National Technical University of Athens
Department of Naval Architecture and Marine Engineering
Laboratory for Ship and Marine Hydrodynamics

Software Development with the Method of Cut Cells and its Application in Computational Fluid Dynamics Problems.

Diploma Thesis

Leonidas Angeletos

Thesis Supervisor: Assistant Professor George Papadakis

Athens,2020

Software Development with the Method of Cut Cells and its Application in Computational Fluid Dynamics Problems.

Diploma Thesis

Leonidas Angeletos

Thesis Supervisor: Assistant Professor George Papadakis

Athens,2020

Abstract

The purpose of this dissertation was to develop a software that will implement the intersecting cell method, a subcategory of the Immersed boundary methods.

The first part of the diploma thesis deals with the general characteristics of computational fluid mechanics, as well as the mathematical background that accompanies it.

Reference is then made to the MaPFlow solver, which is a CFD software used to solve the developing flows. The latter is a compressible solver where the equations are discretized using the finite volume method. To solve the Navier Stokes equations of incompressible and unsteady flows as in the applications we realized in this work, the method of Artificial Compressibility is used, which will be analyzed in detail.

Finally, the applications in which the Cut-Cell method was tested involved one-phase viscous flows around a cylinder, as well as two-phase inviscid flows with wave propagation over a variable bottom. Both applications were on a two-dimensional level.

Ανάπτυξη λογισμικού με τη μέθοδο τεμνόμενων κελιών και εφαρμογή του σε προβλήματα υπολογιστικής ρευστομηχανικής.

Διπλωματική εργασία.

Λεωνίδας Αγγελέτος

Επιβλέπων: Επίκουρος Καθηγητής Γεώργιος Παπαδάκης

Αθήνα,2020

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας ήταν η ανάπτυξη ενός λογισμικού το οποίο θα υλοποιεί τη μέθοδο των τεμνόμενων κελιών, μια υποκατηγορία των μεθόδων των μη οριοδετών ή εμβαπτιζόμενων πλεγμάτων.

Το πρώτο μέρος της εργασίας ασχολείται με τα γενικά χαρακτηριστικά της υπολογιστικής ρευστμηχανικής, καθώς και το μαθηματικό υπόβαθρο που τη συνδεύει.

Στη συνέχεια γίνεται αναφορά στον επιλυτή MaPFlow, ο οποίος αποτελεί ένα λογισμικό υπολογιστικής ρευστομηχανικής που χρησιμοποιήθηκε για την επίλυση των αναπτυσσόμενων ροών. Ο τελευταίος είναι ένας συμπιεστός κώδικας όπου η διακριτοποίηση των εξισώσεων γίνεται με τη μέθοδο των πεπερασμένων όγκων. Για την επίλυση των εξισώσεων Navier Stokes ασυμπίεστων και χρονικά μη-μόνιμων ροών όπως είναι στις εφαρμογές που πραγματοποιήσαμε σε αυτή την εργασία , χρησιμοποιείται η μέθοδος της τεχνητής συμπιεστότητας, που θα αναλυθεί διεξοδικά.

Τέλος, οι εφαρμογές στις οποίες δοκιμάστηκε η μέθοδος των τεμνόμενων κελιών αφορούσαν μοναφασικές συνεκτικές ροές γύρω από κύλινδρο, καθώς και διφασικές μη συνεκτικές ροές με διάδοση κυματισμού πάνω από μεταβαλλόμενο πυθμένα. Και οι δύο εφαρμγές ήταν σε δισδιάστατο επίπεδο.

Acknoweledgements

At this point I would like to thank with all my sincerity my supervising Professor Mr. George Papadakis, for the knowledge he generously imparted to me, as well as for the excellent relationship that we developed throughout our cooperation.

I would also like to give equal and special thanks to the PhD candidate Dimitrios Ntouras for his invaluable contribution and assistance in the realization of this dissertation.

Finally, I want to thank the people who accompany me in my life for the support and love they offer me.

Acronyms

AC	Artifial Compressibility
CFD	Computational Fluid Dynamics
IB	Immersed Boundary
IBM	Immersed Boundary Methods
NS	Navier-Stokes Equations
$NTUA$	National Technical University of Athens
$PDEs$	Partial Differential Equations
VOF	Volume Of Fluid

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Computational Fluid Dynamics

Computational fluid dynamics or CFD is the analysis of systems involving fluid flow, heat transfer and associated phenomena such as chemical reactions by means of computer-based simulation. The technique is very powerful and spans a wide range of industrial and non-industrial application areas [1]. Some examples are:

- aerodynamics of aircraft and vehicles: lift and drag
- hydrodynamics of ships
- biomedical engineering: blood flows through arteries and veins
- meteorology: weather prediction

During the last decades the aerospace industry has integrated CFD techniques into the design, RD and manufacture of aircraft and jet engines. CFD is becoming a vital component in the design of industrial products and processes. The ultimate aim of developments in the CFD field is to provide a capability comparable with other CAE (computer-aided engineering) tools such as stress analysis codes. The main reason why CFD has lagged behind is the tremendous complexity of the underlying behaviour, which precludes a description of fluid flows that is at the same time economical and sufficiently complete. The availability of affordable high-performance computing hardware and the introduction of user-friendly interfaces have led to a recent upsurge of interest, and CFD has entered into the wider industrial community since the 1990s.

ADVANTAGES OF CFD OVER EXPERIMENTS

There are several unique advantages of CFD over experiment-based approaches to fluid systems design:

- substantial reduction of lead times and costs of new designs
- ability to study systems where controlled experiments are difficult or impossible to perform (e.g. very large systems)
- ability to study systems under hazardous conditions at and beyond their normal performance limits (e.g. safety studies and accident scenarios)
- practically unlimited level of detail of results

HOW DOES A CFD CODE WORK?

CFD codes are structured around the numerical algorithms that can tackle fluid flow problems. In order to provide easy access to their solving power all commercial CFD packages include sophisticated user interfaces to input problem parameters and to examine the results. Hence all codes contain three main elements: (i) a pre-processor, (ii) a solver and (iii) a post-processor.

Pre-Processor

Pre-processing consists of the input of a flow problem to a CFD program by means of an operator-friendly interface and the subsequent transformation of this input into a form suitable for use by the solver. The user activities at the pre-processing stage involve:

- Definition of the geometry of the region of interest: the computational domain
- Grid generation – the sub-division of the domain into a number of smaller, non-overlapping sub-domains: a grid (or mesh) of cells (or control volumes or elements)
- Selection of the physical and chemical phenomena that need to be modelled
- Definition of fluid properties
- Specification of appropriate boundary conditions at cells which coincide with or touch the domain boundary

The solution to a flow problem (velocity, pressure, temperature etc.) is defined at nodes inside each cell. The accuracy of a CFD solution is governed by the number of cells in the grid. In general, the larger the number of cells, the better the solution accuracy. Both the accuracy of a solution and its cost in terms of necessary computer hardware and calculation time are dependent on the fineness of the grid.

Solver

There are three distinct streams of numerical solution techniques: finite difference, finite element and spectral methods. We shall be solely concerned with the finite volume method.In outline the numerical algorithm consists of the following steps:

- Integration of the governing equations of fluid flow over all the (finite) control volumes of the domain
- Discretisation – conversion of the resulting integral equations into a system of algebraic equations
- Solution of the algebraic equations by an iterative method

Post-Processor

The leading CFD packages are equipped with versatile data visualisation tools. These include:

- Domain geometry and grid display
- Vector plots
- Line and shaded contour plots
- 2D and 3D surface plots
- Particle tracking
- View manipulation (translation, rotation, scaling etc.)
- Colour PostScript output

More recently these facilities may also include animation for dynamic result display, and in addition to graphics all codes produce trusty alphanumeric output and have data export facilities for further manipulation external to the code.

PROBLEM SOLVING WITH CFD

Prior to setting up and running a CFD simulation there is a stage of identification and formulation of the flow problem in terms of the physical and chemical phenomena that need to be considered. Typical decisions that might be needed are whether to model a problem in two or three dimensions, to exclude the effects of ambient temperature or pressure variations on the density of an air flow, to choose to solve the turbulent flow equations or to neglect the effects of small air bubbles dissolved in tap water. To make the right choices requires good modelling skills, because in all but the simplest problems we need to make assumptions to reduce the complexity to a manageable level whilst preserving the salient features of the problem at hand. It is the appropriateness of the simplifications introduced at this stage that at least partly governs the quality of the information generated by CFD, so the user must continually be aware of all the assumptions, clear-cut and tacit ones, that have been made.

Performing the computation itself requires operator skills of a different kind. Specification of the domain geometry and grid design are the main tasks at the input stage and subsequently the user needs to obtain a successful simulation result. The two aspects that characterise such a result are convergence and grid independence.

It is impossible to assess the validity of the models of physics and chemistry embedded in a program as complex as a CFD code or the accuracy of its final results by any means other than comparison with experimental test work. Anyone wishing to use CFD in a serious way must realise that it is no substitute for experimentation, but a very powerful additional problem solving tool.

Chapter 2

Fundamentals of Fluid Flows

2.1 Governing equations of fluid flows in Cartesian co-ordinates

The governing equations of fluid flow represent mathematical statements of the **conservation laws of physics**:

- The mass of a fluid is conserved
- The rate of change of momentum equals the sum of the forces on a fluid particle (Newton's second law)
- The rate of change of energy is equal to the sum of the rate of heat addition to and the rate of work done on a fluid particle (first law of thermodynamics)

The fluid will be regarded as a continuum. For the analysis of fluid flows at macroscopic length scales the molecular structure of matter and molecular motions may be ignored.

2.1.1 Mass conservation or continuity equation

Firstly, on the mass conservation law a mass balance for the fluid element must be written. So, the rate of increase of mass in fluid element is equal to the net rate of flow of mass into fluid element. Hence, the unsteady, three-dimensional mass conservation or continuity equation in a compressible fluid can be written as 2.1:

$$\frac{\partial \rho}{\partial t} + div(\rho \mathbf{U}) = 0 \tag{2.1}$$

In equation (2.1) ρ is the density and U is the velocity vector. The first term on the left-hand side is the rate of change in time of the density (mass per unit volume) and the second term describes the net flow of mass out of the element across its boundaries and is called the convective term. **For the majority of marine CFD**

simulations, the flow is considered to be incompressible (the density ρ is constant) and so equation 2.1 becomes:

$$div\mathbf{U} = 0 \Rightarrow \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{2.2}$$

2.1.2 Momentum equation

Newton's second law states that the rate of change of momentum of a fluid particle equals the sum of the forces on the particle. The rates of increase of x-, y-, z-momentum per unit volume of a fluid particle are given by equation 2.3:

$$\rho\frac{Du}{Dt} \quad \rho\frac{Dv}{Dt} \quad \rho\frac{Dw}{Dt} \tag{2.3}$$

We distinguish two types of forces on fluid particles:

- Surface forces (pressure, gravity, viscous)
- Body forces (centrifugal, Coriolis, electromagnetic)

It is common practice to highlight the contributions due to the surface forces as separate terms in the momentum equation and to include the effects of body forces as source terms. The pressure, a normal stress, is denoted by p. Viscous stresses are denoted by τ.

The x-component of the momentum equation is found by setting the rate of change of x-momentum of the fluid particle 2.3 equal to the total force in the x-direction on the element due to surface stresses plus the rate of increase of x-momentum due to sources:

$$\rho\frac{Du}{Dt} = \frac{\partial(-p + \tau_{xx})}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + S_{Mx}$$

In the same way,the y-component of the momentum equation is given by:

$$\rho\frac{Dv}{Dt} = \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial(-p + \tau_{yy})}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + S_{My} \tag{2.4}$$

and the z-component by:

$$\rho\frac{Dw}{Dt} = \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial(-p + \tau_{zz})}{\partial z} + S_{Mz}$$

2.1.3 Energy equation

The energy equation is derived from the first law of thermodynamics, which states that the rate of change of energy of a fluid particle is equal to the rate of heat addition to the fluid particle plus the rate of work done on the particle.

As before, we will be deriving an equation for the rate of increase of energy of a fluid particle per unit volume, which is given by:

$$\rho\frac{DE}{Dt} \tag{2.5}$$

The energy equation is:

$$\begin{aligned}\rho\frac{DE}{Dt} = -div(pu) + [\frac{\partial(u\tau_{xx})}{\partial x} + \frac{\partial(u\tau_{yx})}{\partial y} + \frac{\partial(u\tau_{zx})}{\partial z} + \frac{\partial(\upsilon\tau_{xy})}{\partial x} + \frac{\partial(\upsilon\tau_{yy})}{\partial y} + \\ \frac{\partial(\upsilon\tau_{zy})}{\partial z} + \frac{\partial(w\tau_{xz})}{\partial x} + \frac{\partial(w\tau_{yz})}{\partial y} + \frac{\partial(w\tau_{zz})}{\partial z}] + div(kgradT) + S_E\end{aligned} \tag{2.6}$$

2.1.4 Equations of state

The motion of a fluid in three dimensions is described by a system of five partial differential equations: mass conservation (2.1), x-, y- and z-momentum equations (2.4) and energy equation (2.6). Among the unknowns are four thermodynamic variables: ρ, p, i and T. Relationships between the thermodynamic variables can be obtained through the assumption of **thermodynamic equilibrium**.

Equations of state relate the other variables to the two state variables. If we use ρ and T as state variables we have state equations for pressure p and specific internal energy i:

$$p = p(\rho, T) \quad and \quad i = i(\rho, T) \tag{2.7}$$

For a perfect gas the following, well-known, equations of state are useful:

$$p = \rho RT \quad and \quad i = C_v T \tag{2.8}$$

In the flow of compressible fluids the equations of state provide the linkage between the energy equation on the one hand and mass conservation and momentum equations on the other. This linkage arises through the possibility of density variations as a result of pressure and temperature variations in the flow field.

Liquids and gases flowing at low speeds behave as incompressible fluids. Without density variations there is no linkage between the energy equation and the mass conservation and momentum equations. The flow field can often be solved by considering mass conservation and momentum equations only. The energy equation only needs to be solved alongside the others if the problem involves heat transfer.

2.1.5 Navier-Stokes equations

The governing equations contain as further unknowns the viscous stress components τ_{ij}. In many fluid flows the viscous stresses can be expressed as functions of the local deformation rate or strain rate. All gases and many liquids are isotropic. In this work we consider that **all fluids are isotropic**.

In a Newtonian fluid the viscous stresses are proportional to the rates of deformation. The three-dimensional form of Newton's law of viscosity for compressible flows involves two constants of proportionality: the first (dynamic) viscosity, μ, to relate stresses to linear deformations, and the second viscosity, λ, to relate stresses to the volumetric deformation. The nine viscous stress components, of which six are independent, are:

$$\begin{aligned}
\tau_{xx} &= 2\mu\frac{\partial u}{\partial x} + \lambda divU \\
\tau_{yy} &= 2\mu\frac{\partial v}{\partial y} + \lambda divU \\
\tau_{yy} &= 2\mu\frac{\partial w}{\partial z} + \lambda divU \\
\tau_{xy} = \tau_{yx} &= \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}) \\
\tau_{xz} = \tau_{zx} &= \mu(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}) \\
\tau_{yz} = \tau_{zy} &= \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)
\end{aligned} \tag{2.9}$$

Not much is known about the λ-viscosity, because its effect is small in practice. For gases a good working approximation can be obtained by taking the value λ=-2/3μ (Schlichting 1979) [2]. It should be noted that the expressions for the normal stress simplify for incompressible fluid (constant density) ,as in this study, since the divergence of velocity is zero (div U=0). Substitution of the above shear stresses (2.9) into (2.4) yields the so-called Navier–Stokes equations. After some rearrangements the NS or x-,y-,z- momentum equations can be written in the most useful form for the development of the finite volume method:

$$\begin{aligned}
\frac{\partial(\rho u)}{\partial t} + div(\rho u\mathbf{u}) &= -\frac{\partial p}{\partial x} + div(\mu\, grad\, u) + S_{Mx} \\
\frac{\partial(\rho v)}{\partial t} + div(\rho v\mathbf{u}) &= -\frac{\partial p}{\partial y} + div(\mu\, grad\, v) + S_{My} \\
\frac{\partial(\rho w)}{\partial t} + div(\rho w\mathbf{u}) &= -\frac{\partial p}{\partial z} + div(\mu\, grad\, w) + S_{Mz}
\end{aligned} \tag{2.10}$$

2.1.6 General transport equations

As we observe, there are significant commonalities betweeen the five PDEs for mass, momentum and energy conservation. If we introduce a general variable φ the conservative form of all fluid flow equations can usefully be written in the following form:

$$\frac{\partial(\rho\phi)}{\partial t} + div(\rho\phi\mathbf{u}) = div(\Gamma\, grad\ \phi) + S_{\Phi} \tag{2.11}$$

Equation (2.11) is the so-called transport equation for property φ. It clearly highlights the various transport processes: the rate of change term and the convective term on the left hand side and the diffusive term (Γ = diffusion coefficient) and the source term respectively on the right hand side.

The key step of the finite volume method is the integration of (2.11) over a three-dimensional control volume (CV):

$$\int_{CV} \frac{\partial(\rho\phi)}{\partial t} dV + \int_{CV} div(\rho\phi\mathbf{u}) dV = \int_{CV} div(\Gamma\, grad\ \phi) dV + \int_{CV} S_{\Phi} dV \tag{2.12}$$

The volume integrals in the second term on the left hand side, the convective term, and in the first term on the right hand side, the diffusive term, are rewritten as integrals over the entire bounding surface of the control volume by using Gauss's divergence theorem. For a vector **a** this theorem states:

$$\int_{CV} div(\mathbf{a}) dV = \int_{A} \mathbf{n}\ .\ \mathbf{a} dA \tag{2.13}$$

The physical interpretation of **n.a** is the component of vector **a** in the direction of the vector **n** normal to surface element dA. Thus the integral of the divergence of a vector **a** over a volume is equal to the component of **a** in the direction normal to the surface which bounds the volume summed (integrated) over the entire bounding surface A. Applying Gauss's divergence theorem, equation (2.10) can be written as follows:

$$\frac{\partial}{\partial t}(\int_{CV} \rho\phi dV) + \int_{A} \mathbf{n}.(\rho\phi\mathbf{u}) dA = \int_{A} \mathbf{n}.(\Gamma\, grad\ \phi) dA + \int_{CV} S_{\Phi} dV \tag{2.14}$$

The first term on the left hand side of (2.14) signifies the rate of change of the total amount of fluid property φ in the control volume. The second term on the left hand side of (2.14), the convective term, is the net rate of decrease of fluid property φ of the fluid element due to convection. The first term on the right hand side of (2.14), the diffusive term, is associated with a flux into the element and represents the net rate of increase of fluid property φ of the fluid element due to diffusion. The final term on the right hand side of this equation gives the rate of increase of property φ as a result of sources inside the fluid element.

In steady state problems the rate of change term of (2.14) is equal to zero. This leads to the integrated form of the steady transport equation:

$$\int_A \mathbf{n}.(\rho\phi\mathbf{u})dA = \int_A \mathbf{n}.(\Gamma\, grad\ \phi)dA + \int_{CV} S_\Phi dV \tag{2.15}$$

In time-dependent problems it is also necessary to integrate with respect to time t over a small interval Δt from, say, t until t + Δt. This yields the most general integrated form of the transport equation:

$$\begin{gathered}\int_{\Delta t} \frac{\partial}{\partial t}(\int_{CV} \rho\phi dV) + \int_{\Delta t}\int_A \mathbf{n}.(\rho\phi\mathbf{u})dA = \\ \int_{\Delta t}\int_A \mathbf{n}.(\Gamma\, grad\ \phi)dA + \int_{\Delta t}\int_{CV} S_\Phi dV\end{gathered} \tag{2.16}$$

2.2 Classification of physical behaviours

Expept from the conservation equations of fluid flows it is necessary to mention the initial and boundary conditions that are needed in combination with the equations to construct a well-posed mathematical model of a fluid flow. First we distinguish two principal categories of physical behaviour:

- Equilibrium problems
- Marching problems

The problems in the first category are steady state situations, e.g. the steady state distribution of temperature in a rod of solid material. These and many other steady state problems are governed by elliptic equations. The prototype elliptic equation is Laplace's equation. In two dimensions we have :

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0 \tag{2.17}$$

Transient heat transfer, all unsteady flows and wave phenomena are examples of problems in the second category, the marching or propagation problems. These problems are governed by parabolic or hyperbolic equations. Parabolic equations describe time-dependent problems, which involve significant amounts of diffusion. Examples are unsteady viscous flows and unsteady heat conduction. The prototype parabolic equation is the diffusion equation:

$$\frac{\partial\phi}{\partial t} = \alpha\frac{\partial^2\phi}{\partial x^2} \tag{2.18}$$

Hyperbolic equations dominate the analysis of vibration problems. In general they appear in time-dependent processes with negligible amounts of energy dissipation. The prototype hyperbolic equation is the wave equation :

$$\frac{\partial^2 \phi}{\partial t^2} = c^2 \frac{\partial^2 \phi}{\partial x^2} \tag{2.19}$$

Chapter 3

Solver MaPFlow

In this work, to solve the above equations is used **MaPFlow**, a software for calculating and analyzing multiphase flows. MaPFlow is an Eulerian CFD Solver which solves the unsteady compressible Reynolds-Averaged Navier Stokes equations on unstructured grids using the Finite Volume method. The solver is parallelized by using the MPI protocol, while the grid partitioning is performed by the Metis library. [3]

In this dissertation the applications and simulations that were carried out concerned incompressible flows. The mathematical techniques that can be uses so as to deal with this category of flows are the following:

- Vorticity-Stream function method
- Artificial compressibility method
- Projection Methods (Pressure-correction algorithms)

The MaPFlow Solver uses the Artificial Compressibility method for the solution of the incompressible flow. It is necessary to give some basic elements of this method.

3.1 Artificial Compressibility Method

The main difficulty with the solution of the incompressible flow equations is the decoupling of the continuity and momentum equations due to the absence of the pressure (or density) term from the former. Chorin (1967) proposed a special approach in order to overcome the difficulty of the pressure decoupling. This approach is called artificial compressibility (AC). [4]. Chorin was the first who introduced this term, but later the method was fully extended to the general three dimensions by Kwak et al. (1984) and Chang and Kwak (1984). To reflect the physical nature of the pressure projection in this method more accurately, a new term pseudo-compressibility was then introduced. The two terms, artificial compressibility, and pseudo-compressibility have been used interchangeably ever since.

3.1.1 Governing Equations

Constant density one phase incompressible flows are governed by the continuity and momentum equations.

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

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{3.2}$$

In the above equations, **u** is the velocity vector with components (u, v, w) for the three Cartesian directions (x, y, z), respectively, ρ is the fluid density, p is the pressure and ν denotes viscosity.

Chorin introduces a pseudo-time pressure derivative in the continuity equation (τ is the pseudo-time). The modified continuity equation is given below in differential form.

$$\frac{1}{\beta}\frac{\partial p}{\partial \tau} + \nabla \cdot \mathbf{u} = 0 \tag{3.3}$$

where β is the artificial compressibility parameter, analogous to a relaxation parameter, that needs to be properly chosen in order to achieve numerical convergence.

For steady state problems, eq. (3.3) is solved along with the momentum equations until the pseudo-time pressure derivative vanishes. The larger the value of β the more incompressible the equations are. The value of β determines the performance of the method, usually it is chosen between 0.1 and 10. [5]

The essence of the AC method is that it assumes a relation between pressure and density during convergence (Equation (3.4)). This relation is similar to compressible definition of sound speed. However, in this case the parameter β is a numerical parameter that regulates convergence.

$$\frac{\partial \rho}{\partial p} = \frac{1}{\beta} \tag{3.4}$$

Equation (3.4) gives rise to a pseudo-sound speed definition. In case of one-dimensional, one-phase flows, the artificial speed of sound is given by Equation (3.5). The sound speed is regulated through the artificial parameter β.

$$c = \sqrt{\beta + u^2} \tag{3.5}$$

In the present work, we combine the AC method with the Volume of Fluid (VOF) method because we have to deal with two phase unsteady flows of immiscible fluids. An additional equation is added to the system of equations expressing the general fluid properties. [6]

The VOF method uses an indicator function to describe the presence of the liquid or the gas phase. Then the volume fraction is defined as:

$$\alpha_l = \frac{\rho_m - \rho_\alpha}{\rho_w - \rho_\alpha} \tag{3.6}$$

where, let ρ_w be the water density and ρ_α the density of the air. The volume fraction is equal to 0 in regions occupied by the air phase, equals to 1 when only the water phase is present and takes values between 0 and 1 near the free surface.

The interface of the fluids is located in regions of rapid change of the gradient of the indicator function. The properties of the mixture fluid, density (ρ_m) and dynamic viscosity (μ_m), are described by the blending functions.

$$\begin{aligned} \rho_m &= \alpha_l \rho_w + (1 - \alpha_l)\rho_\alpha \\ \mu_m &= \mu_l \rho_w + (1 - \alpha_l)\mu_\alpha \end{aligned} \tag{3.7}$$

The free surface, as presented in Eq. (3.6), is considered to be a material surface.

$$\frac{\partial \alpha_l}{\partial t} + \mathbf{u} \cdot \nabla \alpha_l = 0 \tag{3.8}$$

where $\mathbf{u}$ = (u, v, w) is the three-dimensional velocity vector, $\nabla = (\partial_x, \partial_y, \partial_z)$ is the divergence operator and t denotes the real time variable.

Regarding the simulation of two-phase flows, the momentum equations written in a conservative form and expressed in terms of the mixture quantities are presented in Equation (3.9), in combination with the VOF equation.

$$\begin{aligned} \frac{\partial \rho_m \mathbf{u}}{\partial t} + \nabla \cdot (\rho_m \mathbf{u} \cdot \mathbf{u}) + \nabla p = \nabla \bar{\sigma} + \overrightarrow{F}_\beta \\ \frac{\partial \alpha_l}{\partial t} + \mathbf{u} \cdot \nabla \alpha_l = 0 \end{aligned} \tag{3.9}$$

where, $\bar{\sigma}$ is the stress tensor and the vector $\overrightarrow{F}_\beta$ includes source terms and body forces.

The Equations (3.3) and (3.9) constitute a fully coupled system of equations capable of describing two-phase flows. The formed system has a hyperbolic nature and consequently numerical techniques used for such solvers can be used. By introducing the AC method the system of equation becomes hyperbolic in pseudo-time and consequently numerical techniques used for such solvers can be used. The convergent solution should satisfy the original sets of equations, by eliminating the pseudo-time derivatives. Nevertheless, the system in its original form poses several difficulties. Firstly, the density appears in the eigenvalues of the system yielding it stiff for high density ratios and secondly, the system cannot be written in a conservative form. To mitigate these perplexities, the preconditioner of Kunz is employed.

By introducing, the fictitious time derivative, for the momentum equations, the artificial compressibility method can be used in time-accurate computations. Indeed, employing the dual-time stepping technique each unsteady timestep is treated as a steady state problem. Furthermore, to express the governing equations as a single coupled system of equations the time derivatives (real and fictitious) of the momentum are expressed as a sum of time derivatives for velocity and density. The governing equation can be written in the following integrated form.

$$\Gamma \frac{\partial}{\partial \tau} \int_{D_i} \vec{Q} dD + \Gamma_e \frac{\partial}{\partial \tau} \int_{D_i} \vec{Q} dD + \int_{\partial D_i} (\vec{F_c} - \vec{F_u}) dS = \int_{D_i} \vec{S_q} dD \tag{3.10}$$

The system of Equation (3.10) is a fully coupled system of equations. These equations express the governing equations with respect to the primitive variables $\vec{Q}$. In order to cast the system in conservative form, the transformation matrix Γ_e is used. In Equation (3.11) the conservative variables are given by the vector $\vec{U}$. The three-dimension vector of velocity is denoted with $\vec{u}$, while p is the pressure.

$$\vec{U} = \begin{bmatrix} 0 & \rho\vec{u} & \alpha_l \end{bmatrix}^T \quad \vec{Q} = \begin{bmatrix} p & \vec{u} & \alpha_l \end{bmatrix}^T \tag{3.11}$$

$$\frac{\partial \vec{U}}{\partial t} = \Gamma_e \frac{\partial \vec{Q}}{\partial t}$$

The Jacobian matrix Γ_e and the precondition matrix of Kunz Γ are given by (3.12), where Δ_ρ is the density difference between the heavier and the lighter fluid and I_3x3 is the 3 by 3 identity matrix.

$$\Gamma = \begin{bmatrix} \frac{1}{\beta\rho_m} & 0 & 0 \\ 0 & \rho_m I_{3x3} & \vec{u}\Delta\rho \\ \frac{\alpha_l}{\beta\rho_m} & 0 & 1 \end{bmatrix}, \quad \Gamma_e = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \rho_m I_{3x3} & \vec{u}\Delta\rho \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}$$

Finally, the inviscid and viscous fluxes can be summarized as:

$$\vec{F}_c = \begin{bmatrix} V_n \\ \rho_m u\Delta V + pn_x \\ \rho_m v\Delta V + pn_y \\ \rho_m w\Delta V + pn_z \\ \alpha_l \Delta V \end{bmatrix}, \quad \vec{F}_v = \begin{bmatrix} 0 \\ \tau_{xx}n_x + \tau_{xy}n_y + \tau_{xz}n_z \\ \tau_{yx}n_x + \tau_{yy}n_y + \tau_{yz}n_z \\ \tau_{zx}n_x + \tau_{zy}n_y + \tau_{zz}n_z \\ 0 \end{bmatrix} \tag{3.13}$$

where $V_n = \vec{u} \cdot \vec{n}$, $V_g = \vec{v}_{vol} \cdot \vec{n}$, $\Delta V = V_n - V_g$, while $\vec{v}_{vol}$ is the velocity of the control volume and $\vec{n}$ is the surface normal of the control volume. The viscous stresses τij are computed as:

$$\tau_{ij} = (\mu_m + \mu_t)(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{3}{2}\rho\delta_{ij}k) \tag{3.14}$$

where μ_t is the turbulent dynamic viscosity, k is the turbulent kinetic energy and δ_{ij} is the Kronecker delta.

3.2 Numerical Framework

3.2.1 Spatial Discretization

The equations of flow (3.10) are discretized-based on the finite volume method. Control volumes are defined in every cell of the mesh with its center being the geometric center of the mesh elements. The unknown variables $\vec{Q}$ are expressed over a control volume D_i as:

$$\overline{\vec{Q}} = \frac{1}{D_i}\int_{D_i} \vec{Q}(\vec{x};t)dD \tag{3.15}$$

The surface terms of Equation (3.10) are considered constant in each face of the control volume. Thus, the surface integrals are approximated through a sum of surface terms evaluated at the midpoint of every face. Furthermore, the volume terms are considered constant in each D_i. As a result, the spatial terms are computed based on the following equation.

$$\vec{R_{D_i}} = \sum_{f}^{N_f} (\vec{F_c} - \vec{F_v})_f \Delta S_f - D_i \vec{S_q} \tag{3.16}$$

The inviscid fluxes are computed through the approximate Riemann solver of Roe. [7] As already mentioned, the eigenvalues of the inviscid Jacobian depend on the density field. For this reason, the preconditioned matrix Γ is used in the evaluation of fluxes. The preconditioned Jacobian A_c is defined as:

$$A_c = \frac{\partial \vec{F_c}}{\partial \vec{Q}} = \Gamma \Gamma^{-1} A_c = \Gamma \tilde{A}_c \tag{3.17}$$

where Γ is the preconditioned matrix (3.12) and Γ^{-1} is the inverse matrix.

The convective fluxes are computed, at a face f , using the Roe approximate Riemann solver as:

$$\vec{F}_{c,f} = \frac{1}{2}(\vec{F_c}(\vec{Q_R}) + \vec{F_c}(\vec{Q_L})) - \frac{1}{2}\overline{\Gamma|\tilde{A}_c|_f}(\vec{Q_R} - \vec{Q_L}) \tag{3.18}$$

where $|\tilde{A}_c|$ is the Roe averaged-precoditioned jacobian:

$$|\tilde{A}_c| = \tilde{R}^{-1}|\tilde{\Lambda}|\tilde{R} \tag{3.19}$$

where $\tilde{R}$, $\tilde{R}^{-1}$, $\tilde{\Lambda}$ are the right, left eigenvectors and the eigenvalues of the preconditioned matrix $\tilde{A}_c$.

The eigensystem is evaluated by taking the simple average the two neighbours that share the edge f.

$$\tilde{Q}_f = \frac{Q_{f,R} + Q_{f,L}}{2} \tag{3.20}$$

Continuing, it is necessary to describe the methods used for the reconstruction of the flowfield.

The evaluation of the inviscid fluxes in Equation (3.13) requires an approximation of the left and right state of a face (Figure 3.1). These states are computed based on a reconstruction scheme that extrapolates the cell-centered value of the volumes in the respective face. The right and left state are designated based on the normal vector of the face, which points from the left state to the right. Due to the particular nature of the governing equations, a different reconstruction scheme is adopted for each equation.

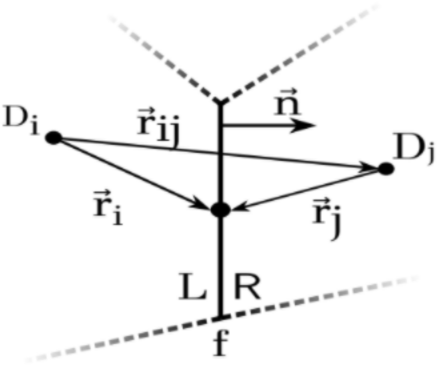


Figure 3.1: Reconstruction of variables on face f

The velocity field is approximated through a piecewise linear interpolation scheme, given by Equation (3.21). Since the surface tension is neglected, the velocity field is continuous even across the free surface. For this reason, the gradients are retained, and no limiter is applied.

$$\begin{aligned} \vec{u_L} &= \vec{u_i} - \nabla\vec{u_i}\cdot\vec{r_i} \\ \vec{u_R} &= \vec{u_j} + \nabla\vec{u_j}\cdot\vec{r_j} \end{aligned} \tag{3.21}$$

The vectors $\vec{r_i}$, $\vec{r_j}$ are pointing from the cell center to the midpoint of the face, as illustrated in Figure (3.1).

Furthermore, the pressure field is a continuous function in space, since surface tension is neglected. However, the pressure gradient is discontinuous across the free surface, due to the density jump. The jump condition that must hold requires $[\frac{\nabla p}{\rho}] = 0$. Many researchers have introduced different schemes to treat this difficulty by adopting a density-based interpolation scheme. In MaPFlow, the work of Queutey et al. [8] is followed. This scheme is introduced only near the free surface, while in the rest of the computational domain a piecewise linear interpolation scheme is used, similar to Equation (3.21).

Finally, of great importance is the reconstruction of volume fraction field (α_l). In order to reduce the numerical diffusion, it is important to adopt a compressive reconstruction scheme. Over recent decades, numerous reconstruction schemes have been introduced which offer low numerical diffusion. The requirements should meet is boundedness and high accuracy even in large CFL numbers. In the present work the STACS [9] scheme is adopted as the free surface capturing scheme.

3.2.2 Temporal Discretization

The present section focuses on the time discretization process. Let $\vec{Q}*$ be the vector of unknowns. At every pseudo-time step iteration a problem of the form (3.22) is solved.

$$\Gamma \frac{\partial(\vec{Q}*)D_i}{\partial \tau} + \vec{R}* \tag{3.22}$$

where $\vec{R}*$ is the unsteady residual defined as:

$$\vec{R}* = \vec{R}_{D_i}(\vec{Q}*) + \Gamma_e \frac{\partial(\vec{Q}*D_i)}{\partial t} \tag{3.23}$$

The convergent solution of the problem is obtained when $\vec{R}* \to 0$ and $\vec{U}* \to \vec{U}^{n+1}$ (n is the time marching iterator).

Discretizing the unsteady term at time level n+1, and writing the time derivative as a series expansion of successive levels backwards in time.

$$\frac{\partial(\vec{Q}*D_i)}{\partial t} = \frac{1}{\Delta t}[\phi_{n+1}(D_i\vec{Q})^{n+1} + \phi_n(D_i\vec{Q})^n + \phi_{n-1}(D_i\vec{Q})^{n-1} + \phi_{n-2}(D_i\vec{Q})^{n-2} + ...] \tag{3.24}$$

By introducing the Geometric Conservation Law (GCL) [10]

$$\frac{d}{dt}\int_{D_i(t)} dD = \oint_{\partial D_i(t)} \vec{u}_{vol} \cdot \vec{n} dS \tag{3.25}$$

Using a similar backwards differentiation in time as in Equation (3.24) the GCL can be written as:

$$\frac{1}{\Delta t}[(\phi_{n+1}D_i^{n+1} + \phi_n D_i^n + \phi_{n-1}D_i^{n-1} + \phi_{n-2}D_i^{n-2}) + ...] = \vec{R}_{GCL}^{n+1} \tag{3.26}$$

where the residual of the GCL is defined as:

$$\vec{R}_{GCL}^{n+1} = \sum_f^{N_f}(V_g \Delta S)_f^{n+1} \tag{3.27}$$

To ensure that the GCL is satisfied, Equation (3.26) is applied directly to the discretization of the unsteady term and thus Equation (3.24) becomes:

$$\frac{\partial(\vec{Q}D_i)}{\partial t} = \vec{Q}\vec{R}_{GCL}^{n+1} + \frac{1}{\Delta t}[(\phi_{n+1}(\vec{Q}^{n+1} - \vec{Q}^{n})D_i^{n+1} + \phi_{n-1}(\vec{Q}^{n-1} - \vec{Q}^{n})D_i^{n-1} + \phi_{n-2}(\vec{Q}^{n-2} - \vec{Q}^{n})D_i^{n-2} + ...] \tag{3.28}$$

In MaPFlow two successive levels of solution are retained, yielding a second order accurate scheme in time. The fictitious time derivative of equation is discretized using a first-order backward difference scheme:

$$\frac{\partial(\vec{Q}D_i)}{\partial t} = D_i^{n+1}\frac{\vec{Q}^{*,k+1} - \vec{Q}^{*,k}}{\Delta\tau} = D_i^{n+1}\frac{\Delta\vec{Q}^{*,k}}{\Delta\tau} \tag{3.29}$$

To facilitate convergence the local time stepping technique is used. The local pseudo-timestep is determined by:

$$\Delta\tau = CFL\frac{D_i}{\widehat{\Lambda}_{c,i}} \tag{3.30}$$

where $\widehat{\Lambda}_{c,i}$ is the convective spectral radii and it is defined by:

$$\widehat{\Lambda}_{c,i} = \sum_{j=1}^{N_f}(\mid \vec{V}_n \mid + c - \frac{V_g}{2})_{ij}\Delta S_{ij} \tag{3.31}$$

Chapter 4

Immersed Boundary Method

4.1 Fundamental Characteristics

The **Immersed Boundary Method (IBM)** was developed by Peskin in 1972. [11] He simulated heart function and blood flow through the heart by solving the Navier-Stokes three-dimensional equations. This method has been extremely innovative, as its philosophy differs significantly from that of the already existing Body-fitted Methods. More specifically, while in the methods of the Body-fitted grids the grid is adapted to the geometry under study, in Immersed Boundary Method geometry is immersed in the Cartesian grid that has been produced.

It is understood that as the geometry is immersed within the grid, other computational cells belong exclusively to the geometry, others belong exclusively to the flow, while there are also those intersected by the geometry. In computing cells belonging to the flow, the determination of the flow sizes of the field is performed with Euler variables (Eulerian specification of the flow field). Therefore, it is considered that in each computing cell there is a stationary observer monitoring the flow that enters and leaves it. Instead, the motion of the immersed body is described by adopting Lagrange variables, where the change in the position of the geometry is determined by the observer who follows its motion.

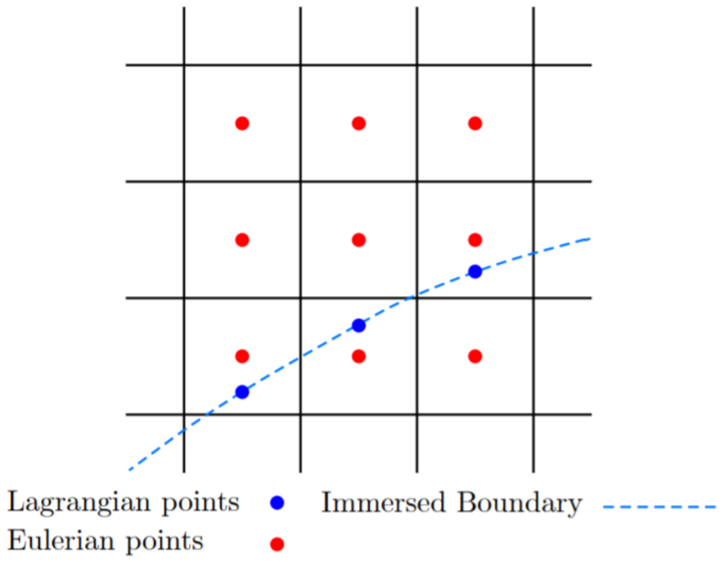


Figure 4.1: Fixed grid using description of Euler variables for storage of flow quantities and immersed solid boundary using a description of Lagrange variables.

In the case of body-fitted grids, the process to construct the grid involves the following two steps:

- Production of surface grid that completely covers the geometry under study. In this interface it is placed the boundary condition for the grid to be constructed for the area outside the solid body where the flow happens.
- Discretization of the equations that describe the flow, either using finite finite-difference method, either using finite-element method or, finally, using finite-volume method. Of course, the choice of each of the above methods of discretization requires the use of a specific form of equations that describe the flow.

On the other hand, in IBM, although the surface geometry of the solid body is preserved, it does not transmit information to the Cartesian grid produced for the domain containing the flow. Thus, geometry penetrates into Cartesian grid "randomly" cutting a large number of computational cells. Hence, the fact that the Cartesian grid is not adapted to geometry, requires the search for other methods of applying the boundary conditions and consequently the creation of different Immersed boundary methods.

4.2 Categories of Immersed Boundary Methods

As already mentioned, the way the boundary conditions are imposed on the immersed boundary determines for the most part how an IB algorithm will look like. It is also what distinguishes one IB method from another. The simulation of a viscous incompressible flow past a body is described by the Navier-Stokes equations (governing equations). [12]

$$\rho\,(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) + \nabla p = 0 \tag{4.1}$$

$$\nabla \cdot \mathbf{u} = 0 \;\; , in \;\; \Omega_f \;\; and \tag{4.2}$$

$$\mathbf{u} = \mathbf{u}_\Gamma \;\; on \, \Gamma_b \tag{4.3}$$

where $\mathbf{u}(\mathbf{x}$, t) is the fluid velocity and p($\mathbf{x}$, t) is the pressure. The coefficients ρ and μ are the constant fluid density and viscosity, respectively. The solid body occupies the domain Ω_b, with boundary denoted by Γ_b, and the surrounding fluid domain denoted by Ω_f.

We will examine the various categories of the Immersed Boundary Methods in more detail below.

4.2.1 Continuous Forcing Approach

In the continuous approach, a forcing function **F** is used, which simulates the effect of the solid boundary over the whole area of the computational domain. This function can be considered as a source term, which is added to the NS equations (4.1) and has an increased effect on nearby cells at the solid limit and dampens away from it. [13]

$$\rho\,(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}) + \nabla p = \mathbf{F}_k \tag{4.4}$$

Several methods make use of a continuous forcing approach [12] :

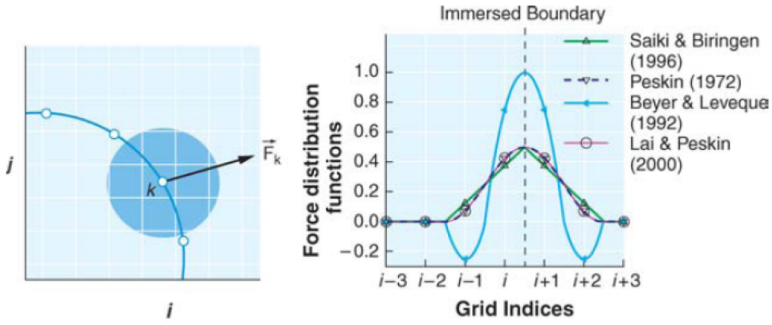
- **Elastic boundaries**: This category was used by Charles S. Peskin in 1972 for the simulation of cardiac function [1] [4]. Such as mentioned above, this method is a mixture of Euler and Lagrange variables, and finite differences, which calculates the interaction of flow with a flexible immersed limit. [11] [13]. Peskin defines the force density, $\mathbf{f}(\mathbf{x}$,t), by a δ-function layer that represents the force applied by the immersed boundary to the fluid. Therefore, the forcing is distributed over a band of cells around each Lagrangian point (Fig. 3.1(a)) and this distributed force will be used in the momentum equations of the surrounding nodes. There are more approaches for the distribution function developed over the years and some of them are shown in Fig. 3.1(b).

- **Rigid boundaries**: The first approach for rigid boundaries is called virtual boundary method, used by Goldstein et al. [14] The main idea of the virtual boundary method is to treat the body surface as a virtually existent boundary embedded in the fluid. This boundary applies force on the fluid so that the fluid will be at rest on the surface (no-slip condition). To avoid interpolating the velocity field from grid points to the boundary points, Goldstein et al. let the boundary points coincide with grid points. However, in order to generate a smooth surface rather than a step-like surface, the boundary force is multiplied by a narrow Gaussian distribution so that the nearby grid points can receive a part of the force influences.

- **Distributed Lagrange multiplier method**: The distributed Lagrange multiplier method (DLM), proposed by Glowinski et al. [15], uses a variational principle (finite element) as framework. The idea is to introduce Lagrange multipliers (i.e. body force) on the immersed rigid body to satisfy the no-slip condition. After that, a finite element approximation is used on the rewritten problem with Lagrange multipliers.

- **Immersed Interface Method**: In IIM, as developed by L. Lee and R.J. Leveque [16], the equations of force of the elastic limits are used and the interface is detected in Lagrangian manner.

An advantage of the continuous forcing approach is that the above described methods are independent of the underlying spatial discretization in contrast to methods that are based on a discrete forcing approach. Therefore, this approach can be implemented into an existing Navier-Stokes solver with relative ease. A disadvantage of these methods is that the smoothing of the forcing function inherently leads to an inability to provide a sharp representation of the immersed boundary and therefore these methods are not useful for high Reynolds number flows. Another drawback of the continuous approach is that they all require the solution of the governing equations inside immersed body. With increasing Reynolds numbers the proportion of grid points inside the IB also increases.



(a) Transfer of forcing $\mathbf{F}_k$ from Lagrangian boundary point ($\mathbf{X}_k$) to surrounding fluid nodes.

(b) Distribution functions.

Figure 4.2: Effect of explicitly adding a forcing function.

4.2.2 Discrete Forcing Approach

The discrete approach is better suited for higher Reynolds numbers, due to imposing the velocity boundary conditions at the immersed boundary, without introducing or computing any forcing term. The methods that will be discussed here in detail are the direct-forcing method and extensions of it, like the ghost-cell method, and the hybrid Cartesian/immersed boundary method. [12]

- **Direct forcing method**: The (spectral) method of Mohd-Yusof [17] uses a forcing term, which is determined by the difference between the interpolated velocities in the boundary points and the desired (physical) boundary velocities. The forcing term, generated in this manner, thus directly compensates the errors between the calculated velocities and the desired velocity profile on the body surface. The force is thus determined by pairing the velocity at the internal point to the velocity at the external point with a weighted linear interpolation, to enforce the desired tangential velocity on the boundary, i.e. the method mirrors the velocity field across the immersed boundary. An example of such a pair is illustrated in Fig. 4.3 between Point 1 and Point 2

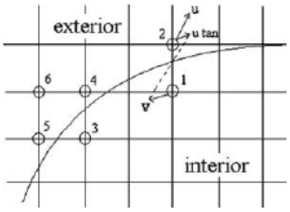
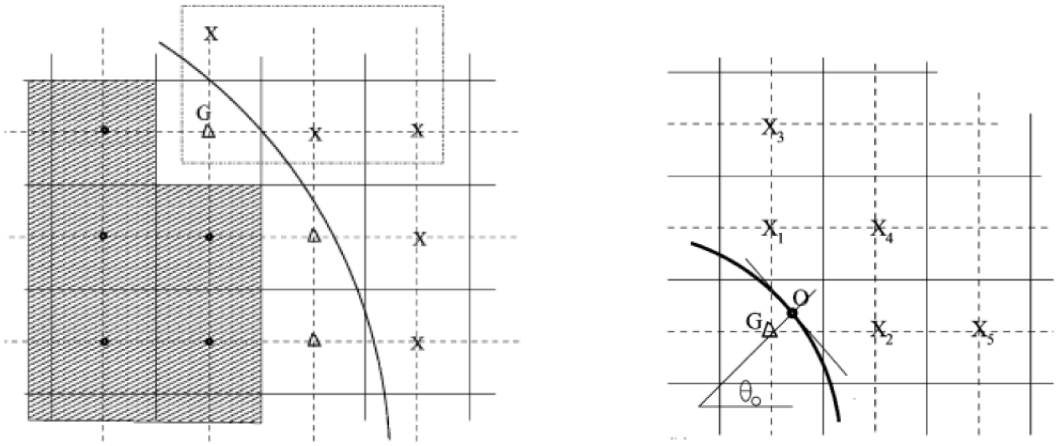


Figure 4.3: Schematic interpolation of Mohd-Yusof method.

- **Ghost-Cell method**: It was developed by Yu-Heng Tseng and Joel H. Ferziger [5] and according to this method, a higher order representation of the boundary is attempted, using a zone with pseudo-cells inside the body. Pseudo-cells are called cells that have at least one adjacent cell with a node in the fluid. For each such cell, one interpolation scheme must be created, which indirectly incorporates the boundary condition into immersed boundary. A simple solution for the interpolation scheme is bilinear interpolation for a two-dimensional field or a trilinear interpolation for three-dimensional field. Another scheme is the combination of linear (in the tangential direction) and polynomial interpolation (in the normal direction). Example of the Ghost-cell method is shown supervisingly in Figure 4.4.



(a) Schematic of computational domain with an immersed boundary. X, point in the physical domain and Δ, the ghost cell domain.

(b) Schematic representation of the linear interpolation procedure

Figure 4.4: Ghost-Cell method.

An advantage of the methods treated in this section is that they all can make a sharp representation of the immersed boundary, which is necessary for high Reynolds numbers. They do not introduce any extra stability constraints in the representation of solid bodies, due to absence of user-specified parameters in the forcing and the elimination of associated stability constraints. The methods decouple the equations for fluid nodes from solid grid points. A disadvantage of these methods is that they all strongly depend on the discretization method in contrast to the continuous forcing approach. However, this allows direct control over the numerical accuracy, stability and discrete conservation properties of the solver. Another drawback is that these methods are not straightforward for implementation due to first discretization and then introducing a forcing term. The methods which are categorized into the continuous forcing section do not suffer from this difficulty. Also inclusion of boundary motion can be more difficult.

4.3 Advantages and Disadvantages of IB methods

The various IB methods have advantages but also disadvantages compared to the classic methods of body-fitted grids (structured, unstructured or hybrid grid), which adapt the grid in each geometry. [12]

- **Advantages**: First of all, the grid generation using an IB method is much easier, since a body does not necessarily have to fit conform a Cartesian grid. Another benefit is that grid complexity and quality are not significantly affected by the complexity of the geometry when carrying out a simulation on a non-boundary conforming Cartesian grid. Also, an immersed boundary method can handle moving boundaries, due to the stationary non-deforming Cartesian grid. As a result of the above remarks, an immersed boundary method uses less memory and CPU compared to the usual method, a body fitted grid and the thereby belonging transformations. In comparison with structured curvilinear body-fitted grids, Cartesian grids reduce the per-grid-point operation count due to absence of additional terms associated with grid transformations. When comparing to unstructured curvilinear body-fitted grids, Cartesian grid-based IB methods are amenable to powerful line-iterative techniques and geometric multigrid methods, leading to a lower per-grid-point operation count. Also multi-phase and multi-material problems, where the interface is between different materials, can be regarded as immersed boundary problems.

- **Disadvantages**: A disadvantage is that imposing of the boundary conditions is not straightforward compared to the traditional methods. Also, the ramifications of the boundary treatment on accuracy and conservation properties of numerical schemes are not trivial. Another drawback is the following. Alignment between grid lines and body surface in boundary conforming grids allows for better control of the grid resolution in the vicinity of the body and this has implications for the increase of grid size with increasing Reynolds numbers. However, a substantial fraction of grid points can be inside the solid body, i.e. where the fluid flow equations need not to be solved.

4.4 Cut-Cell Algorithm

The method of discretization of the computational domain used in this dissertation is the Cut-Cell Method. The Cut-Cell method can be categorized into discrete approach methods, but at the same time it differs significantly, since its basic element is the intersection of the cells (ie the change in the geometry of the cells), which are intersected by the immersed boundary.

The Cut-Cell method makes possible the discretization with the method of finite volumes. This capability of the method, makes it be the only distinct method IB, which explicitly satisfies, both in each cell separately, as well as throughout the computational area, the laws of conservation of mass, momentum and flow energy. Also, another advantage of the method of intersecting cells, compared to other distinct IB methods is the saving of computational time, because the flow equations are not solved at the nodes of the space located within the solid boundaries.

For a start, the grids were produced in the commercial program **ANSA** a computer-aided engineering tool for Finite Element Analysis and CFD Analysis developed by **BETA CAE Systems**.

Through this program, a computational space of certain dimensions is created, which is spatially differentiated into a number of cells. The grid distance, the number of cells and the grid type (structured or unstructured) differ from region to region within the computational space depending on the physical conditions of the problem at a time. The common denominator of the grids made is that in the area where the body geometry is to be created (via the IBM method) the grid is structured Cartesian.

Once constructed, the grid is inserted into the MaPFlow solver where the intersecting cell algorithm works to create the body geometry we want in our simulation.

Thus, the Cartesian grid is cut and adapted to the geometry of the solid body.

Now it would be useful for reasons of completeness of the dissertation to mention in more detail the programming details for the implementation of the cut cell algorithm.

Initially, each computational cell in the Cartesian grid is described by 4 nodes, to which a number is assigned to distinguish between them. Then each of the nodes has some spatial coordinates that determine its position in the computational domain of the grid. The geometries of the external boundaries of the bodies that we will examine in this dissertation are characterized by simple mathematical equations.

For example the two-dimensional plane cylinder described below is expressed by the equation:

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \tag{4.5}$$

with the center being at the point $(x_0\ ,\ y_0)$ and the radius being "r".

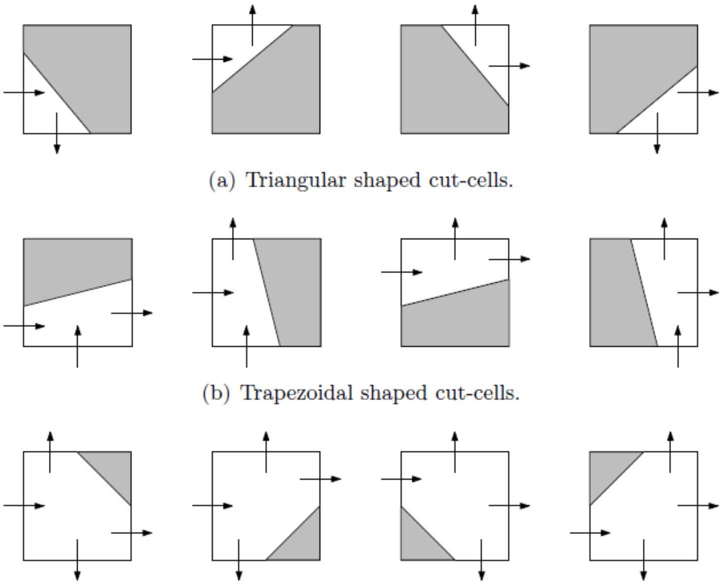
On the other hand, in the simulation of the wave propagation over a bar that we will see in a later chapter, the change in the geometry of the 'bottom' is described by simple linear equations.

$$y = a \cdot x + b \tag{4.6}$$

Therefore, equations like these in equations (4.5) and (4.6) intersect some cells in our grid and are thus interposed between the nodes of these cells. Solving these equations, with the help of the coordinates of the nodes located on either side of the body boundary, the new nodes of the cells that are the intersections of the cells with the geometric boundary of the body are obtained.

The resulting new cells require us to update the information that describes them. This information concerns the number and numbering of nodes and faces that describe each cut cell.

Eventually, the intersecting cells that arise will be triangular, trapezoidal or pentagonal. (fig. 4.2)



(a) Triangular shaped cut-cells.

(b) Trapezoidal shaped cut-cells.

(c) Pentagonal shaped cut-cells.

Figure 4.5: Overview of all possible Cut-cells.

The criterion used in each intersecting cell to figure out which of the three categories it belongs to (triangular, trapezoidal, pentagonal), and then to change the information that describes it and to create new cells, is to repeatedly measure its nodes that lie outside the physical boundary of the body.

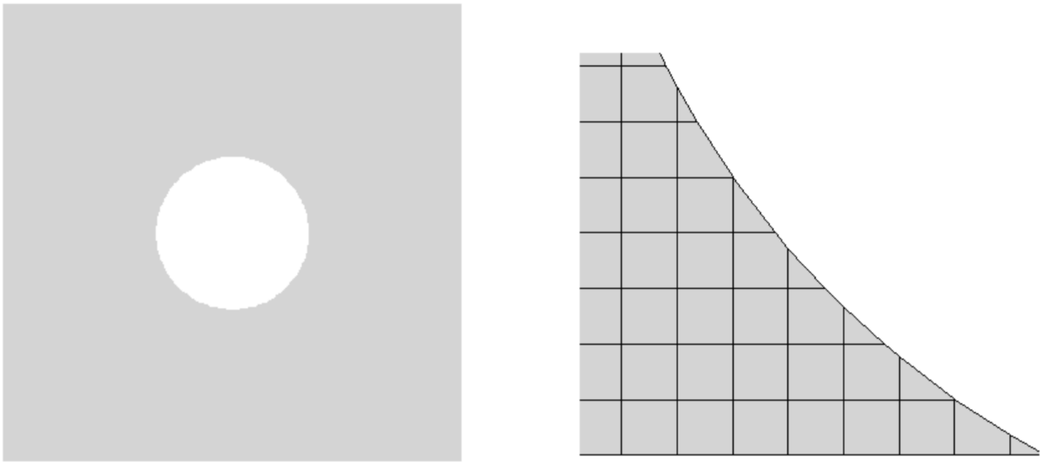
Cells with three external nodes are converted to pentagonal, cells with two to trapezoidal and finally cells with one to triangular.

A special case is when one or more of the nodes of the intersecting cell rest directly on the boundary of the body. This case is divided into three different subcases. First, when one or more of the cell nodes are inside the body and the rest are in the contour, then the latter are considered internal nodes and the cell as we will see below is completely removed. If one or more of the cell nodes are outside the body and the remaining nodes are in the contour, then the latter are considered external nodes and the cell is not modified by the algorithm. In intermediate states the nodes located on the boundary of the body are considered internal and the cell is modified accordingly.

This means that the triangular cells are described by 3 nodes and 3 faces, the trapezoidal cells by 4 nodes and 4 faces while the polygonal cells by 5 nodes and 5 faces.

As it is easily understood curved boundaries, such as the geometric boundary of the cylinder, are approximated by linear line segments. The result is that the cylinder is finally equivalent in our grid to a polygon, the number of angles of which depends on the radius of the cylinder and the grid spacing of the cells.

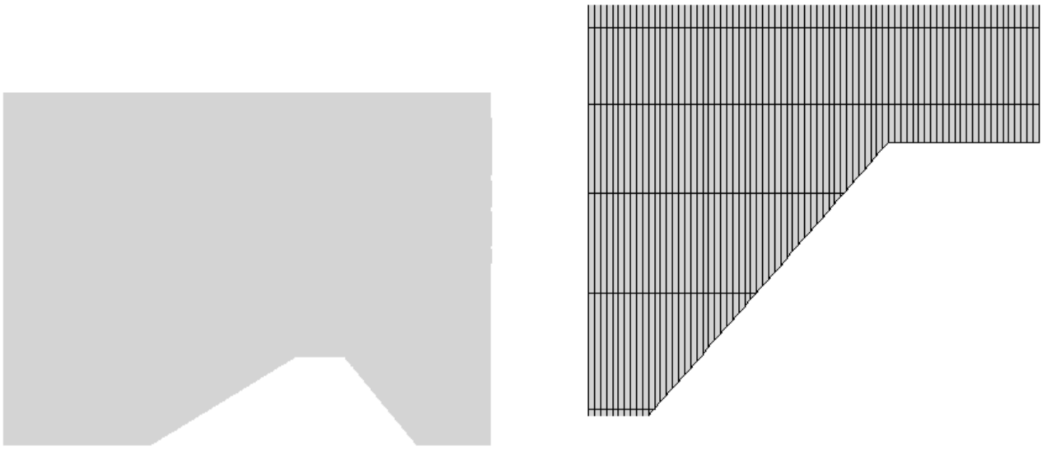
The final image of the cylinder after the reconstruction of the cells is given in the figure (4.3). Also, the form of the grid is given for the study of the wave propagation over a bar in the figure (4.4).



(a) Cylinder's normal view.

(b) Grid in zoomed view.

Figure 4.6: The shape of the cylindrical body in the grid after its adjustment.



(a) Bottom's normal view.

(b) Grid in zoomed view.

Figure 4.7: The shape of the 'bottom' in the grid after its adjustment.

As we observe in both figures above, the area within the boundaries of the solid body is not covered by computational cells. The latter moved away from this area while setting a new boundary condition on the faces that contribute to the creation of the border, that of the wall.

For each face that participates in the composition of the body boundary there is a ghost cell, ie a pseudo-cell that serves to define the boundary condition at that point in the computational space.

The removal of the cells that are inside the body means that the numbering of the cells that remain in the computational space as well as the nodes and the faces that describe them must be adjusted accordingly. Also, after changing the numbering of the cells, the information about which cell is to their right and which to their left must be modified.

m

Then, the last step for the final shape of the grid before applying the finite volume method, is the recalculation of the area and the center of the cells after their intersection.

The centers of the new cells are calculated by finding the average of the coordinates of their nodes, while the area is found after dividing each cell into triangular parts and adding the areas of the triangles.

Chapter 5

Laminar Flow past a Cylinder

The intersecting Cut-Cell algorithm developed in this dissertation found as its first application an unsteady laminar flow around a cylinder. In spite of extensive experimental and numerical studies almost over a century, flow around a circular cylinder still remains a challenging problem in fluid mechanics, where intensive investigations are continued even today to understand the complex unsteady dynamics of the cylinder wake flow. [18]

All flow simulation is performed on a two-dimensional level. The equations, therefore, solved for finding the effects of flow in the presence of the cylinder are the two-dimensional mass conservation and the x- and y- components of the momentum equation. The MaPFlow solver is of course used to solve the equations. The case includes simulations made with a number of different Reynolds numbers to observe the behaviour of the flow in a variety of regimes.

The physical mechanism which rules flows around a circular cylinder as well as its mathematical treatment are highly complex. Some difficulties encountered in this case are the boundary layer detachment and oscillatory effects that without a CFD contribution would be very complicated to determine either analytically or experimentally. [19]

The hypotheses made in the study of a bidimensional laminar flow around a circular cylinder are those listed just below :

- Incompressible flow
- Laminar flow
- Newtonian flow
- Bidimensional flow ($\frac{\partial}{\partial z} = 0$)
- Negligible gravitatory effects

5.1 Physics of the problem

We consider the two-dimensional, unsteady, cross-flow of an incompressible and Newtoninan liquid streaming with a uniform velocity (Uo) over a circular cylinder (of diameter, D = 1 m), as shown in Fig. 1. The unconfined flow condition is simulated here by considering the flow in a channel with the cylinder placed symmetrically in between the two plane walls with slip boundary conditions, as shown schematically in Fig. 1. The length and height of the computational domain are L (= Lu + Ld) and H, respectively. The cylinder is placed at an upstream distance of Lu from the inflow boundary and at a distance of Ld (downstream length) from the outflow boundary. The dimensions of the computational domain (Fig. 1) are taken to be sufficiently large to minimize the boundary effects, as discussed in detail in a later section. [20]

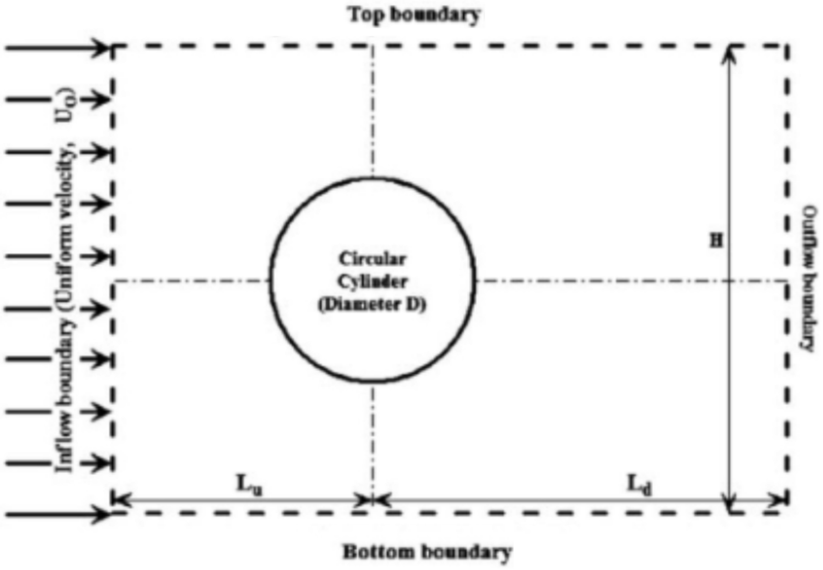


Figure 5.1: Schematic representation of an unconfined flow over a cylinder

The continuity and momentum equations for this flow are written as follows:

- Continuity equation :

$$div\,\mathbf{u} = 0 \tag{5.1}$$

- Momentum equation :

$$\begin{aligned} x - component, \quad & \frac{\partial(\rho u)}{\partial t} + div(\rho u \mathbf{u}) = -\frac{\partial p}{\partial x} + div(\mu\, grad\, u) \\ y - component, \quad & \frac{\partial(\rho v)}{\partial t} + div(\rho v \mathbf{u}) = -\frac{\partial p}{\partial y} + div(\mu\, grad\, v) \end{aligned} \tag{5.2}$$

The physically realistic boundary conditions for this flow configuration can be written as follows :

- At the inlet bound : The uniform flow condition is imposed at the inlet.

$$U_x = U_0 \quad and \quad U_y = 0 \tag{5.3}$$

- On the surface of the cylinder: The standard no-slip condition is used.

$$U_x = 0 \quad and \quad U_y = 0 \tag{5.4}$$

- At the exit boundary we take the cells in our exit boundary as internal cells using ghost cells outside of the computational domain.

- At the top and bottom boundaries: The slip flow condition is imposed.

$$\frac{\partial U_x}{\partial y} = 0 \quad and \quad U_y = 0 \tag{5.5}$$

The numerical solution of the governing equations (Eq.(5.1) and (5.2)) in conjunction with the above-noted boundary conditions (Eq.(5.3)–(5.6)) maps the flow domain in terms of the primitive variables, i.e., velocity (Ux and Uy) and pressure (p) fields.

Also, it is useful to introduce the dimensionless numbers involved in the problem:

- Reynolds Number:

$$Re = \frac{D \mid U_o \mid}{\nu} \tag{5.6}$$

  where ν is the kinematic viscosity of the fluid (m^2/s) and U is the flow speed (m/s).

- Pressure coefficient (C_P) :

$$C_P = \frac{Static\ pressure}{Dynamic\ pressure} = \frac{p - p_\infty}{\frac{1}{2}\rho U_\infty^2} \tag{5.7}$$

  where p is the static pressure at the point at which pressure coefficient is being evaluated and p_∞ is the static pressure in the freestream.

- Total drag and lift coefficients, C_D and C_L, respectively :

$$C_D = \frac{F_D}{\frac{1}{2}\rho \mid U \mid^2 S} \tag{5.8}$$

$$C_L = \frac{F_L}{\frac{1}{2}\rho \mid U \mid^2 S} \tag{5.9}$$

where F_D and F_L are the drag and lift forces, respectively. S is the reference area.

Lift is the component of the force that the fluid exerts on the body and it is perpendicular to the oncoming flow direction, while the drag force is the component of the force parallel to the flow direction.

Lift and Drag can be found by integrating pressure and wall-shear stresses.

$$F_D = \int_A dF_D = \int_A (-Pcos\theta + \tau_w sin\theta) dA \tag{5.10}$$

$$F_L = \int_A dF_L = -\int_A (-Psin\theta + \tau_w cos\theta) dA \tag{5.11}$$

- Strouhal number (St) is, the dimensionless frequency of the vortex shedding, defined as:

  $$S_t = \frac{fD}{U_0} \tag{5.12}$$

  where f is the frequency of vortex shedding. This frequency can be obtained using the **Fast Fourier Transform of the lift coefficient** time distribution.

One of the central concepts involved in this particular simulation is that of **vorticity**. For this reason it would be useful to define it in a brief and coherent way.

The vorticity is a pseudovector field that describes the local spinning motion of a fluid near some point, as would be seen by an observer located at that point and traveling along with the fluid. [19] Mathematically, the vorticity of a given velocity field,

$$\mathbf{u} = (u, v, w) \tag{5.13}$$

is defined to be the curl of the velocity field, and is usually denoted with the greek letter ω,

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} \tag{5.14}$$

The vorticity of a two-dimensional flow is always perpendicular to the plane of the flow. It can be considered a scalar field and its unit is 1/s, i.e. Hz.

It plays a relevant role in the current chapter due to the existence of vortices generated by the cylinder forming the von Kármán street.

5.2 Computational Grids

In total, three computational grids were constructed to simulate the flow around the cylinder. All three grids had some common characteristics. We did not want the computational domain to be uniform, which is why we divided it in different blocks.

In the area around the walls of the cylinder we created a square domain which contained a very dense Cartesian grid. Instead, in the area behind the cylinder and downstream we define an unstructured grid with a density similar to that of the grid around the cylinder. Moving away from the cylinder and its outlet section, the mesh density gradually decreases. In the rest of the space, between the cylinder and the walls, as well as on the flow inlet side, we choose the grid to be unstructured again, but quite sparse in relation to the aforementioned parts of the computational domain.

In table 5.1 we attach the general characteristics of the three grid confirugations.

Table 5.1: Parameters used for the different grid configurations. Δ_S: grid spacing in the Cartesian domain around the cylinder, L: characteristic dimension of the grid in the space covered by the wake, L_u: upstream length,i.e.,the distance from the inlet boundary to center of the cylinder, m, L_d: downstream length,i.e.,distance from center of the cylinder to outflow boundary, m, H height of the computational domain, m.

Grids		L_u	L_d	H	Δ_S	L	Total nodes
G1	2-D	12	22	14	0.05	0.1	19 262
G2	2-D	20	40	80	0.05	0.1	28 729
G3	2-D	20	40	80	0.025	0.1	37 106

On the walls and downstream the cylinder it is required to construct a highly refined grid. It will allow us to compute the drag coefficient and to observe the von Kármán street phenomenon with high accuracy. However, if the simulation has a high computational cost over time or it is done with a Reynolds number such that doesn't cause the generation of any vortices, it is preferable to reduce the refinement of the grid.

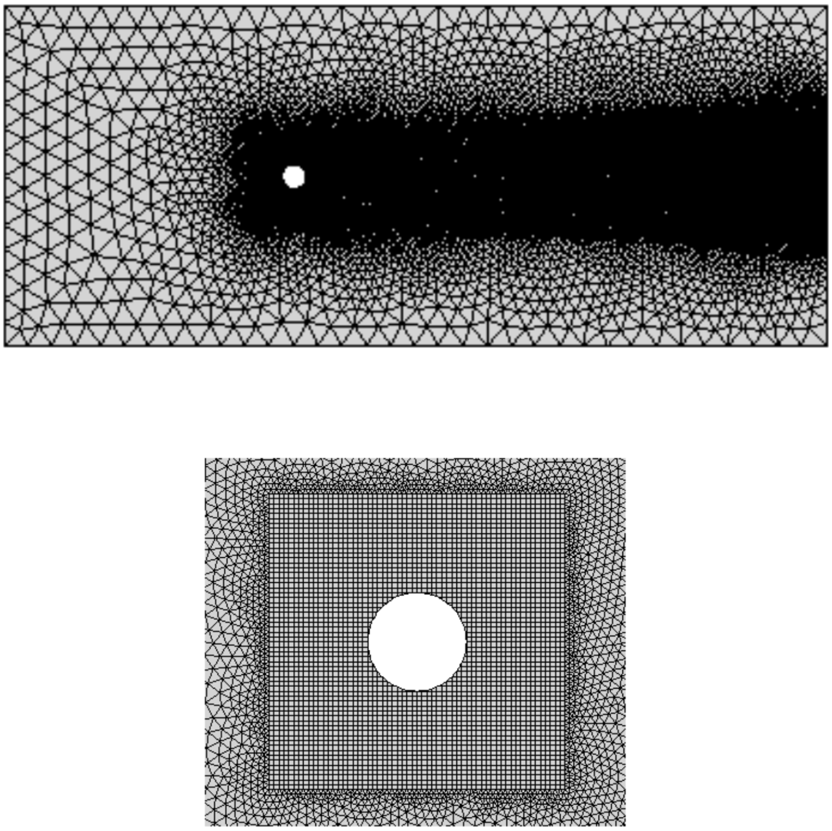
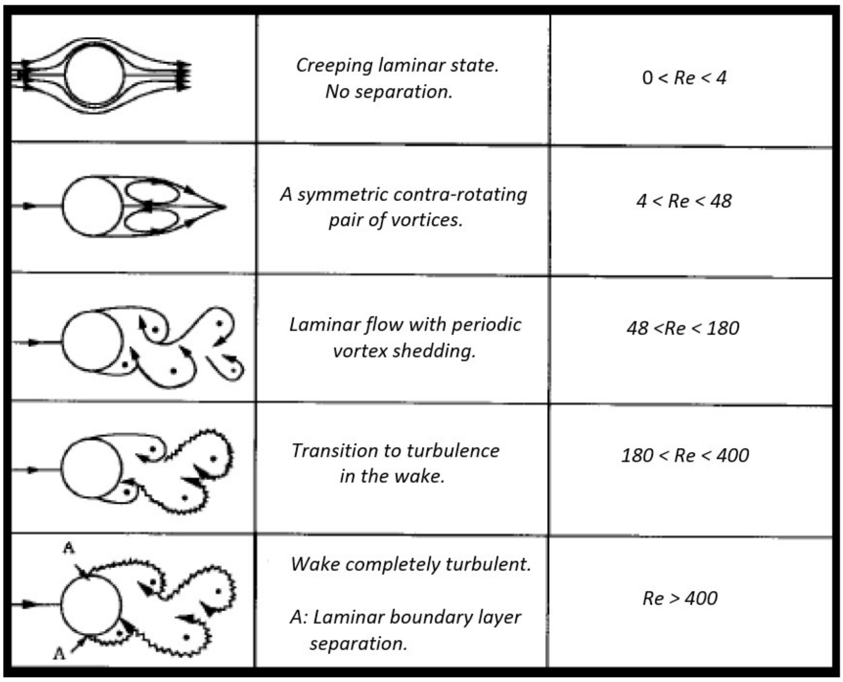


Figure 5.2: Computational grid G1. The pictures on the top shows the whole domain of G1, while the bottom picture shows an expanded view in the vicinity of the cylinder.

In 1997 Zdravkovich [21], compiled almost all the experimental, analytical and numerical simulation data on flow past cylinders, available since 1938 and systematically classified this challenging flow phenomenon into five different flow regimes based on the Reynolds number. The relation between the behaviour of the flow and the Reynolds number is shown at Fig. 5.2:



Creeping laminar state. *No separation.*	*0 < Re < 4*
A symmetric contra-rotating pair of vortices.	*4 < Re < 48*
Laminar flow with periodic vortex shedding.	*48 <Re < 180*
Transition to turbulence in the wake.	*180 < Re < 400*
Wake completely turbulent. *A: Laminar boundary layer separation.*	*Re > 400*

Figure 5.3: Flow structure depending on the Reynolds number

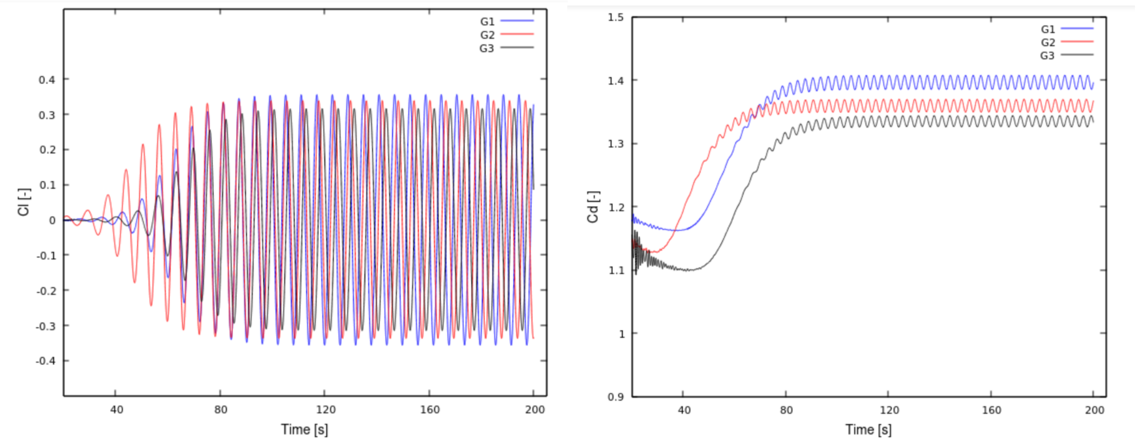
Due to the symmetry of the cylinder, at certain velocities, an oscillating flow takes place named **vortex shedding** which leads to the **von Kármán street** phenomenon. The latter consists of alternate vortices emitted by the cylinder. The vortices are detached periodically and their frequence is directly related to the Strouhal number. Hence, this phenomenon is responsible for the unsteady separation of flow of a fluid around blunt bodies.

Bluff body wake flows have direct engineering significance. The alternate shedding of vortices in the near wake, in the classical vortex street configuration, leads to large fluctuating pressure forces in a direction transverse to the flow and may cause structural vibrations, acoustic noise, or resonance. This means that if the bluff structure is not mounted rigidly and the frequency of vortex shedding matches the resonance frequency of the structure, then the structure can begin to resonate, vibrating with harmonic oscillations driven by the energy of the flow, which in some cases can trigger failure. [22]

As the simulation is going to be run with laminar flow, the Reynolds number is not going to be set higher than 200. This will allow to observe the **von Kármán street** and its transition while guaranteeing that the whole fluid domain remains laminar.

5.3 Results

We quote the comparative results for computational grids G1, G2 and G3, for Reynolds number Re = 100.



(a) C_l in relation to real time (b) C_D in relation to real time

Figure 5.4: Comparison of lift and drag coefficients for the three computational grids at Re = 100.

In detail, the numerical values of the lift and drag coefficients and the Strouhal number are listed in Table 5.2 just below:

Table 5.2: Numerical values of $\bar{C}_D$, C_l and St for the G1, G2 and G3 grids, for Re = 100

	$\bar{C}_D$	C_L, max	St
G1	1.3962	0.3559	0.1599
G2	1.3593	0.3387	0.1599
G3	1.3437	0.3163	0.1599

In the first grid, with the narrower geometric limits, the values of lift and drag coefficients as well as the strouhal number for reynolds number Re = 100, showed that there is a deviation with those found in the rest of the bibliography. We quote the relevant diagrams at Fig. 5.2 directly below :

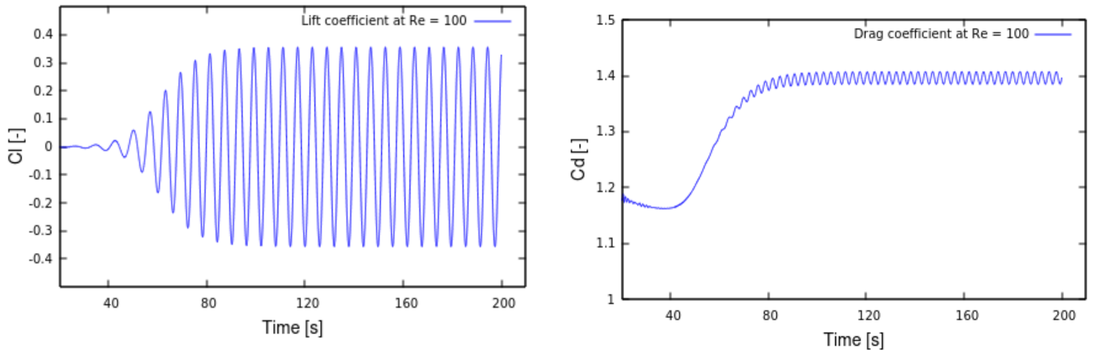
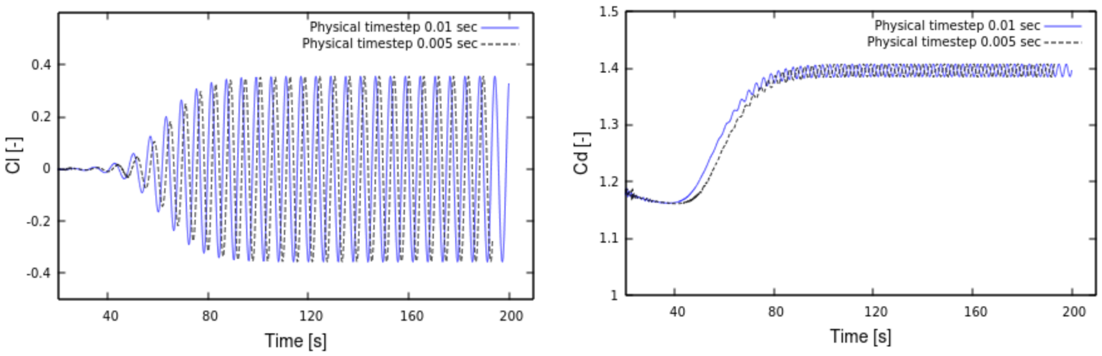


Figure 5.5: Lift and Drag coefficients for computational grid G1 at Re = 100.

We tried to improve the obtained results by reducing the physical timestep in the middle, doubling at the same time the number of total timesteps. As we observe at Fig. 5.3, the difference of the physical timestep in the two cases does not affect the results.



(a) C_l in relation to real time

(b) C_D in relation to real time

Figure 5.6: Comparison of lift and drag coefficients for two different physical time steps at Re = 100.

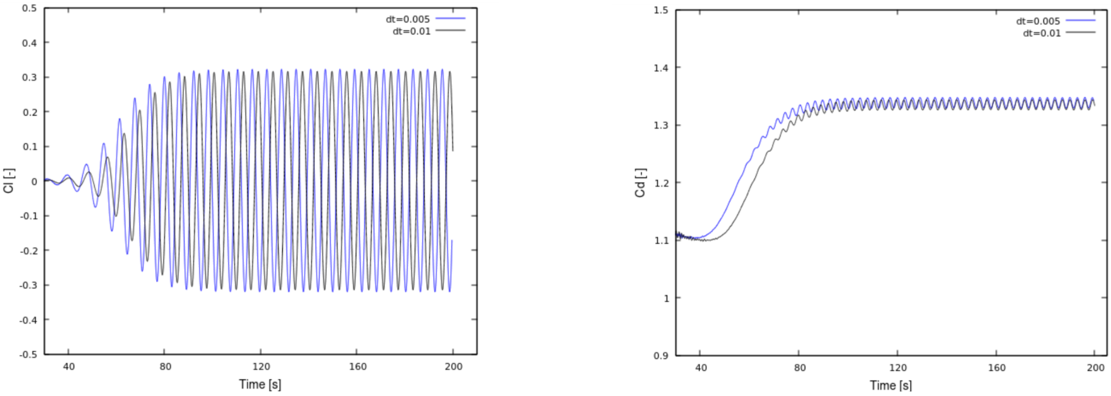
The first grid was 'narrow', this accelerated the flow and therefore increased the resistance. For this reason we built a new grid this time with expanded boundaries (larger outer frame) which improved the exported results. In this second grid, we had two cases: in one case in the zone of the Cartesian grid around the cylinder we had a denser area (cell dimension 0.025) while in the second case we had a thinner area (cell dimension 0.05). In both cases the results were satisfactory with small differences between them.

In the table below we compare the results of the literature with those we produced in this dissertation.

Table 5.3: Comparison of mean drag coefficient $\bar{C}_D$, C_L, max and Strouhal number, for Re = 100, with those of other authors. [20]

Source	$\bar{C}_D$	C_L, max	St
Present	1.3437	0.3163	0.1599
Baranyi (2004)	1.3460 ± 0.006	-	0.1630
Cheng et al. (1997)	1.3200	-	-
Mettu et al. (2006)	1.3020	0.2690	0.1600
Henderson (1995)	1.3490	-	-
Sivakumar et al. (2006)	1.3250	0.3200	0.1641
Mittal (2005)	1.3220	0.3190	0.1644
Clift et al. (1978)	1.3300	-	0.1670
Ding et al. (2004)	1.325 ± 0.008	0.2800	0.1640
Braza et al. (1986)	1.364 ± 0.015	0.2500	0.1600
Liu et al. (1998)	1.350 ± 0.012	0.3390	0.1640

For our densest grid, ie the third, we also perform a time step independence for values dt = 0.005 s and dt = 0.01 s.

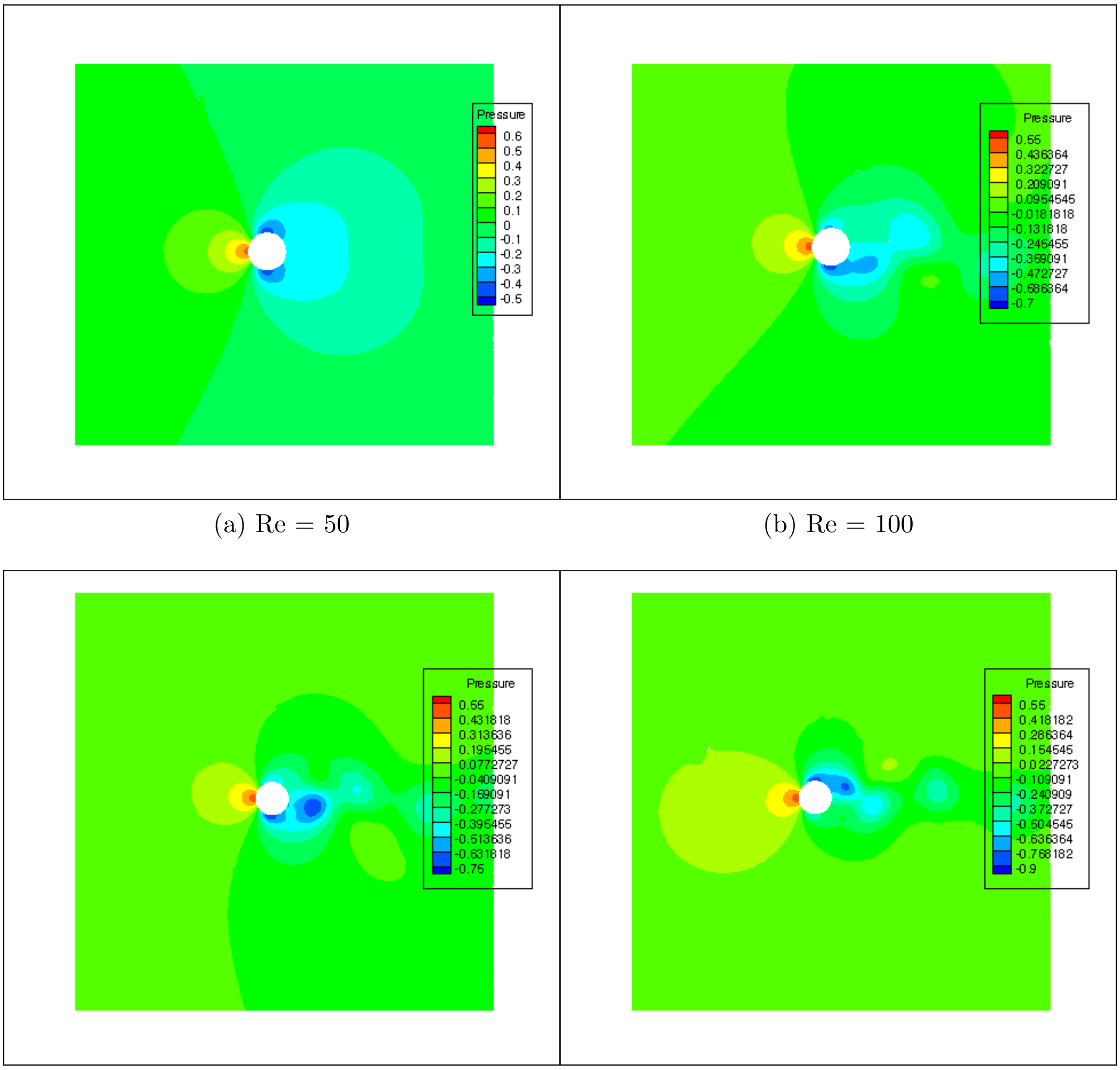


(a) C_l in relation to real time

(b) C_D in relation to real time

Figure 5.7: Comparison of lift and drag coefficients for two different physical time steps at Re = 100, for G3.

Fig. (5.7) shows the pressure field for four different Reynolds numbers and Fig. (5.8) shows the corresponding vorticity field.



(a) Re = 50

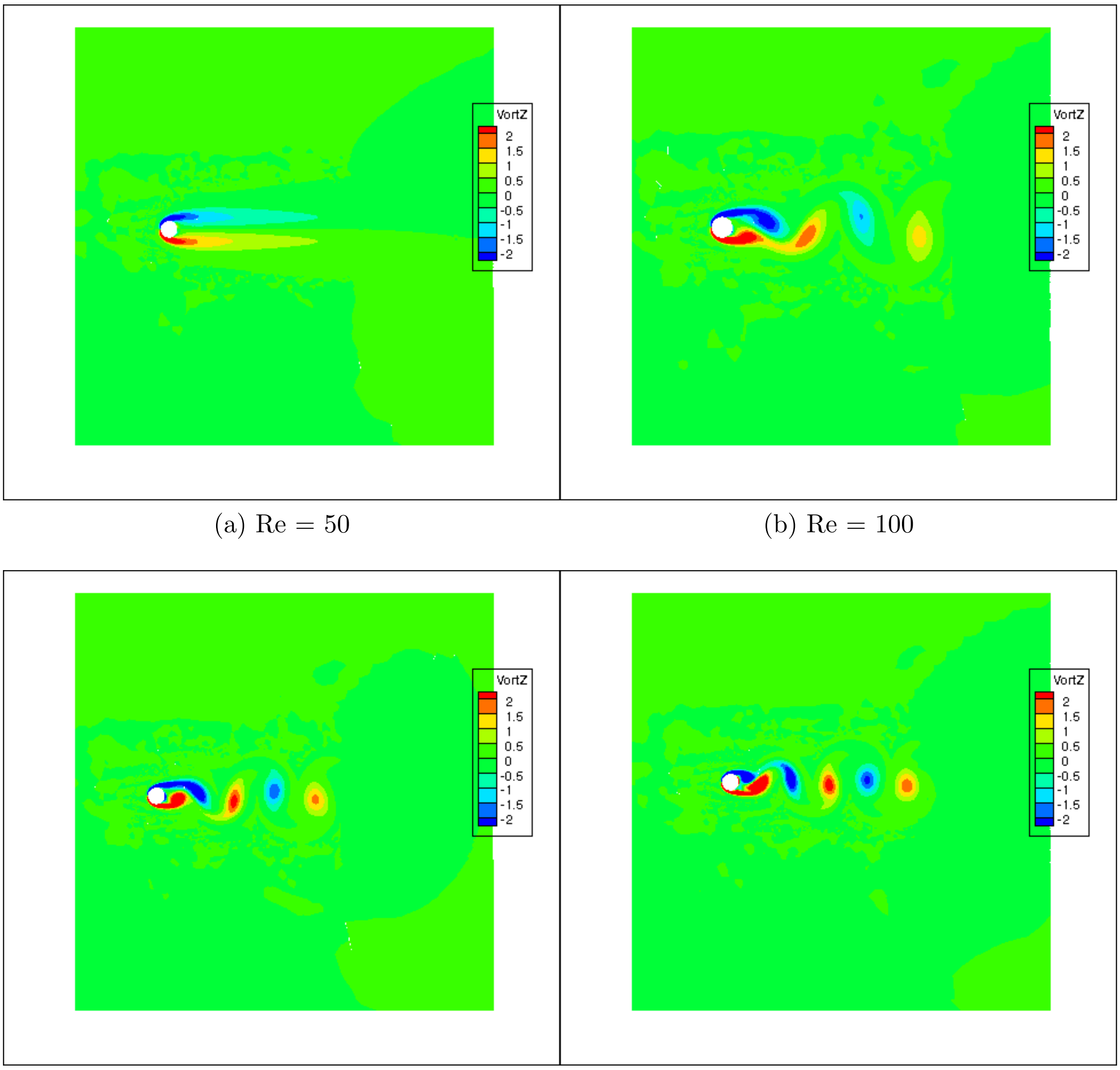
(b) Re = 100

(c) Re = 150

(d) Re = 200

Figure 5.8: Pressure field for different Reynolds numbers.

As we can see in Figure (5.8), the shape of the vortices produced confirms the categorization made by Zdravkovich (fig. 5.3).



(a) Re = 50

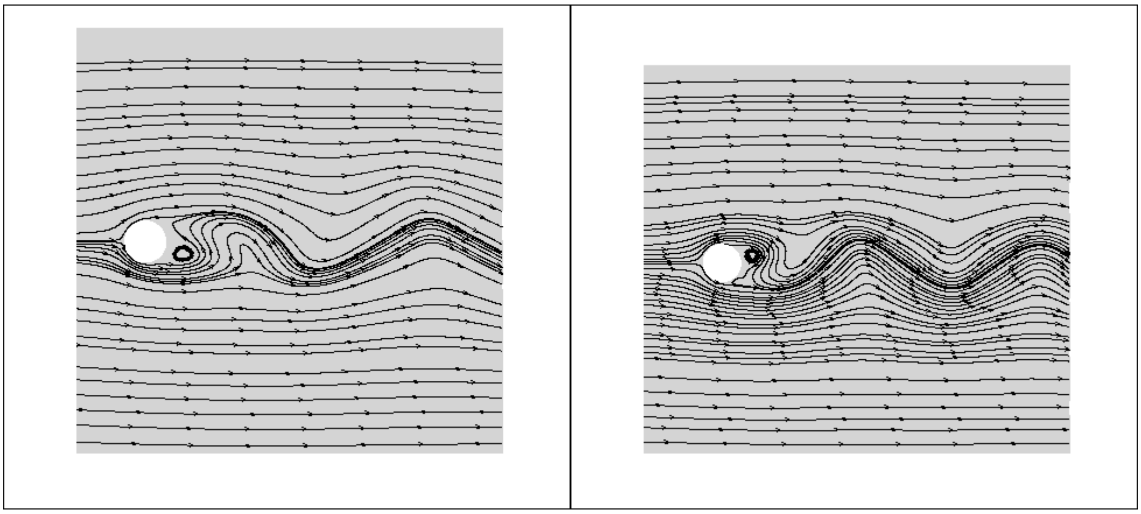
(b) Re = 100

(c) Re = 150

(d) Re = 200

Figure 5.9: Vorticity field for different Reynolds numbers.

Streamlines for Reynolds numbers 100 and 200 confirm the natural phenomena mentioned in the figure 5.3.



(a) Re = 100

(b) Re = 200

Figure 5.10: Flow past a circular cylinder. Streamline visualization.

Chapter 6

Wave propagation over a bar

6.1 Physics of the problem

The second case in which the Immersed Boundary Method developed in this dissertation was examined and tested was the two-dimensional fluid–structure interaction of a wave with a trapezoidal bathymetry. The experiment that inspired us to simulate this physical state was performed by Beji and Battjes [23]. In this initial experiment seven gauges were placed in the wave tank. The experiment was later repeated by Dingemans [24] who added four more gauges. The location of the wave stations are depicted in Figure 6.1. The generated wave profile has height h = 2.0 cm and period T = 2.02 s.

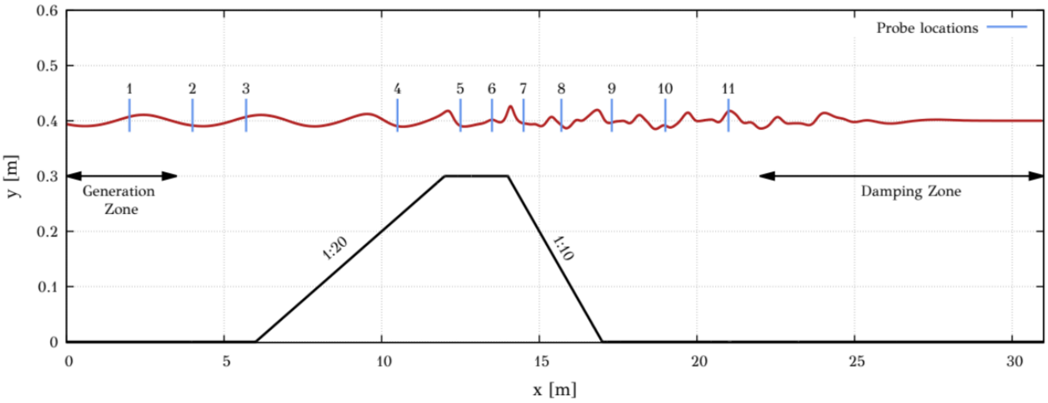


Figure 6.1: Numerical setup of the wave interaction with variable bathymetry test case.

Figure 6.1 reveals additional information about the computational space that was built. The length of the numerical wave tank is 31 m. The damping zone is 9 m and the wave generation zone extends for 3.5 m. In addition, the numerical wave tank depth is d = 0.4 m .

In this dissertation, the numerical generation and absorption of the free surface waves is performed through source terms applied in the momentum equations, in specific zones of the computational domain near the farfield boundaries. Typically, these zones extend for a few wavelengths. The general form of the source terms is given by Equation (6.1). This source term drives the solution to the imposed $\vec{\upsilon}_{tar}$ velocity vector.

$$\vec{S}_{nwt} = C_{nwt}\rho_m(\vec{\upsilon} - \vec{\upsilon}_{tar}) \tag{6.1}$$

The form of the function C_{nwt} is given by the equation (6.2).

$$C_{nwt} = \alpha\frac{exp(x_r^n) - 1}{exp(1) - 1}, \quad x_r = \frac{x_s - x}{x_s - x_e} \tag{6.2}$$

The source term of Equation (6.2) is a function of the non-dimensional space variable x_r, which depends on the starting position x_s of the specified zone and the end point of the zone x_e.

The maximum value of the function is regulated through parameter α, while its spatial distribution through n parameter. The function is zero away from the boundaries of the computational domain, scales exponentially inside the specified zone and it reaches its maximum value α at the boundary. In the present work, the desired numerical solution (e.g., a specific wave profile in the generation zone) is obtained implicitly, meaning that the solver needs to converge to that solution through the numerical procedure.

The wave generation is performed by imposing the velocity $\vec{\upsilon}_{tar}$ as given by an appropriate wave theory. The wave solution is obtained from the semi-analytical method of Fenton's Stream Function Theory [25].

In case of the wave damping zones, the target is to minimize the transverse velocity components.

The selection of the factors related to the generation and damping zone of the wave was made based on the information obtained from other research works. [6]. The Damping Factor α is chosen equal to 60 and the Damping Function Power n is equal to 3.5.

In Figure (6.2) C_{nwt} is plotted for different values of α and n.

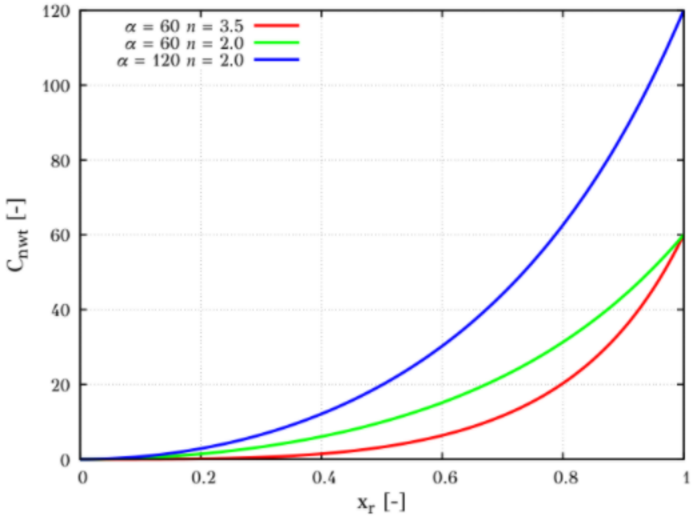


Figure 6.2: The effect of the values α and n in function C_{nwt}.

6.2 Grid and Timestep Indepedence

In this dissertation a timestep and grid sensitivity study was conducted. The timestep values, depending on the period of the wave, which were examined were dt = T/800, T/1000 and T/2000. Regarding the characteristics of the grids used, they are presented in Table 6.1.

Table 6.1: Parameters used for the different grid configurations. L: length of the computational domain, m, H: height of the computational domain, m.

Grids		L	H	Total nodes
G1	2-D	31	0.6	93 714
G2	2-D	31	0.6	177 567
G3	2-D	31	0.6	262 165

In all three grids made their first common feature was that the area that characterizes the air the grid selected is quite sparse and unstructured. In the simulation that was created we did not pay much attention to what is happening in the air.

Then, in all the grids tested, special attention was paid to the area near the free surface where we wanted to achieve the most accurate and close to reality elevation of the free surface due to the wave generation.

Finally, the high density of the grid is selected in the area within the water where the bottom changes since there is a need to see in more detail the changes in the field of speeds and pressures. In the area of wave damping, ie towards the end of the computational space, the grid dilutes considerably.

In the first grid, grid spacing is selected in the area of water along the axis of x equal to 0.02. This applies to the area up to the point where the wave damping begins. From this point on we choose a gradual dilution of the computer grid until its end.

In the second grid, which contains almost twice as many computational cells as in the first, exactly the same thing applies with the difference that the grid spacing from the beginning of the computing space to the beginning of the damping area is equal to $dx = 0.01$.

In the third and last grid that we tested, it was chosen to further thicken the area where the bottom rises with the bar that is formed, since there the phenomena that concern us are more intense and we need greater detail of the solutions.

An indicative image of the area where the bar is formed in our computational grid is shown below (Fig. 6.3). Specifically, a portion of the grid is shown that covers the trapezoidal area of the bottom as well as the area near the free surface above the raised bottom.

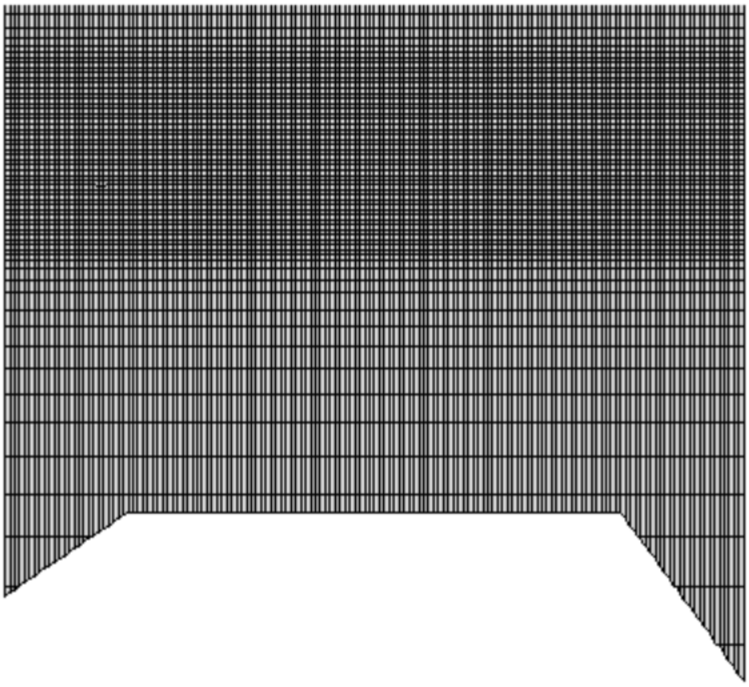
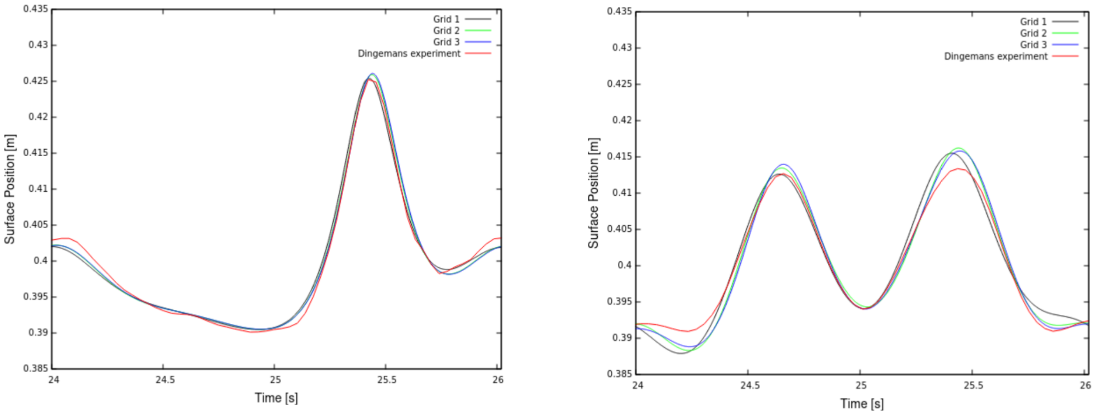


Figure 6.3: Area of the computational grid for the study of the bathymetry.

There is, as we said, the gradual thickening along the y-axis starting from the bottom to the large thickening on the free surface, as well as the uniform thickening along the x-axis.

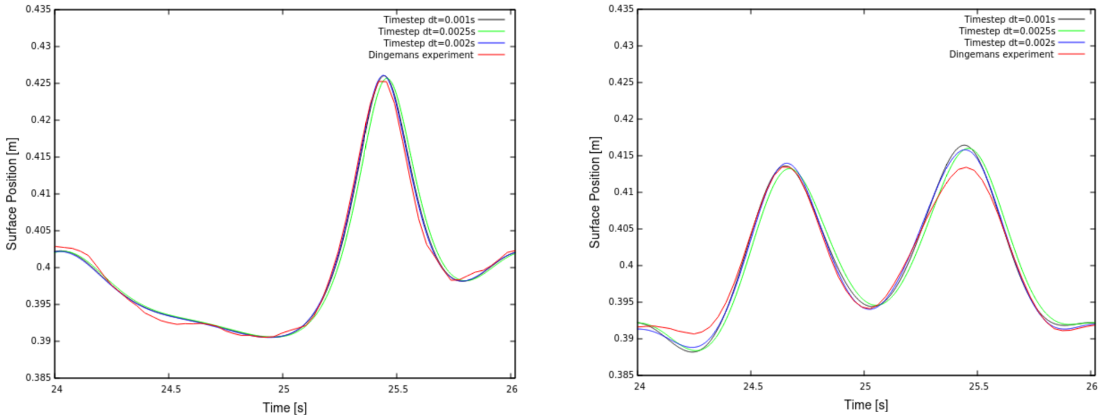
The sensitivity study concluded for the timestep resolution to $dt = 0.002s$, which corresponds to ≈ 1000 steps per period, and to the third computational grid (G3). This choice was made after the convergence of MaPFlow results with Dingemans experiments in these situations. The relevant diagrams are presented below (Fig. 6.4 and Fig. 6.5).



(a) Station 6 at x = 13.5 m

(b) Station 10 at x = 19.0 m

Figure 6.4: Grid independence study. (a) Grid independence at station 6 for a period of time. (b) Grid independence at station 10 for a period of time.



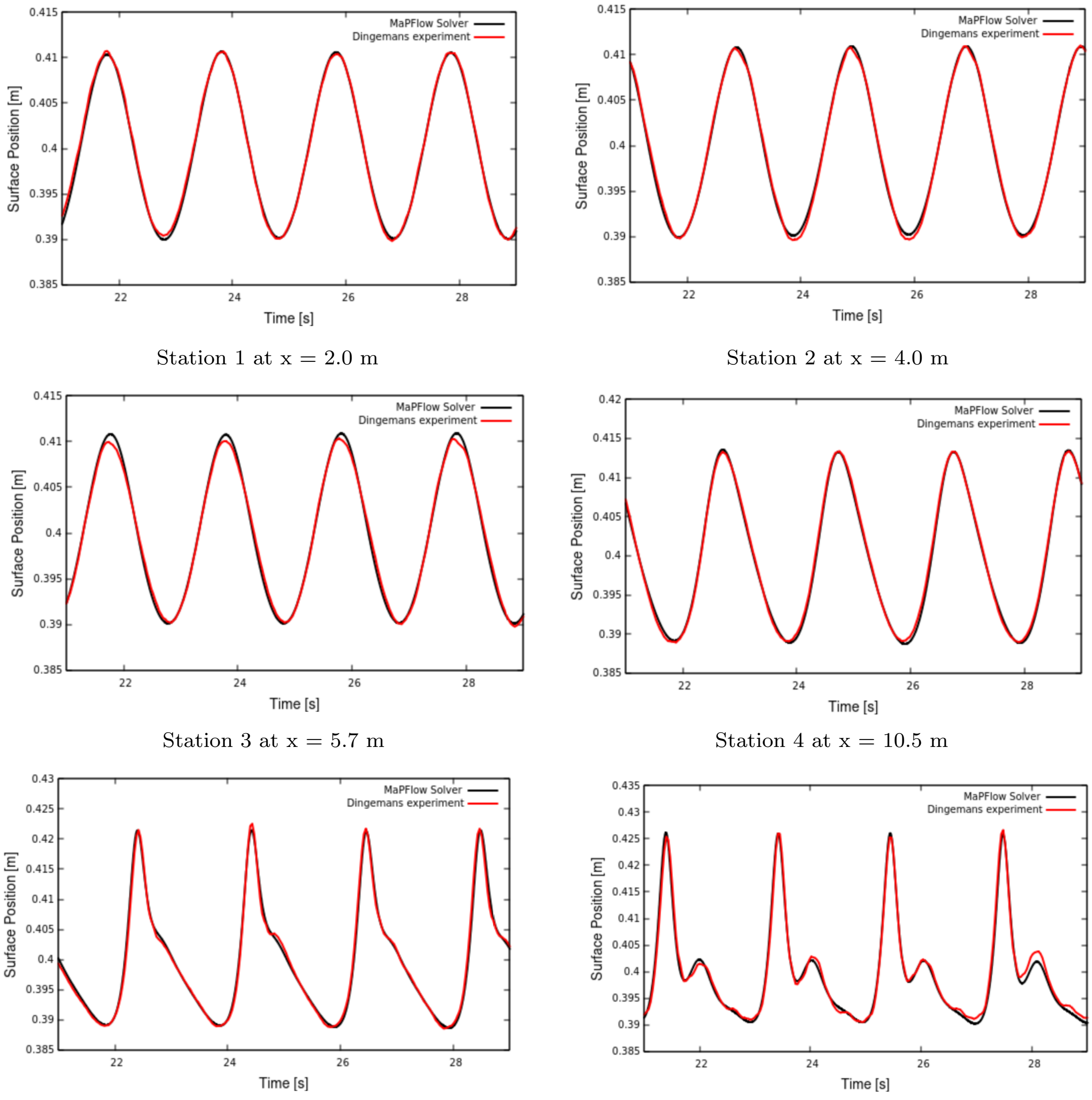
(a) Station 6 at x = 13.5 m

(b) Station 10 at x = 19.0 m

Figure 6.5: Timestep independence study. (a) Timestep independence at station 6 for a period of time. (b) Grid independence at station 10 for a period of time.

6.3 Results

The following (Fig. 6.7) are the free surface positions as obtained from the MaPFlow solver compared to the experimental results performed by Dingemans.



Station 1 at x = 2.0 m

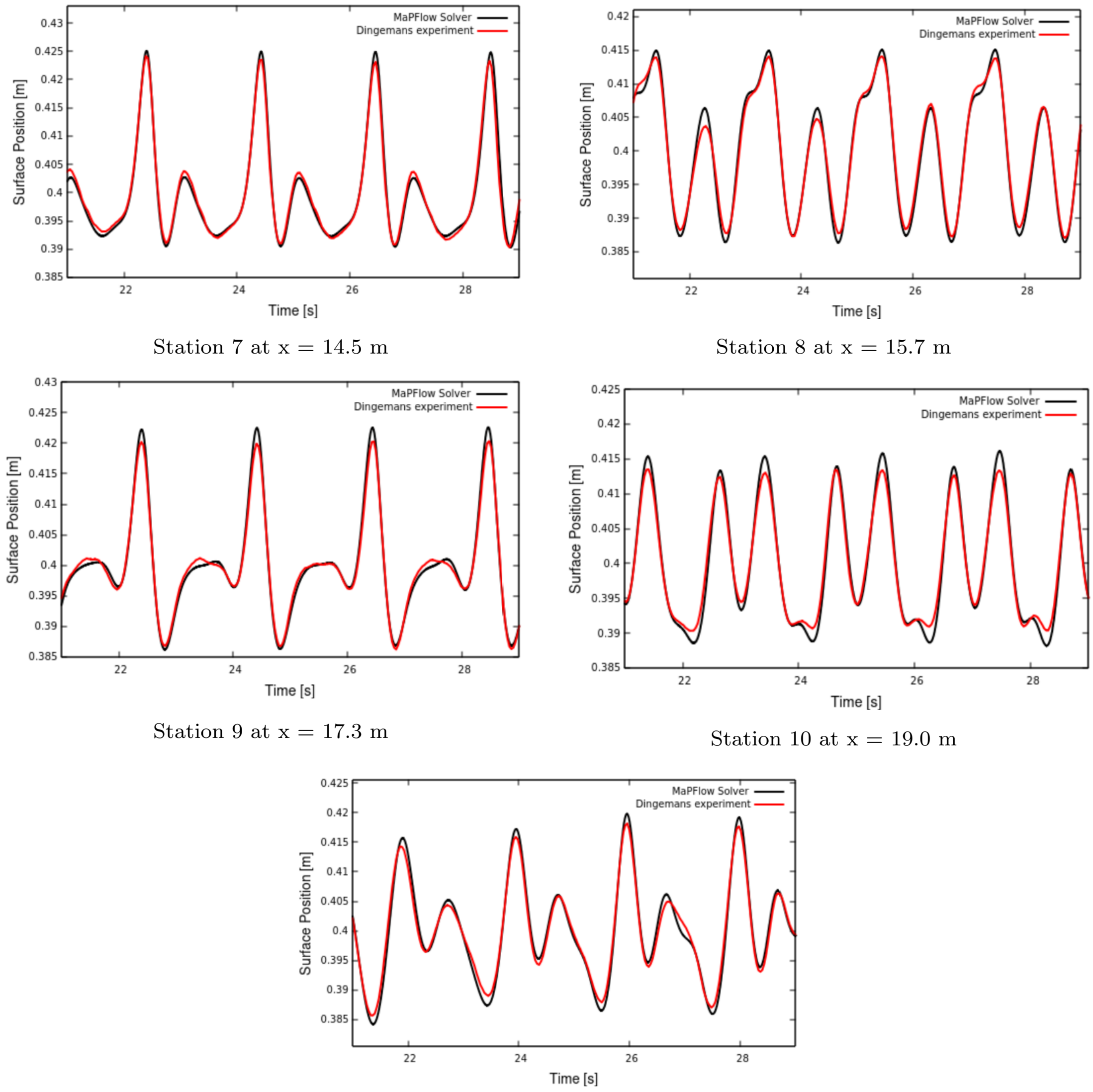
Station 2 at x = 4.0 m

Station 3 at x = 5.7 m

Station 4 at x = 10.5 m

Station 5 at x = 12.5 m

Station 6 at x = 13.5 m



Station 7 at x = 14.5 m

Station 8 at x = 15.7 m

Station 9 at x = 17.3 m

Station 10 at x = 19.0 m

Station 11 at x = 21.0 m

Figure 6.7: Free surface positions at the wave stations. Comparison of the numerical results and experimental data.

6.4 Second examined Bathymetry

To validate the intersecting cell algorithm developed in this dissertation as well as the MaPFlow solver, we studied another different bathymetry inspired by the experiment conducted by T. Ohyama, w. Kioka and A. Tada [26]. Gerassimos A. Athanassoulis and Christos E. Papoutsellis [27] have also conducted a study on this bathymetry, comparing their results with those of the original work.

The shape of the depth meter under study will be as shown in the figure below (Fig. 6.8).

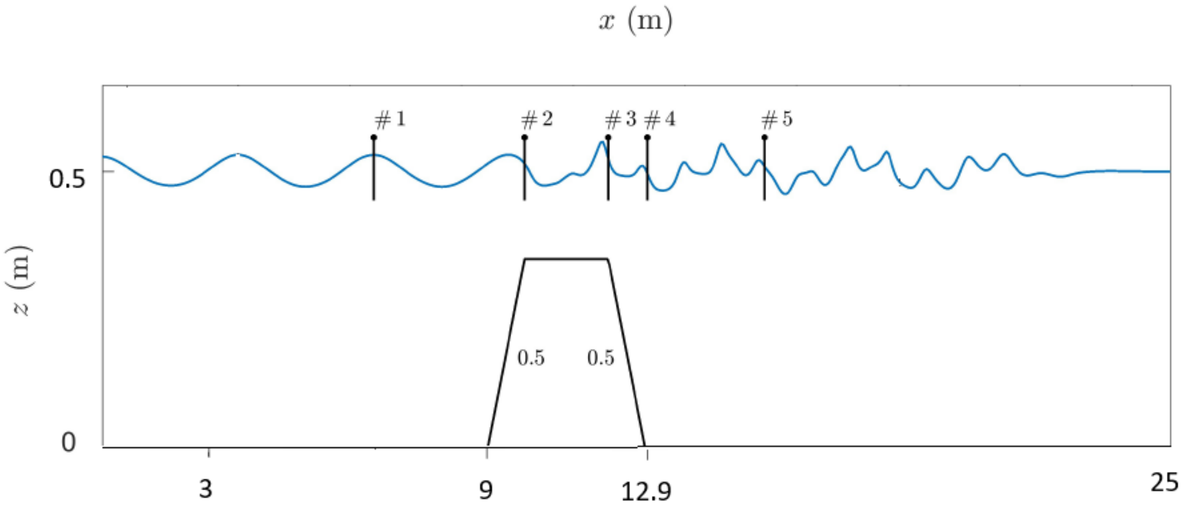


Figure 6.8: Configuration of the numerical wave tank and locations of the measuring stations 1-5 used in (Ohyama et al., 1995).

As shown in Fig. 6.8, the water surface elevations were measured at five different locations-stations.

Six different conditions were investigated in the experiment of (Ohyama et al.): three different wave periods (referred to as ”short”, ”intermediate” and ”long” waves) with two different incident wave heights (referred to as ”smaller” and ”larger” waves).

In the present work, we consider Case 2 of (Ohyama et al., 1995) involving ”short” incident wave conditions, with corresponding height and period (H_0, T_0) given by (0.05,1.341).

The generation zone of the wave for this bathymetry is placed from zero to 3 meters while the damping zone from 16 to 25 meters.

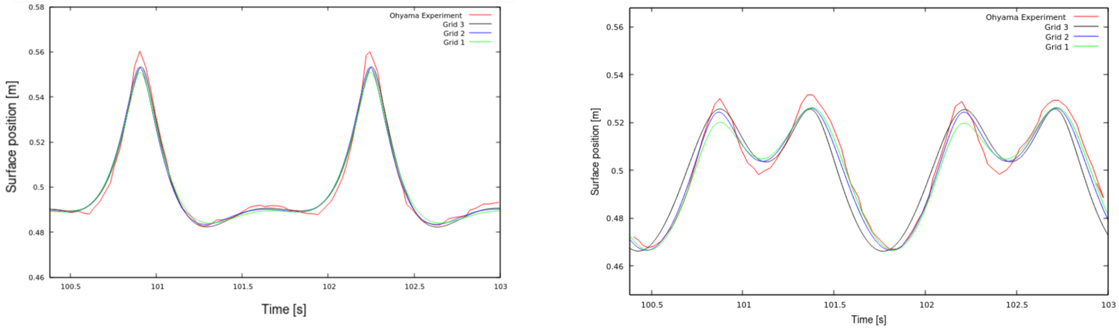
In order to achieve grid independence, three computational grids were tested again, as in the case of the bathymetry of the experiment of Beji and Battjes.

Table 6.2: Parameters used for the different grid configurations. L: length of the computational domain, m, H: height of the computational domain, m.

Grids		L	H	Total nodes
G1	2-D	25	2.5	77 574
G2	2-D	25	2.5	132 003
G3	2-D	25	2.5	249 672

And in these computational grids the density per region follows the rules we mentioned for the case of the first bathymetry.

In figure 6.9, we compare the free surface elevation at stations 3 and 5 with the digitized experimental data (Ohyama et al., 1995).



(a) Station 3 at x = 12.2 m

(b) Station 5 at x = 14.0 m

Figure 6.9: Grid independence study. (a) Grid independence at station 3 for two periods of time. (b) Grid independence at station 5 for two periods of time.

As we observe better convergence to the experimental data is ensured with our third computational grid which is the densest of the three.

Chapter 7

Summary, Conclusions and Suggestions for Future Projects

7.1 Conclusions

The intersecting cell software developed in this dissertation proved to be fully functional after its integration into the CFD solver MaPFlow. Of course during the test of the states we studied the code was modified several times to harmonize each time the computational grid with the geometry of the interfering in the solid body flow. Like any software, it can be further modified and improved to make it more universal.

Both in the study of the laminar flow around a cylinder and for the free surface wave propagation over a variable bottom, measured results were obtained that approached to a very satisfactory degree the data we have from real experiments. This comparison was a very important criterion in order to decide whether the results of the code are valid or not.

In the case of laminar flow around a two-dimensional cylinder, the change in the vorticity is examined depending on the change in the Reynolds coefficient. In addition, we perform a grid independence for a given Reynolds number equal to 100, for which we also quote the values of the resistance and lift coefficients relative to the values available from other literature sources.

In the case of free surface wave propagation, two different bathymetries were examined for completeness, where the free surface position, the trapezoidal bottom shape and the size of the computing space differed. Thus we observe, among other things, the differentiation that exists in the elevations of the free surface depending on the change of the bottom.

7.2 Suggestion for Future Projects

The intersecting cell method, a subcategory of the Immersed Boundary Method developed in this dissertation, could be extended beyond the two applications in which it was tested. This means that in addition to the cylinder and the variable bottom, the algorithm of Cut-Cells could be considered in the case of other classic geometric bodies, such as an airfoil or more complex random geometries.

In this case of course the algorithm would have to undergo some necessary changes to accept the new different geometry since, as it has been pointed out, in the construction of each body the geometric equation that describes its boundary is taken into account.

Another idea that could be implemented with this method could be the simultaneous presence of more than one body in the same computationl domain. Therefore, we could study the interaction that bodies would have with each other in a simulation of a physical state.

An interesting extension of the method could also be the examination of a moving body that would perform periodic and non-periodic movements. In this work the geometries studied were fixed in space. For example, a cylinder connected at one end by a spring could perform a heave motion.

Finally, an important perspective for the development of the algorithm would be if it concerns three-dimensional computational spaces and bodies in addition to two-dimensional. Thus the applications and simulations of the algorithm would have greater validity and value, as well as greater relevance to real physical states and phenomena.

Bibliography

[1] H K Versteeg and W Malalasekera. *Introduction to computational fluid dynamics.* Vol. 43. 2006.

[2] Hermann Schlichting. *HermannSchlichting-Boundary-layertheory-McGraw-Hill1979.* 1979.

[3] George Papadakis. *Development of a hybrid compressible vortex particle method and application to external problems including helicopter flows.* 2014.

[4] D.Drikakis and W.Rider. *The Artificial Compressibility Method.* High-Resolution Methods for Incompressible and Low-Speed Flows. Springer, Berlin, Heidelberg, 2005. ISBN: 978-3-540-26454-5.

[5] Joel H.Ferziger and Yu-Heng Tseng. "A ghost-cell immersed boundary method for flow in complex geometry". In: *Journal of Computational Physics* 192.2 (2003), pp. 593–623. DOI: `https://doi.org/10.1016/j.jcp.2003.07.024`.

[6] Dimitris Ntouras and George Papadakis. "A Coupled Artificial Compressibility Method for Free Surface Flows". In: *Journal of Marine Science and Engineering* 8.8 (2020), p. 590. DOI: `https://doi.org/10.3390/jmse8080590`.

[7] P. L. Roe. "Approximate Riemann solvers, parameter vectors, and difference schemes". In: *Elsevier, Journal of Computational Physics* 43.2 (1981), pp. 357–372. DOI: `https://doi.org/10.1016/0021-9991(81)90128-5`.

[8] Patrick Queutey and Michel Visonneau. "An interface capturing method for free-surface hydrodynamic flows". In: *Elsevier, Computer Fluids* 36.9 (2007), pp. 1481–1510. DOI: `https://doi.org/10.1016/j.compfluid.2006.11.007`.

[9] M. Darwish and F. Moukalled. "Convective Schemes for Capturing Interfaces of Free-Surface Flows on Unstructured Grids". In: *An International Journal of Computation and Methodology* 49.1 (2005), pp. 19–42.

[10] Hyung TaekAhn and Yannis Kallinderis. "Strongly coupled flow/structure interactions with a geometrically conservative ALE scheme on general hybrid meshes". In: *Elsevier, Journal of Computational Physics* 219.2 (2006), pp. 671–696. DOI: `https://doi.org/10.1016/j.jcp.2006.04.011`.

[11] Charles S. Peskin. "Flow patterns around heart valves: A numerical method". In: *Elsevier* 10.2 (1972), pp. 252–271. DOI: `https://doi.org/10.1016/0021-9991(72)90065-4`.

[12] Henry Bandringa. *Immersed boundary methods.* University of Groningen,Institute of Mathematics and Computing Science, Master Thesis in Applied Mathematics. 2010.

[13] Charles S. Peskin. “The immersed boundary method”. In: *Cambridge University Press* 11 (2002), pp. 479–517. DOI: https://doi.org/10.1017/S0962492902000077.

[14] L.Sirovich D.Goldstein R.Handler. “Modeling a No-Slip Flow Boundary with an External Force Field”. In: *Elsevier, Journal of Computational Physics* 105.2 (1993), pp. 354–366. DOI: https://doi.org/10.1006/jcph.1993.1081.

[15] J.Périaux R.Glowinski T.W.Pan. “Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies”. In: *Elsevier, Computer Methods in Applied Mechanics and Engineering* 151.1-2 (1998), pp. 181–194. DOI: https://doi.org/10.1016/S0045-7825(97)00116-3.

[16] Long Lee and Randall J. LeVeque. “An Immersed Interface Method for Incompressible Navier-Stokes Equations”. In: *SIAM Journal on Scientific Computing* 25.3 (2003), pp. 832–856. DOI: https://doi.org/10.1137/S1064827502414060.

[17] E.A.Fadlun R.Verzicco P.Orlandi J.Mohd-Yusof. “Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations”. In: *Journal of Computational Physics* 161.1 (2000), pp. 36–60. DOI: https://doi.org/10.1006/jcph.2000.6484.

[18] Sekhar Majumdar B.N. Rajani A. Kandasamy. “Numerical simulation of laminar flow past a circular cylinder”. In: *Elsevier* 33 (2008), pp. 1228–1247. DOI: https://doi.org/10.1016/j.apm.2008.01.017.

[19] Jordi Casacuberta Puig. *Bidimensional laminar flow around a circular cylinder.* URL: http://files.the-foam-house5.webnode.es/200000362-1646b173f4/Chapter3_Cylinder.pdf.

[20] Raj P. Chhabra Vijaya K. Patnana Ram P.Bharti. “Two-dimensional unsteady flow of power-law fluids over a cylinder”. In: *Elsevier* 64.12 (2009), pp. 2978–2999. DOI: https://doi.org/10.1016/j.ces.2009.03.029.

[21] M. Zdravkovich. *Flow Around Circular Cylinders, vol. 1, Oxford Science Publication.* 1997.

[22] C.H.K. Williamson. “VORTEX DYNAMICS IN THE CYLINDER WAKE”. In: *Annu. Rev. Fluid. Mech.* (1996), 28:477–539.

[23] S. Beji and J.A. Battjes. “Experimental investigation of wave propagation over a bar”. In: *Elsevier Science Publishers B.V., Amsterdam* 19 (1993), pp. 151–162.

[24] M. Dingemans. “Comparison of Computations With Boussinesq-Like Models And Laboratory;” in: *Technical Report,Delft Hydralics: Delft, The Netherlands* (1993).

[25] J.D. FENTON. “THE NUMERICAL SOLUTION OF STEADY WATER WAVE PROBLEMS ”. In: *Department of Civil Engineering, University of Auckland, Private Bag, Auckland, New Zealand* 14.3 (1988), pp. 357–368.

[26] Akihide Tada Takumi Ohyama Wataru Kioka. “Applicability of numerical models to nonlinear dispersive waves”. In: *Elsevier, Coastal Engineering* 24 (1995), pp. 297–313. DOI: https://doi.org/10.1016/0378-3839(94)00033-T.

[27] G.A. Athanassoulis Ch.E. Papoutsellis. “A new efficient Hamiltonian approach to the nonlinear water-wave problem over arbitrary bathymetry”. In: *ResearchGate* (2017).