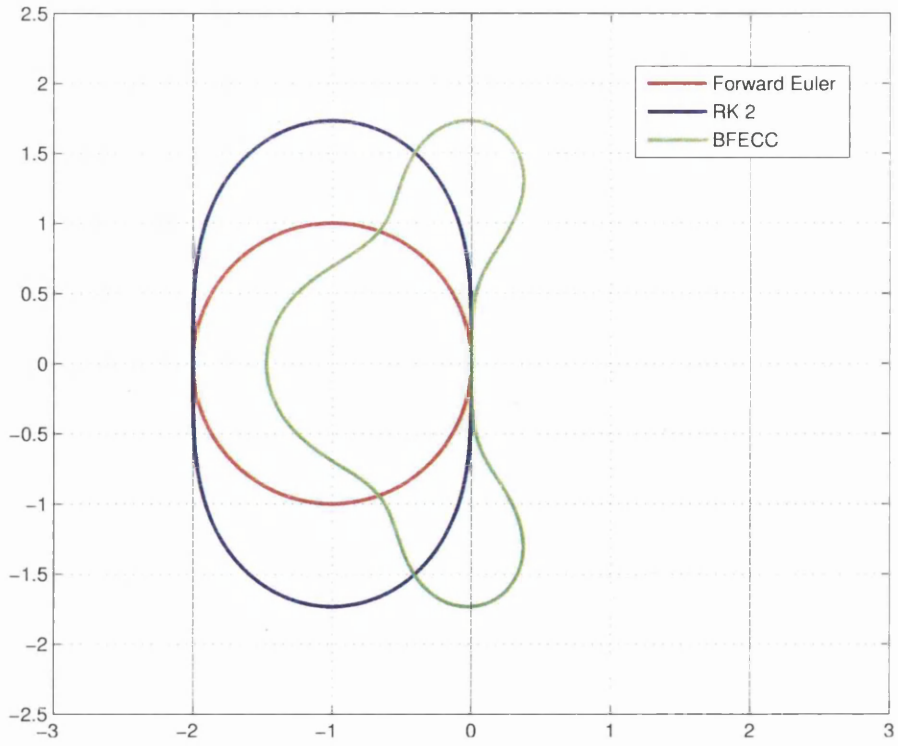


Figure C.1: Stability regions for the ordinary differential equation using different numerical schemes, where x is the real part of $\lambda\Delta t$ and y is the imaginary part of $\lambda\Delta t$.

C.2 Truncation error

Let us consider the one-dimensional advection equation

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0, \tag{C.13}$$

where q is a scalar variable and u is the speed with which it is advected.

In order to analyse the truncation error of Eq. (C.13), a *one step error* is introduced. This *one step error* indicates the amount of error introduced in a single time step

$$\textit{one step error} = \mathcal{N}(q^n) - q^{n+1}, \tag{C.14}$$

where q^{n+1} and q^n represent the true solution evaluated at time t^{n+1} and t^n. The *local truncation error* is defined by dividing the *one step error* by Δt as

$$\tau^n = \frac{1}{\Delta t}\left(\mathcal{N}(q^n) - q^{n+1}\right). \tag{C.15}$$

Temporal discretisation with forward Euler and first order upwind spatial discretisation

Application of the space-time discretisation scheme to Eq. (C.13) yields

$$q_i^{n+1} = q_i^n - \frac{\Delta t}{\Delta x} u(q_i^n - q_{i-1}^n). \tag{C.16}$$

The *local truncation error* is

$$\tau^n = \frac{1}{\Delta t}\left(q_i^n - \frac{\Delta t}{\Delta x} u(q_i^n - q_{i-1}^n) - q_i^{n+1}\right). \tag{C.17}$$

Recalling the Taylor series expansion for q_{i-1}^n, q_i^{n+1}

$$q_{i-1}^n = q_i^n - \left(\frac{\partial q}{\partial x}\right)_i^n \Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{\Delta x^2}{2} + \mathcal{O}(\Delta x^3), \tag{C.18}$$

$$q_i^{n+1} = q_i^n + \left(\frac{\partial q}{\partial t}\right)_i^n \Delta t + \left(\frac{\partial^2 q}{\partial t^2}\right)_i^n \frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3), \tag{C.19}$$

and the consideration of Eq. (C.13), the *local truncation error* is

$$\tau^n = \frac{u\Delta x}{2}(1-\lambda)\left(\frac{\partial^2 q}{\partial t^2}\right)_i^n + \mathcal{O}(\Delta t^3), \tag{C.20}$$

where $\lambda = u\frac{\Delta t}{\Delta x}$ is called Courant-Friedrich-Levy (CFL) number[1]. And therefore

$$\tau^n = \mathcal{O}(\Delta x) \tag{C.21}$$

resulting in a first order scheme.

First order semi-Lagrangian method

Assuming $\Delta t < \Delta x/u$, we consider the starting point of the characteristic trajectory that ends at grid point x_i in the interval $[x_{i-1}, x_i]$. Linear interpolation between q_{i-1}^n and q_i^n at point $x_i - \Delta t u$ gives

$$q_i^{n+1} = \frac{\Delta t u}{\Delta x} q_{i-1}^n + \left(1 - \frac{\Delta t u}{\Delta x}\right) q_i^n. \tag{C.22}$$

Or alternatively

$$q_i^{n+1} = q_i^n - \Delta t u \left(\frac{q_i^n - q_{i-1}^n}{\Delta x}\right), \tag{C.23}$$

[1]The variable Δx is removed with this CFL number.

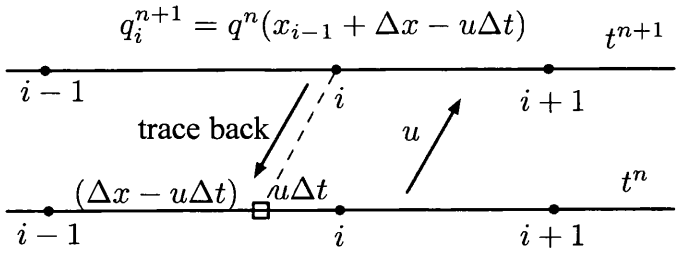


Figure C.2: Illustration of tje one-dimensional semi-Lagrangian method.

which coincides with a forward Euler in time and a one-sided finite difference in space. The *local truncation error* is then the same as above

$$\tau^n = \frac{u\Delta x}{2}(1-\lambda)\left(\frac{\partial^2 q}{\partial t^2}\right)_i^n + \mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta x^3), \tag{C.24}$$

which proves, as expected, that the method is first order.

Temporal discretisation using a BFECC and first order upwind spatial discretisation

Let us recall the upwind discretisation scheme in Eq. (C.16) defined above

$$q_i^{n+1} = (1-\lambda)q_i^n + \lambda q_{i-1}^n. \tag{C.25}$$

The BFECC method (introduced in the previous Appendix B.2.2) can be written as:
First a step forward in time

$$\hat{q}_i^{n+1} = (1-\lambda)q_i^n + \lambda q_i^n - 1, \tag{C.26}$$

then by a step backward in time

$$\hat{q}_i^{n+1} = (1-\lambda)\hat{q}_i^n + \lambda q_{i-1}^n, \tag{C.27}$$

followed by a correction of the original data using the estimated error

$$\hat{q}_i^n = q_i^n - \frac{1}{2}(\hat{q}_i^n - q_i^n) = q_i^n - \frac{1}{2}\lambda(1-\lambda)(q_{i+1}^n - 2q_i^n + q_{i-1}^n), \tag{C.28}$$

by a step forward in time using the error corrected data

$$\hat{q}_i^{n+1} = (1-\lambda)\hat{q}_i^n + \lambda\hat{q}_{i-1}^n. \tag{C.29}$$

After applying the above steps, the final numerical solution is evaluated as

$$\begin{aligned}\hat{q}_i^{n+1} &= \left(-\tfrac{1}{2}\lambda^2+\tfrac{1}{2}\lambda^3\right)q_{i-2}^n+\left(\tfrac{1}{2}\lambda+2\lambda^2-\tfrac{3}{2}\lambda^3\right)q_{i-1}^n \\ &+\left(1-\tfrac{5}{2}\lambda^2-\tfrac{3}{2}\lambda^3\right)q_i^n+\left(-\tfrac{1}{2}\lambda+\lambda^2-\tfrac{1}{2}\lambda^3\right)q_{i+1}^n.\end{aligned} \tag{C.30}$$

Therefore, the *local truncation error* is

$$\tau^n=\frac{1}{\Delta t}\left(\hat{q}_i^{n+1}-q_i^{n+1}\right). \tag{C.31}$$

Let us recall the Taylor series for q_{i-2}^n, q_{i-1}^n, q_{i+1}^n and q_i^{n+1}

$$q_{i-2}^n=q_i^n-\left(\frac{\partial q}{\partial x}\right)_i^n 2\Delta x+\left(\frac{\partial^2 q}{\partial x^2}\right)_i^n\frac{4\Delta x^2}{2}-\left(\frac{\partial^3 q}{\partial x^3}\right)_i^n\frac{8\Delta x^3}{6}+\mathcal{O}(\Delta x^4), \tag{C.32a}$$

$$q_{i-1}^n=q_i^n-\left(\frac{\partial q}{\partial x}\right)_i^n\Delta x+\left(\frac{\partial^2 q}{\partial x^2}\right)_i^n\frac{\Delta x^2}{2}-\left(\frac{\partial^3 q}{\partial x^3}\right)_i^n\frac{\Delta x^3}{6}+\mathcal{O}(\Delta x^4), \tag{C.32b}$$

$$q_{i+1}^n=q_i^n+\left(\frac{\partial q}{\partial x}\right)_i^n\Delta x+\left(\frac{\partial^2 q}{\partial x^2}\right)_i^n\frac{\Delta x^2}{2}+\left(\frac{\partial^3 q}{\partial x^3}\right)_i^n\frac{\Delta x^3}{6}+\mathcal{O}(\Delta x^4), \tag{C.32c}$$

$$q_i^{n+1}=q_i^n+\left(\frac{\partial q}{\partial t}\right)_i^n\Delta t+\left(\frac{\partial^2 q}{\partial t^2}\right)_i^n\frac{\Delta t^2}{2}+\left(\frac{\partial^3 q}{\partial t^3}\right)_i^n\frac{\Delta t^3}{6}+\mathcal{O}(\Delta t^4). \tag{C.32d}$$

After some algebraic modifications, we arrive of the following expression for the *local truncation error*

$$\tau^n=\frac{\Delta t^2}{6}\left(\frac{\partial^3 q}{\partial t^3}\right)_i^n\left(-\frac{3}{\lambda}+\frac{1}{\lambda^2}+3\right)+\mathcal{O}(\Delta x^3). \tag{C.33}$$

And therefore

$$\tau^n=\mathcal{O}(\Delta t^2), \tag{C.34}$$

which proves that it is a second order scheme.

Temporal discretisation using a two-stage Runge-Kutta and spatial discretisation with the QUICK scheme

The QUICK spatial discretisation of the flux is given by

$$\frac{\lambda}{8}(q_{i-2}-7q_{i-1}+3q_i+3q_{i+1}). \tag{C.35}$$

The two-stage Runge-Kutta method is

$$\hat{q}_i^{n+1}=q_i^n-\frac{\lambda}{8}(q_{i-2}^n-7q_{i-1}^n+3q_i^n+3q_{i+1}^n), \tag{C.36a}$$

$$q_i^{n+1}=\frac{1}{2}q_i^n-\frac{\lambda}{8}(\hat{q}_{i-2}^n-7\hat{q}_{i-1}^n+3\hat{q}_i^n+3\hat{q}_{i+1}^n), \tag{C.36b}$$

$$\begin{aligned} q_i^{n+1} &= q_i^n - \lambda(-\tfrac{\lambda}{128}q_{i-4}^n - \tfrac{14\lambda}{128}q_{i-3}^n + \tfrac{8-55\lambda}{128}q_{i-2}^n \\ &+ \tfrac{-56+36\lambda}{128}q_{i-1}^n + \tfrac{24}{128}q_i^n + \tfrac{24-18\lambda}{128}q_{i+1}^n + \tfrac{-9\lambda}{128}q_{i+2}^n). \end{aligned} \tag{C.37}$$

After some simplification, the *local truncation error* is

$$\tau^n = \frac{1}{\Delta t}\left(\hat{q}_i^{n+1} - q_i^{n+1}\right). \tag{C.38}$$

Recalling the Taylor series for q_{i-4}^n, q_{i-3}^n, q_{i-2}^n, q_{i-1}^n, q_{i+1}^n, q_{i+2}^n and q_i^{n+1}

$$q_{i-4}^n = q_i^n - \left(\frac{\partial q}{\partial x}\right)_i^n 4\Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{16\Delta x^2}{2} - \left(\frac{\partial^3 q}{\partial x^3}\right)_i^n \frac{64\Delta x^3}{6} + \mathcal{O}(\Delta x^4), \tag{C.39a}$$

$$q_{i-3}^n = q_i^n - \left(\frac{\partial q}{\partial x}\right)_i^n 3\Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{9\Delta x^2}{2} - \left(\frac{\partial^3 q}{\partial x^3}\right)_i^n \frac{27\Delta x^3}{6} + \mathcal{O}(\Delta x^4), \tag{C.39b}$$

$$q_{i-2}^n = q_i^n - \left(\frac{\partial q}{\partial x}\right)_i^n 2\Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{4\Delta x^2}{2} - \left(\frac{\partial^3 q}{\partial x^3}\right)_i^n \frac{8\Delta x^3}{6} + \mathcal{O}(\Delta x^4), \tag{C.39c}$$

$$q_{i-1}^n = q_i^n - \left(\frac{\partial q}{\partial x}\right)_i^n \Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{\Delta x^2}{2} - \left(\frac{\partial^3 q}{\partial x^3}\right)_i^n \frac{\Delta x^3}{6} + \mathcal{O}(\Delta x^4), \tag{C.39d}$$

$$q_{i+1}^n = q_i^n + \left(\frac{\partial q}{\partial x}\right)_i^n \Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{\Delta x^2}{2} + \left(\frac{\partial^3 q}{\partial x^3}\right)_i^n \frac{\Delta x^3}{6} + \mathcal{O}(\Delta x^4), \tag{C.39e}$$

$$q_{i+2}^n = q_i^n + \left(\frac{\partial q}{\partial x}\right)_i^n \Delta x + \left(\frac{\partial^2 q}{\partial x^2}\right)_i^n \frac{4\Delta x^2}{2} + \left(\frac{\partial^3 q}{\partial x^3}\right)_i^n \frac{8\Delta x^3}{6} + \mathcal{O}(\Delta x^4), \tag{C.39f}$$

$$q_i^{n+1} = q_i^n + \left(\frac{\partial q}{\partial t}\right)_i^n \Delta t + \left(\frac{\partial^2 q}{\partial t^2}\right)_i^n \frac{\Delta t^2}{2} + \left(\frac{\partial^3 q}{\partial t^3}\right)_i^n \frac{\Delta t^3}{6} + \mathcal{O}(\Delta t^4). \tag{C.39g}$$

Employing the above, we arrive at

$$\tau^n = \Delta t^2 \left(\frac{\partial^3 q}{\partial t^3}\right)_i^n (H(\lambda)) + \mathcal{O}(\Delta t^3), \tag{C.40}$$

where $H(\lambda)$ is a polynomial in the CFL number λ. Therefore,

$$\tau^n = \mathcal{O}(\Delta t^2), \tag{C.41}$$

and the method is second order accurate.

Bibliography

[1] J. Sarrate, A. Huerta, and J. Donea. Arbitrary Lagrangian-Eulerian formulation for fluid-rigid body interactions. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3171–3188, 2001.

[2] W. Dettmer and D. Peric. A computational framework for fluid-rigid body interaction: Finite element formulation and applications. *Computer Methods in Applied Mechanics and Engineering*, 195(13-16):1633–1666, 2006.

[3] C. S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[4] A. J. Gil, A. Arranz Carreño, J. Bonet, and O. Hassan. The Immersed Structural Potential Method for haemodynamic applications. *Journal of Computational Physics*, 229(22):8613–8641, 2010.

[5] A. J. Gil, A. Arranz Carreño, J. Bonet, and O. Hassan. An enhanced Immersed Structural Potential Method for fluid-structure interactions. *Journal of Computational Physics*, 250(0):178–205, 2013.

[6] E. Walhorn, A. Kölke, B. Hübner, and D. Dinkler. Fluid-Structure coupling within a monolithic model involving free surface flows. *Computers and Structures*, 83(25-26):2100–2111, #sep# 2005.

[7] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005.

[8] T. Ye, R. Mittal, H. Udaykumar, and W. Shyy. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *Journal of Computational Physics*, 156(2):209–240, 1999.

[9] Y. H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, 2003.

[10] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. *Journal of Computational Physics*, 209(2):448–476, 2005.

[11] J. Yang and F. Stern. A simple and efficient direct forcing immersed boundary framework for fluid-structure interactions. *Journal of Computational Physics*, 231(15):5029–5061, 2012.

[12] C. Crowe, T. Troutt, and J. Chung. Numerical models for two-phase turbulent flows. *Annual Review of Fluid Mechanics*, 28(1):11–43, 1996.

[13] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146–159, 1994.

[14] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of fluids*, 8(12):2182, 1965.

[15] Y. Zhao, H. H. Tan, and B. Zhang. A High-Resolution Characteristics-Based Implicit Dual Time-Stepping VOF Method for Free Surface Flow Simulation on Unstructured grids. *Journal of Computational Physics*, 183(1):233–273, 2002.

[16] S.-R. Hysing and S. Turek. *Evaluation of commercial and academic CFD codes for a two-phase flow benchmark test case.* Techn. Univ., Fak. für Mathematik, 2012.

[17] A. Shakibaeinia and Y. C. Jin. A mesh-free particle model for simulation of mobile-bed dam break. *Advances in Water Resources*, 34(6):794–807, 2011.

[18] M. Schafer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark computations of laminar flow around a cylinder. In E. Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 48 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 547–566. Vieweg+ Teubner Verlag, 1996.

[19] D. K. William Rider. Stretching and tearing interface tracking methods. In *12th Computational Fluid Dynamics Conference.* American Institute of Aeronautics and Astronautics, 1995.

[20] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31(3):335–362, 1979.

[21] R. Temam and A. Miranville. *Mathematical Modeling in Continuum Mechanics.* Cambridge University Press, 2000.

[22] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*, volume 3. Springer Berlin, 1996.

[23] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of computation*, 22(104):745–762, 1968.

[24] J. Marshall, A. Adcroft, C. Hill, L. Perelman, and C. Heisey. A finite-volume, incompressible Navier-Stokes model for studies of the ocean on parallel computers. *Journal of Geophysical Research: Oceans*, 102(C3):5753–5766, 1997.

[25] A. N. Brooks and T. J. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the

incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1):199–259, 1982.

[26] F. Bassi and S. Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 131(2):267–279, 1997.

[27] S. Gross and A. Reusken. *Numerical Methods for Two-phase Incompressible Flows*. Springer Series in Computational Mathematics. Springer, 2011.

[28] J. Bonet. *Nonlinear continuum mechanics for Finite Element analysis*. Cambridge University Press, 1997.

[29] D. Wilcox. *Turbulence Modeling for CFD*. Turbulence Modeling for CFD. DCW Industries, Incorporated, 2006.

[30] H. Schlichting, K. Gersten, and K. Gersten. *Boundary-layer theory*. Springer Science & Business Media, 2000.

[31] G. Iaccarino, A. Ooi, P. Durbin, and M. Behnia. Reynolds averaged simulation of unsteady separated flow. *International Journal of Heat and Fluid Flow*, 24(2):147–156, 2003.

[32] P. Sagaut. *Large Eddy Simulations for Incompressible Flows*, volume 3. Springer Berlin, 2000.

[33] A. J. Gil, Z. Zhang, O. Hassan, and K. Morgan. Parallel multigrid detached eddy simulation algorithm for three-dimensional unsteady incompressible flows on unstructured meshes. *Journal of Aerospace Engineering*, 19(4):271–280, 2006.

[34] J. Peraire and P. O. Persson. The Compact Discontinuous Galerkin (CDG) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30(4):1806–1824, 2008.

[35] R. Codina, J. Principe, O. Guasch, and S. Badia. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196:2413–2430, 2007.

[36] Y. Maday and A. T. Patera. Spectral element methods for the incompressible Navier-Stokes equations. In *State-of-the-art surveys on Computational Mechanics (A90-47176 21-64). New York, American Society of Mechanical Engineers, 1989, p. 71-143. Research supported by DARPA.*, volume 1, pages 71–143, 1989.

[37] S. McKee, M. Tomé, V. Ferreira, J. Cuminato, A. Castelo, F. Sousa, and N. Mangiavacchi. The MAC methods. *Computers & Fluids*, 37(8):907–930, 2008.

[38] S. Balachandar and J. K. Eaton. Turbulent dispersed multiphase flow. *Annual Review of Fluid Mechanics*, 42:111–133, 2010.

[39] J. Brackbill, D. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.

[40] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, 113(1):134–147, 1994.

[41] M. Meier, G. Yadigaroglu, and B. L. Smith. A novel technique for including surface tension in PLIC-VOF methods. *European Journal of Mechanics-B/Fluids*, 21(1):61–73, 2002.

[42] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proceedings of the National Academy of Sciences*, pages 1591–1595, 1995.

[43] D. Jacqmin. Calculation of two-phase Navier-Stokes flows using phase-field modeling. *Journal of Computational Physics*, 155(1):96–127, 1999.

[44] N. Takashi. ALE finite element computations of fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 112(1-4):291–308, 1994.

[45] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, 1992.

[46] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, 30(6):775–793, 1999.

[47] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169(2):708–759, 2001.

[48] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83–116, 2002.

[49] Z. Li, F. A. Jaberi, and T. I. Shih. A hybrid Lagrangian Eulerian particle level set method for numerical simulations of two-fluid turbulent flows. *International Journal for Numerical Methods in Fluids*, 56(12):2271–2300, 2008.

[50] E. Aulisa, S. Manservisi, and R. Scardovelli. A mixed markers and Volume-Of-Fluid method for the reconstruction and advection of interfaces in two-phase and free-boundary flows. *Journal of Computational Physics*, 188:611–639, July 2003.

[51] Y. Wang, S. Simakhina, and M. Sussman. A hybrid level set-volume constraint method for incompressible two-phase flow. *Journal of Computational Physics*, 231(19):6438–6471, 2012.

[52] S. E. Hieber and P. Koumoutsakos. A Lagrangian particle level set methods. *Journal of Computational Physics*, 210(1):342–367, 2005.

[53] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics*, 130(2):269–282, #jan# 1997.

[54] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Transactions. Graphics.*, 21(3):736–744, #jul# 2002.

[55] P. Mullen, A. McKenzie, Y. Tong, and M. Desbrun. A variational approach to Eulerian geometry processing. *ACM Transactions. Graphics.*, 26(3), #jul# 2007.

[56] C. Wood, A. J. Gil, O. Hassan, and J. Bonet. Partitioned block-Gauss–Seidel coupling for dynamic fluid–structure interactions. *Computers and Structures*, 88(23):1367–1382, 2010.

[57] C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[58] C. S. Peskin and D. M. McQueen. Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 37(1):113–132, 1980.

[59] C. S. Peskin and D. M. McQueen. A three-dimensional computational method for blood flow in the heart I. immersed elastic fibers in a viscous incompressible fluids. *Journal of Computational Physics*, 81(2):372–405, 1989.

[60] D. M. McQueen and C. S. Peskin. A three-dimensional computational method for blood flow in the heart. II. contractile fibers. *Journal of Computational Physics*, 82(2):289–297, 1989.

[61] C. S. Peskin. The immersed boundary method. *Acta numerica*, 11:479–517, 2002.

[62] D. Devendran and C. S. Peskin. An Immersed Boundary energy-based method for incompressible viscoelasticity. *Journal of Computational Physics*, 231(14):4613–4642, 2012.

[63] Y. Kim and C. S. Peskin. Numerical study of incompressible fluid dynamics with nonuniform density by the Immersed Boundary Method. *Physics of Fluids*, 20(6):062101, 2008.

[64] Y. Kim, M. C. Lai, and C. S. Peskin. Numerical simulations of two-dimensional foam by the immersed boundary method. *Journal of Computational Physics*, 229(13):5194–5207, 2010.

[65] B. E. Griffith, R. D. Hornung, D. M. McQueen, and C. S. Peskin. An adaptive, formally second order accurate version of the Immersed Boundary Methods. *Journal of Computational Physics*, 223(1):10–49, 2007.

[66] C. Hesch, A. J. Gil, A. Arranz Carreño, J. Bonet, and P. Betsch. A mortar approach for Fluid-Structure interaction problems: Immersed strategies for deformable and rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 278:853–882, 2014.

[67] L. Yang, A. Arranz Carreño, A. J. Gil, and J. Bonet. An Immersed Structural Potential Method framework for incompressible flexible/rigid/multi-phase flow interactions. In *Proc. 22nd UK Conference of the Association for Computational Mechanics in Engineering (Exeter, UK, April 2014)*, 2014.

[68] M. H. Chung. Cartesian cut cell approach for simulating incompressible flows with rigid bodies of arbitrary shape. *Computers & Fluids*, 35(6):607–623, 2006.

[69] N. A. Patankar, P. Singh, D. D. Joseph, R. Glowinski, and T. W. Pan. A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows. *International Journal of Multiphase Flow*, 26(9):1509–1524, 2000.

[70] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Priaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flows. *Journal of Computational Physics*, 169(2):363–426, 2001.

[71] N. A. Patankar. Physical interpretation and mathematical properties of the stress-DLM formulation for rigid particulate flows. *International Journal for Computational Methods in Engineering Science and Mechanics*, 6(2):137–143, 2005.

[72] N. Sharma and N. A. Patankar. A fast computation technique for the direct numerical simulation of rigid particulate flows. *Journal of Computational Physics*, 205(2):439–457, 2005.

[73] M. Carlson, P. J. Mucha, and G. Turk. Rigid fluid: animating the interplay between rigid bodies and fluid. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 377–384. ACM, 2004.

[74] N. Zhang and Z. C. Zheng. An improved direct-forcing Immersed Boundary Method for finite difference applications. *Journal of Computational Physics*, 221(1):250–268, 2007.

[75] Z. Yu and X. Shao. A direct-forcing fictitious domain method for particulate flows. *Journal of Computational Physics*, 227(1):292–314, 2007.

[76] S. V. Apte, M. Martin, and N. A. Patankar. A numerical method for fully resolved simulation (FRS) of rigid particle–flow interactions in complex flows. *Journal of Computational Physics*, 228(8):2712–2738, 2009.

[77] A. A. Shirgaonkar, M. A. MacIver, and N. A. Patankar. A new mathematical formulation and fast algorithm for fully resolved simulation of self-propulsion. *Journal of Computational Physics*, 228(7):2366–2390, 2009.

[78] J. Mohd-Yusof. Combined Immersed-Boundary/B-spline methods for simulations of ow in complex geometries. *Annual Research Briefs. NASA Ames Research Center Stanford University Center of Turbulence Research: Stanford*, pages 317–327, 1997.

[79] E. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined Immersed-Boundary Finite-Difference Methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161(1):35–60, 2000.

[80] R. Glowinski, T. W. Pan, T. Hesla, and D. Joseph. A distributed Lagrange multiplier/Fictitious Domain Method for particulate flows. *International Journal of Multiphase Flow*, 25(5):755–794, 1999.

[81] K.-J. Bathe. The inf–sup condition and its evaluation for mixed finite element methods. *Computers & structures*, 79(2):243–252, 2001.

[82] R. J. LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[83] S. Badia, A. Quaini, and A. Quarteroni. Splitting methods based on algebraic factorization for fluid-structure interaction. *SIAM Journal on Scientific Computing*, 30(4):1778–1805, 2008.

[84] S. Badia, F. Nobile, and C. Vergara. Robin–Robin preconditioned Krylov methods for Fluid–Structure Interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 198(33):2768–2784, 2009.

[85] S. Badia, F. Nobile, and C. Vergara. Fluid–structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227(14):7027–7051, 2008.

[86] J. D. Anderson and J. Wendt. *Computational fluid dynamics*, volume 206. Springer, 1995.

[87] C. Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*, volume 1. Butterworth-Heinemann, 2007.

[88] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct numerical simulations of gas-liquid multiphase flows*. Cambridge University Press, 2011.

[89] R. Bracewell. The Fourier Transform and IIS Applications. *New York*, 1965.

[90] N. Patankar. A formulation for fast computations of rigid particulate flows. *Center for Turbulence Research Annual Research Briefs*, 2001:185–196, 2001.

[91] C. Bajaj. *Introduction to implicit surfaces*. Morgan Kaufmann, 1997.

[92] E. Dimas and D. Briassoulis. 3D geometric modelling based on NURBS: a review. *Advances in Engineering Software*, 30(9-11):741–751, 1999.

[93] G. Turk and J. F. O'brien. Variational implicit surfaces. *Techical Reports GIT-GVU-99-15*, 1999.

[94] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[95] W. F. Noh and P. Woodward. SLIC (simple line interface calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340. Springer, 1976.

[96] A. Iafrati, A. Di Mascio, and E. Campana. A level set technique applied to unsteady free surface flows. *International journal for numerical methods in fluids*, 35(3):281–297, 2001.

[97] S. Pillapakkam and P. Singh. A level-set method for computing solutions to viscoelastic two-phase flow. *Journal of Computational Physics*, 174(2):552–578, 2001.

[98] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. Multiple interacting liquids. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 812–819. ACM, 2006.

[99] D. P. Starinshak, S. Karni, and P. L. Roe. A new Level Set model for multimaterial flows. *Journal of Computational Physics*, 262(0):1–16, 2014.

[100] B. P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19(1):59–98, 1979.

[101] R. Temam. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires (ii). *Archive for Rational Mechanics and Analysis*, 33(5):377–385, 1969.

[102] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965.

[103] P. Gaskell and A. Lau. Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithms. *International Journal for Numerical Methods in Fluids*, 8(6):617–641, 1988.

[104] J. Zhu. A low-diffusive and oscillation-free convection scheme. *Communications in applied numerical methods*, 7(3):225–232, 1991.

[105] R. Codina and J. Blasco. Stabilized finite element method for the transient Navier–Stokes equations based on a pressure gradient projection. *Computer Methods in Applied Mechanics and Engineering*, 182(3):277–300, 2000.

[106] R. D. Falgout and U. M. Yang. hypre: A library of high performance preconditioners. In *Computational Science-ICCS 2002*, pages 632–641. Springer, 2002.

[107] C. Van Loan. *Computational frameworks for the fast Fourier transform*, volume 10. Siam, 1992.

[108] D. Hartmann, M. Meinke, and W. Schröder. Differential equation based constrained reinitialization for Level Set methods. *Journal of Computational Physics*, 227(14):6821–6845, 2008.

[109] A. Chéné, C. Min, and F. Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *Journal of Scientific Computing*, 35(2-3):114–131, #jun# 2008.

[110] C. Min. On reinitializing Level Set functions. *Journal of Computational Physics*, 229(8):2764–2772, 2010.

[111] R. Giovanni and P. Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163:51–67, September 2000.

[112] D. Adalsteinsson and J. A. Sethian. A fast Level Set Method for propagating interfaces. *Journal of Computational Physics*, 118(2):269–277, 1995.

[113] M. Sussman and M. Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM Journal on Scientific Computing*, 31(4):2447–2471, 2009.

[114] M. Sussman. A method for overcoming the surface tension time step constraint in multiphase flows II. *International Journal for Numerical Methods in Fluids*, 68(11):1343–1361, 2012.

[115] N. Thürey, C. Wojtan, M. Gross, and G. Turk. A multiscale approach to mesh-based surface tension flows. *ACM Transactions. Graphics.*, 29(4):48:1–48:10, #jul# 2010.

[116] Z. Li and M. C. Lai. The immersed interface method for the Navier-Stokes equations with singular forces. *Journal of Computational Physics*, 171(2):822–842, 2001.

[117] X. Wang and W. K. Liu. Extended Immersed Boundary Method using FEM and RKPM. *Computer Methods in Applied Mechanics and Engineering*, 193(12):1305–1321, 2004.

[118] O. Hassan, E. Probert, K. Morgan, and N. Weatherill. Unsteady flow simulation using unstructured meshes. *Computer Methods in Applied Mechanics and Engineering*, 189(4):1247–1275, 2000. ¡ce:title¿Adaptive Methods for Compressible CFD¡/ce:title¿.

[119] D. Z. Zhang, X. Ma, and P. T. Giguere. Material point method enhanced by modified gradient of shape function. *Journal of Computational Physics*, 230(16):6379–6398, 2011.

[120] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed Finite Element Methods. *Computer Methods in Applied Mechanics and Engineering*, 193(21):2051–2067, 2004.

[121] S. Bardenhagen, J. Brackbill, and D. Sulsky. The material-point method for granular materials. *Computer Methods in Applied Mechanics and Engineering*, 187(3):529–541, 2000.

[122] A. M. Roma, C. S. Peskin, and M. J. Berger. An adaptive version of the immersed boundary method. *Journal of Computational Physics*, 153(2):509–534, 1999.

[123] A. K. Tornberg and B. Engquist. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics*, 200(2):462–488, #nov# 2004.

[124] A. G. Malan, R. W. Lewis, and P. Nithiarasu. An improved unsteady, unstructured, artificial compressibility, finite volume scheme for viscous incompressible flows: Part II. Application. *International Journal for Numerical Methods in Engineering*, 54(5):715–729, 2002.

[125] J. C. Martin and W. J. Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 244(1):312–324, 1952.

[126] C. Hu and M. Sueyoshi. Numerical simulation and experiment on dam break problem. *Journal of Marine Science and Application*, 9(2):109–114, 2010.

[127] F. Kelecy and R. Pletcher. The development of a free surface capturing approach for multidimensional free surface flows in closed containers. *Journal of Computational Physics*, 138(2):939–980, 1997.

[128] J. Xia, B. Lin, R. A. Falconer, and G. Wang. Modelling dam-break flows over mobile beds using a 2D coupled approach. *Advances in Water Resources*, 33(2):171–183, 2010.

[129] T. W. H. Sheu, C. H. Yu, and P. H. Chiu. Development of a dispersively accurate conservative level set scheme for capturing interface in two-phase flows. *Journal of Computational Physics*, 228(3):661–686, #feb# 2009.

[130] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids*, 60(11):1259–1288, 2009.

[131] S. Turek and C. Becker. FEATFLOW Finite element software for the incompressible Navier-Stokes equations, User Manual Release 1.1. *Preprint, Heidelberg*, 4, 1998.

[132] V. John and G. Matthies. MooNMD-a program package based on mapped finite element methods. *Computing and Visualization in Science*, 6(2-3):163–170, 2004.

[133] A. Comsol. COMSOL multiphysics users guide. *Version: September*, 2005.

[134] A. Fluent. 12.0 Theory Guide. *Ansys Inc*, 6, 2009.

[135] I. Janosi, D. Jan, K. Szabo, and T. Tel. Turbulent drag reduction in dam-break flows. *Experiments in Fluids*, 37(2):219–229, 2004.

[136] A. Khayyer and H. Gotoh. On particle-based simulation of a dam break over a wet bed. *Journal of Hydraulic Research*, 48(2):238–249, 2010.

[137] M. Gomez-Gesteira, B. D. Rogers, R. A. Dalrymple, and A. J. Crespo. State-of-the-art of classical SPH for free-surface flows. *Journal of Hydraulic Research*, 48(S1):6–27, 2010.

[138] A. Crespo, M. Gómez-Gesteira, and R. Dalrymple. Modeling dam break behavior over a wet bed by a SPH technique. *Journal of Waterway, Port, Coastal, and Ocean Engineering*, 134(6):313–320, 2008.

[139] F. Blanchette. Simulation of mixing within drops due to surface tension variations. *Physical Review Letters*, 105:074501, 2010.

[140] C. Josserand and S. Zaleski. Droplet splashing on a thin liquid film. *Physics of Fluids*, 15(6):1650–1657, 2003.

[141] B. S. Dooley, A. E. Warncke, M. Gharib, and G. Tryggvason. Vortex ring generation due to the coalescence of a water drop at a free surface. *Experiments in Fluids*, 22(5):369–374, 1997.

[142] J. Eggers and E. Villermaux. Physics of liquid jets. *Reports on Progress in Physics*, 71(3):036601, 2008.

[143] L. Xu, W. W. Zhang, and S. R. Nagel. Drop splashing on a dry smooth surface. *Physical Review Letters*, 94:184505, May 2005.

[144] A. Yarin. Drop impact dynamics: Splashing, spreading, receding, bouncing. *Annual Review of Fluid Mechanics*, 38(1):159–192, 2006.

[145] R. Wal, G. Berger, and S. Mozes. The splash/non-splash boundary upon a dry surface and thin fluid film. *Experiments in Fluids*, 40(1):53–59, 2006.

[146] G. Agbaglah, S. Delaux, D. Fuster, J. Hoepffner, C. Josserand, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski. Parallel simulation of multiphase flows using octree adaptivity and the Volume-Of-Fluid methods. *Comptes Rendus Mécanique*, 339(2-3):194–207, 2011.

[147] C. Kassiotis, A. Ibrahimbegovic, and H. Matthies. Partitioned solution to Fluid-Structure interaction problem in application to free-surface flows. *European Journal of Mechanics-B/Fluids*, 29(6):510–521, 2010.

[148] S. Idelsohn, J. Marti, A. Limache, and E. Oñate. Unified Lagrangian formulation for elastic solids and incompressible fluids: Application to fluid structure interaction problems via the PFEM. *Computer Methods in Applied Mechanics and Engineering*, 197(19 20):1762–1776, 2008.

[149] A. Rafiee and K. P. Thiagarajan. An SPH projection method for simulating fluid-hypoelastic structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2785–2795, 2009.

[150] C. Antoci, M. Gallati, and S. Sibilla. Numerical simulation of fluid-structure interaction by SPH. *Computers and Structures*, 85(11-14):879–890, 2007. Fourth MIT Conference on Computational Fluid and Solid Mechanics.

[151] K. Kleefsman, G. Fekken, A. Veldman, B. Iwanowski, and B. Buchner. A Volume-Of-Fluid based simulation method for wave impact problems. *Journal of Computational Physics*, 206(1):363–393, 2005.

[152] P. Lin. A fixed-grid model for simulation of a moving body in free surface flows. *Computers & Fluids*, 36(3):549–561, 2007.

[153] J. Yang and F. Stern. Sharp interface Immersed-Boundary Level-Set Method for wave body interactions. *Journal of Computational Physics*, 228(17):6590–6616, 2009.

[154] J. Happel and H. Brenner. *Low Reynolds number hydrodynamics: with special applications to particulate media*, volume 1. Springer Science & Business Media, 1983.

[155] A. Staniforth and J. Côté. Semi-Lagrangian integration schemes for atmospheric models-a review. *Monthly weather review*, 119(9):2206–2223, 1991.

[156] J. Donea and A. Huerta. *Finite Element Methods for Flow Problems*. Finite Element Methods for Flow Problems. John Wiley & Sons, 2003.

[157] F. Dupont and Y. Liu. Back and Forth Error Compensation and Correction methods for removing errors induced by uneven gradients of the level set functions. *Journal of Computational Physics*, 190:311–324, September 2003.

[158] V. Mihalef, D. Metaxas, and M. Sussman. Textured liquids based on the Marker Level set. *Computer Graphics Forum*, 26(3):457–466, 2007.