TÉCNICO LISBOA

INSTITUTO SUPERIOR TÉCNICO

APPLICATION OF IMMERSED-BOUNDARY METHOD IN HEAT-TRANSFER PROBLEMS

RENATO MATOS SALGADO

THESIS TO OBTAIN THE MASTER OF SCIENCE DEGREE IN

INTEGRATED MASTER IN AEROSPACE ENGINEERING

SUPERVISORS

PROFESSOR DUARTE MANUEL SALVADOR FREIRE SILVA DE ALBUQUERQUE

EXAMINATION COMMITTEE

CHAIRPERSON: PROF. AFZAL SULEMAN

SUPERVISOR: DR. DUARTE MANUEL SALVADOR FREIRE SILVA DE ALBUQUERQUE

MEMBER OF THE COMMITTEE: PROF. GONÇALO NUNO GUERREIRO DE JESUS SILVA

May 2022

Declaração de Originalidade

Declaro que o presente documento é um trabalho original da minha autoria e que cumpre todos os requisitos do Código de Conduta e Boas Práticas da Universidade de Lisboa.

Declaration of Originality

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgements

I would like to express my sincere thanks to my supervisor Professor Duarte Albuquerque whose knowledge, expertise and experience were essential for this project. For all your patience, availability and dedication, my sincere thanks.

On a more personal note, I would like to thank my family for their unconditional help and care.

Dad, Mom, thank you for supporting me and providing me the opportunity to study and grow. Mariana, thank you for your companionship.

Additionally, a special thanks to my older sister Cátia Salgado and Joaquim Viegas for their assistance and advice on thesis matters.

Similarly, my profound gratitude to my girlfriend Rita Boto Pereira for all the unconditional love, support, and patience. Thank you for always believing in me.

Equally, an important and final thanks to all my good friends who accompanied me throughout these difficult college years and made them enjoyable.

Resumo

Este trabalho tem como objetivo estudar simulações térmicas usando o método de fronteira imersa no software *SOL*. Particularmente, na interpolação de variáveis térmicas relevantes da fronteira sólida para a fronteira IB. Dois métodos de interpolação são analisados, tendo capacidade de impor condições de fronteira de temperatura e de gradiente térmico. Os métodos são verificados para problemas de Taylor-Couette em 2D para baixo Reynolds e com transferência de calor, enquanto a sua ordem de precisão é verificada. Um estudo de robustez a alterações de malha é realizado em ambos os métodos implementados utilizando malhas estruturadas triangulares, quadriláteras e hexagonais. Os métodos são também testados usando malhas híbridas onde se observa a robustez do método quadrático desenvolvido. Esta metodologia verifica a inabilidade do método linear em lidar com malhas de conectividade arbitrária. O método de interpolação quadrática foi também usado para simular o escoamento sobre um cilindro com transferência de calor, estudando-se a amostragem de número de Nusselt ao longo da superfície sólida. Após melhoramentos ao método Neumann foram obtidos resultados concordantes com dados da literatura e com dados obtidos em simulações body fit desenvolvidas no *SOL*.

De forma geral, este trabalho verificou o software *SOL* para escoamentos de Taylor-Couette com transferência de calor através de um método de fronteira imersa com precisão de segunda ordem, quer para condições de fronteira de Dirichlet quer de Neumann. Adicionalmente, provou-se capaz de resolver problemas mais complexos de transferência de calor em fronteira imersa com valores de Nusselt comparáveis à literatura.

Palavras-chave: Método de Fronteira Imersa, Mecânica dos Fluidos Computacional, Transferência de Calor, Métodos de interpolação, Ordem de Precisão

Abstract

This work focuses on the thermal simulation capabilities of *SOL* using an immersed boundary method. In particular, on the interpolation of the relevant variables from the original solid boundary to the IB one. Two schemes for interpolation are studied, capable of imposing both temperature and heat flux temperature boundary conditions. The schemes are then verified for low-Reynolds 2D Taylor-Couette flow problems with heat transfer, while also having their theoretical order of accuracy verified. A mesh robustness study is conducted across both schemes and boundary conditions, using triangular, quadrilateral, and hexagonal meshes verifying full capability at handling these grid topologies. The methods are also employed in simulations using hybrid meshes, where the least squares interpolation method capabilities are demonstrated. This methodology proves that the linear method lacks the robustness necessary to handle grids with arbitrary connectivity. The least squares interpolation method was also used to simulate flow over a cylinder with heat transfer, employing sampling of Nusselt numbers across the surface of the solid boundary. After the introduction of enhancements to the Neumann interpolation method, the achieved results were concordant with the relevant literature and with the body fit simulations developed in *SOL* as a benchmark.

Overall, this work verifies *SOL* for heat transfer Taylor-Couette flow simulations using the immersed boundary method for up to second order both for Dirichlet and Neumann boundary conditions. Furthermore, it proved capable of solving more complex boundary heat transfer problems, with a Nusselt number evolution comparable with the literature.

Keywords: IB-Method, CFD, Heat Transfer, Interpolation Methods, Order of Accuracy

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations and Acronyms

2D	Two-Dimensional
3D	Three-Dimensional
BC	Boundary Condition
CAD	Computer Assisted Design
CFD	Computational Fluid Dynamics
IB	Immersed-Boundary
IBM	Immersed-Boundary Method
NSE	Navier-Stokes Equations
PISO	Pressure-Implicit with Splitting of Operators
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
TC	Tangential Correction

Greek Symbols

α	Thermal Diffusivity
β	Least Squared Error Coefficient
η	Distance ratio
Γ_ϕ	Scalar diffusivity of ϕ
λ	Heat conductivity coefficient
ν	Kinematic viscosity
Ω	Finite Volume
ϕ	Transport variable
ρ	Fluid density
θ	Nusselt Sampling Angle
$\|\|\epsilon_i\|\|$	Mean Error for variable i
$\|\|\epsilon_i\|\|_{MAX}$	Maximum Error for variable i

Latin Symbols

C_V	Volumetric heat capacity
$Cell_{area}$	Cell area
d	Distance
f	Face centre of the cell

g	Gravity acceleration
h	Characteristic Length
m	Number of Solid points in Stencil
n	Number of Fluid cell centres in Stencil
Nu	Nusselt Number
P	Centroid of the cell
p	Fluid Pressure
Pr	Prandtl Number
Q	Volumetric heat source term
q	Heat Flux
q''_{IB}	Heat Flux at the Immersed Boundary
R	Radius
Re	Reynolds Number
S	Surface
S_f	Face surface vector
S_ϕ	Transport equation source term
T	Temperature
t	Time
T_{IB}	Temperature at the Immersed Boundary
T_{outer}	Temperature condition at the boundary outer wall
u	Fluid velocity
u_θ	Angular velocity
V	Volume

1 Introduction

1.1 Motivation

Whether it is necessary to describe the behavior of an atmospheric entry vehicle, to design a heat exchanger or even to evaluate a biological system, engineers are faced with a common issue: fluids. Fluids comprise liquids, gases, and plasmas, and thus is relevant in a variety of fields ranging from mechanical, aerospace, chemical, civil, and biomedical engineering to meteorology, biology, or geophysics. The branch of physics responsible for the study and evaluation of fluids is called fluid mechanics. [1, 2]

These studies can be related to fluids at rest – in the case of fluid statics – or to the effect of forces on fluids in motion – fluid dynamics. Both are active fields of research; however, fluid dynamics takes a major role due to its higher applicability despite its increased complexity. Even though it is a field of study of recognized importance with multiple historical advancements over the years, most fluids' related problems are partly or wholly unsolved. Numerical methods and empirical equations and simplifications are often used to make these problems solvable. [1]

Due to the nature of fluid mechanics, problems related to fluid flow will satisfy a set of well-documented and modelled laws. However, from a practical point of view, approaching a real-life problem considering the full range of fluid behaviour known mostly leads to unsolvable situations. The main objections to the complete theoretical treatment are geometry and viscosity. Complex geometries and changes in the viscosity of the fluids lead to complex problems that can be solved using numerical analysis and data structure, thus being solved with computational fluid dynamics (CFD). [1]

1.2 Computational Fluid Dynamics

CFD started to be used in the late 50's motivated by the design of aerospace vehicles. The design and development of these vehicles is hardly conditioned by the harsh and different conditions to which they are exposed. This implies that sometimes most of the conditions cannot be replicated or, if they can, the cost is extremely high. To counter these obstacles, CFD surged as an option to simulate the behavior of the reentering the atmosphere. This not only allowed for the reduction of the price of the experiments but also a broader set of conditions could be explored and tested. Nowadays, these two arguments are still valid for all the areas stated above. Whether it is the aerodynamics of a race car or the behavior of a chemical reactor, simulations using CFD can amplify the spectrum of possibilities while decreasing the price of the development, drastically. [2, 3]

In the last decades, CFD has become widely popular in multiple areas of study. Figure 1.1 shows the evolution of papers issued by decade that used CFD methodologies. A growing processing and speed capacity of computers aligned with the above-mentioned reasons are positioning CFD in the spotlight in science and engineering works. [2, 3, 4]

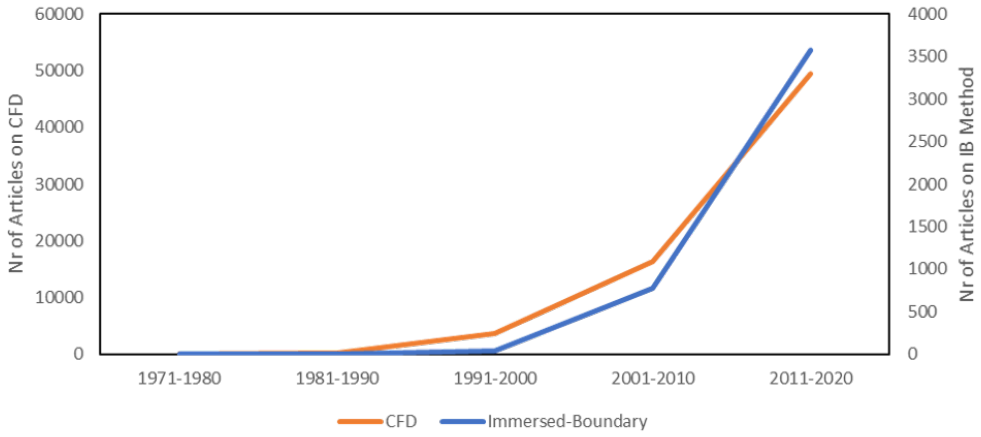


Figure 1.1: *Evolution of CFD and IB method dedicated papers published in the literature over decades. [4]*

The fundamental basis for fluid dynamics and thus, CFD, is the Navier-Stokes Equations (NSE). These constitute a set of partial-differential equations that are responsible for the description of a fluid in motion. To this day, no analytical solution has yet been found to the NSE, leading to the need for numerical methods, simplifications and data structures already mentioned.

Solving a fluid dynamics problem, independently of its nature, often follows an established methodology. This encompasses three distinct procedures: preprocessing, simulation and postprocessing (Figure 1.2).

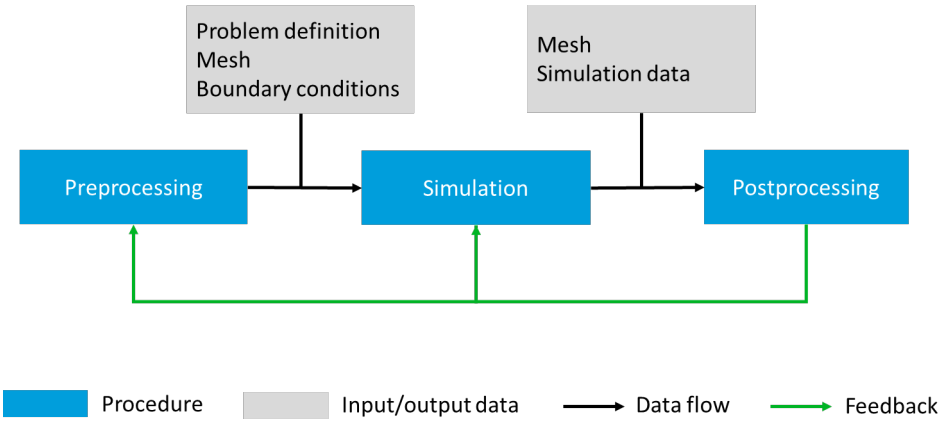


Figure 1.2: *Flowchart of CFD simulation methodology.*

Preprocessing is often considered to be the step with the most influence, especially considering it affects both the simulation and postprocessing. Here the geometry and the physical boundaries of the problem are defined and often a computer aided design is used. From here, the fluid domain is selected and divided into discrete cells. The complete set of discrete cells forms the mesh. The mesh can be uniform or non-uniform and the cells can take innumerous shapes (triangular, squared, hexagonal, polyhedric, variable, etc.). Once the mesh is defined, a model based on the NSE and other possible equations (e.g. enthalpy, entropy, chemical reaction or kinetics) is derived. Lastly, the boundary conditions are defined and chosen to meet the physical problem. Thus, the preprocessing is finished making it possible to proceed with the simulation.

During the simulation the derived model is solved iteratively, one time step at a time in the case of transient simulations. The simulation is responsible for recording all data considered relevant, which will then be used in the postprocessing. These results can be organized into plots or animations to allow

for a visual representation, they can be used for further calculations, and they can become the initial conditions, or defining parameters, for a new fluid dynamics problem.

A necessary step in the preprocessing of every CFD problem is the generation of an appropriate mesh for the domain. This process is of extreme importance and is called meshing. Smaller cells imply denser meshes and more accuracy of the results but also lead to prolonged computations during the simulation process. The fact that the fluid domain is being divided means that space is being discretized. This discretization of space aligned with the discretization of time – possible due to the iteration process during simulation – allow for the solution of the defined model.

There are several discretization methods, both for space and time. The methods used in this project will be detailed further ahead in this document. However, there is one methodology that is to be highlighted and explored throughout this work: the Immersed Boundary (IB) method. The IB method is not a discretization scheme but instead it aims to remove the need to discretize the domain several times over the course of a single transient simulation. For complex geometries or bodies in motion/deformation, the meshing process is time-consuming as it implies complex meshes to describe the boundary of the body or remeshing as bodies change their position/shape, respectively. By using the IB method, the mesh used is fixed and the body can move freely and change its shape with the fluid domain, without the need for remeshing at each considered point in time (time-step). [5]

This method was introduced in 1972 by Charles Peskin when trying to simulate flow patterns around heart valves. It was already known that the living heart suffers contraction and distention while working, thus changing its shape over time. Peskin's main objective was to evaluate the blood patterns without having to redefine the mesh several times. [5, 6, 7]

From here, many scientists started to implement this IB method on their simulations and it has been used in many different fields and with varied applications. The implementation of the IB method is becoming more and more popular and it has accompanied the evolution of CFD popularity. However, in the last years, the growth of the IB method in the literature surpassed the CFD growth, meaning that there is an increasingly tendency to solve CFD problems using IB methods. This tendency is again illustrated in Figure 1.1. [4]

Following the use of CFD over many different applications, the IB method is also starting to be applied in multiple disciplines. It started with a biology problem, but nowadays it is used in aerodynamic studies of vehicles (cars, airplanes etc), wind turbines and the influence of the wind in some terrains, impellers performance on stirring tanks, evaluation of multi-phase systems, chemical reactors, parachutes, and wings evaluation and many more. [8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]

Considering the wings of an airplane as an example. During flight, the load on the wings is such that they deform elastically, this process being a major point of study in aerodynamics. With CFD heat transfer conjugated problems using IB method, it is possible to evaluate the performance of the wings both mechanically and thermally taking into consideration the deformation at that instant in time, without the need for remeshing at each time-step. For a simulation of this magnitude of complexity the ability to mesh only once cannot be overvalued, being responsible for massive improvements in overall simulation time and removing the need to interpolate values from one mesh to another, which would be required if not for the immersed boundary method. The benefits of IB method with thermal calculation capabilities can also be applied in spaceships, chemical reactors, wind turbines or ships. Any system that currently requires remeshing over time due to time-variant interfaces of fluid-solid boundaries with heat transfer can theoretically be improved. [5, 8, 22]

1.3 Objectives

From the examples given above, it is clear that the great majority of the problems that use IB methods are already quite complex. The present work focuses on the implementation of IB methods to study mechanisms of heat transfer within a domain with a moving body. Currently, the scientific community is leaning towards the resolution of conjugated problems of heat transfer with IB method implementation. With that, complex body movement and shape can be evaluated together with heat transfer mechanisms, opening the door for the simulation of more complex systems. The implementation of these mechanisms can dictate several design specifications, such as how a given material will behave under certain fluid flow at a given temperature. [5, 8, 22]

Following the tendencies of the CFD branch, the main objective of the present work is to explore and implement IB methods for conjugated heat problems, in particular to enhance and improve the IB method functionality of a pre-existing CFD software developed by LASEF: SOL. This thesis presents several verifications, tests, evaluations and implementations that contributed to one major aspect: new thermal capabilities for the IB method of SOL. A description of the software functionalities as well as its capabilities is featured in the following chapters. Published work developed using SOL can be seen in [23, 24, 25].

1.4 Contributions to *SOL*

Both the linear interpolation methods and the least squares interpolation methods presented in this work were already implemented in SOL when the work presented in this document began. No documentation about these methods was available. After thorough testing and exploration of the available code, the methods were found to be analogous to those implemented by Jonas Pereira in [23] for the velocity interpolation. Chapter 3 documents and explains the methodology implemented in these thermal interpolation methods.

Code for comparing the analytical solution in a Taylor-Couette flow problem with the temperature field obtained in SOL had already been implemented when the work presented in this documented began, however it was only functional for unitary Dirichlet conditions. This was improved to be functional with any Dirichlet or Neumann boundary condition, and for varying mesh geometries, as shown in Chapter 4.

The least squares interpolation methods for Neumann boundary conditions available in *SOL* failed to produce the expected results in more complex physical problems, as will be shown in Chapter 5. As such, Chapter 6 details the implementation of two separate compensation methods implemented in *SOL* to overcome this limitation. These methods are implemented with thorough comments in the code to allow for ease of understanding.

The *SOL* function responsible for calculating the Nusselt number at the IB boundary was extended to also allow for Neumann boundary conditions.

A function that calculates the Nusselt number for every face in a Body Fit boundary was also implemented in *SOL* to facilitate the comparison between simulations utilizing this method and those that use the IB method, as presented in Chapters 5 and 6.

1.5 Thesis Outline

Throughout the next chapter, Chapter 2, a brief description on the theory behind all the above stated topics is exposed, in particular: fluid dynamics, CFD and its methods with a special focus on IB method and the CFD software used - SOL.

Chapter 3 firstly exposes interpolation methods implemented for thermal values at the immersed boundary. Secondly, it exposes mesh refinement studies for each interpolation scheme, providing an initial verification of the current implementation of the immersed boundary method on *SOL* using Taylor Couette, 2D coannular cylinders, simulations.

In Chapter 4 the same interpolation methods are then subjected to different mesh types and their robustness and sensitivity to mesh quality and type is studied and discussed.

Chapter 5 presents the study of the developed methods in a flow over a heated cylinder problem and compares with both literature data and body fit simulations.

Chapter 6 shows the alterations performed on the Neumann interpolation method, while continuing the simulations of flow over a heated cylinder, until results concordant with the literature are achieved.

Chapter 7 draws the main conclusion on the present work and some possible ideas for future work.

2 Literature Review

The present chapter contextualizes and explains the fundamentals and the concepts needed for the development of this project. This literature review will focus on the state-of-art on CFD and IB Method.

The first part includes a brief presentation of the pre-existing work developed related to the project scope. Then, a theory exposure is presented for a better understanding of the chapters that follow. Lastly, a brief description of the software *SOL* is shown.

2.1 State of Art

The development of numerical algorithms and computer capacity has been growing rapidly in the last decades. With this, advanced numerical simulation has become a promising tool to study and understand fluid dynamics. As mentioned previously, computational fluid dynamics is becoming more and more popular among several scientific areas. CFD is now trending both in academia and industry. In fact, it is a growing billion-dollar business responsible for a rise to a large number of employment opportunities. [2]

Nowadays, there is a lot of information available on CFD in several different fields of application. Some examples of these applications were already presented in Chapter 1. Other examples include the evaluation of the aerodynamics of a car, design of aircraft wings or evaluation of the performance of hot-air balloons. Detailed information about these works can be found elsewhere. [26, 27, 28, 29, 30, 31, 32]

Each day more industries and engineers rely on CFD for analysis, optimization, design, performance evaluation, among others. Its use has become so widespread that it is even used in animation films and movies [33]. Trends show that CFD utilization will continue to grow and become an even more powerful tool.

As the complexity of the problems and geometries to be evaluated increases, different methods to speed and simplify the solution process arise. The scope of the present work is mainly focus in the evaluation of one of these methods: the IB Method. As it was observed in the last chapter, the popularity of this method has accompanied the trends of CFD. In fact, in the last years, more and more problems are being solved with resort to this methodology. And, just like CFD, the nature of the type of problems couldn't be more varied. Ranging from the analysis of the behaviour of fins and wings in nature, to the evaluation of transonic turbulent flow, or airfoils and nuclear reactors. [34, 35, 36, 37, 38, 39]

As CFD gains interest - as it can save a lot of time, money and effort – the availability of different commercial software is also growing. Nowadays, an assortment of CFD software is available ranging from open source to licenced. Some examples include *OpenFoam*, *Ansys*, *Fluent*, *Star CCM+*, *Autodesk CFD* or *SimFlow*. CFD became so popular that even *MATLAB* decided to include the *CFDTool*. *OpenFoam* is an open source software and it's considered one of the most popular among the CFD community. While some of these software are already equipped with tools for pre-processing (drawing and meshing tools) and for post-processing (plotting, animation, data visualization), there are also some programmes to aid in these particular steps. For meshing, *Gmsh* or *Gambit* are some examples. *Tecplot* and *ParaView* are some of the most used programmes for post processing.

For the purpose of this work, the CFD software used was *SOL*, which is an internal CFD software under development in *LASEF* – a CFD laboratory in Instituto Superior Técnico. Other software and tools were also used for this project, namely *OpenFoam*, *Gmsh*, *Texplot*, *SimFlow* and *MATLAB*.

2.2 Numerical Methods

The main focus of this work is to evaluate the performance and enhance the IB Method already implemented in *SOL*. The IB Method is a method that is responsible for solving the transport phenomena at the interface between the fluid and the solid.

The purpose of this section is to provide the reader a brief explanation on the main equations and numerical methods used in this work. To note that these are some of the possible methods available in *SOL*.

2.2.1 Types of mesh

To solve a CFD problem, a step of the pre-processing is required. During this step, the computer aided design (CAD) is performed in accordance with the geometry of the system to be studied. Afterwards, the meshing process takes place. During this step, the geometric defined domain is represented by smaller discrete cells over which the partial differential equations can be solved. [40]

The process of creating a mesh with good quality and good suitability to the problem to be analysed is a field of study by itself. This process requires a lot of fine tuning of the parameters of the grid to the physical problem considered. To note that the solution of the governing equations within several cells is what is used to approximate the solution of the overall domain. A high-quality mesh is the one that provides the most accurate solution in the shortest period of time. These two parameters, accuracy and time, have to find a good compromise since high accuracy requires smaller cells (lower mesh size – higher number of cells with smaller size) leading to longer computational cost. [40, 41, 42]

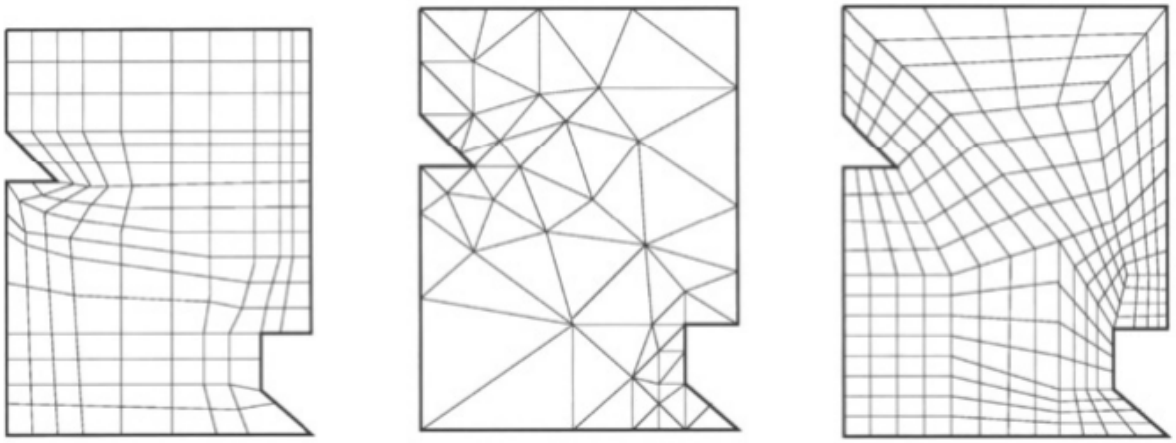
Meshes can be divided into three different types: structured, unstructured and hybrid. These different types of meshes have both their advantages and disadvantages in terms of solution accuracy, complexity of the mesh generation process and computational cost.

Generally, structured meshes are characterized by implicit connectivity, meaning that all interior vertices are topologically alike. When comparing structured to unstructured meshes, the former offer higher space efficiency (due to the definition of the neighbourhood relationships by storage arrangement), better convergence and higher resolution. These meshes are simpler and allow for an easier data access. [40, 41, 42, 43]

On the other hand, unstructured meshes are defined by irregular connectivity, this implies that theses meshes are hardly expressed as a two-dimensional or three-dimensional array in computer memory. The vertices of cells of these meshes have arbitrarily varying local neighbourhoods. Here, the connectivity is defined explicitly. This type of meshes are faster for the user to create as they do not require the division of the domain into several blocks depending on the geometry. The generation of unstructured grids involves two steps: point creation and definition of connectivity between these points. This type of grid offers more flexibility and automation, making them more favourable to CFD problems. However, their accuracy is relatively low when compared to the structured meshes due to the presence of skewed elements in the sensitive regions – boundary layers, for example. [40, 41, 42, 43]

In order to combine the advantages from both structured and unstructured meshes, hybrid meshes can be generated. These meshes contain a mixture of structured portions – where the geometry of the domain is simple – and unstructured portions – where the geometry of the domain is complex. This fact causes the mesh to be extremely flexible for adaption to complex geometry while not compromising so much resolution accuracy.

For a better understanding of the geometrical differences of these three types of meshes, Figure 2.1 represents an example of each type.



(a) *Structured Mesh* ***(b)*** *Unstructured Mesh* ***(c)*** *Hybrid Mesh*

Figure 2.1: *Different types of meshes. [43]*

For the purpose of comparing mesh sizes it is necessary to define a reference length or characteristic length (h). The characteristic length defines the mean cell length and provides an idea about the number of cells that compose the mesh as well as the computational cost. This measurement can be used for any type of mesh and is calculated according to Equation 1.

$$h = \sqrt{\text{Cell}_{\text{area}}} \tag{1}$$

Where $Cell_{area}$ designates the area of the cell.

Currently, *SOL* is capable of solving CFD problems using any type of mesh.

2.2.2 Governing Equations

As previously mentioned, any problem related to fluid flow will satisfy a set of well-documented and modelled laws. The fundamental basis for fluid dynamics and thus CFD, is the Navier-Stokes Equation. The NSE is a set of partial differential equations that describe a fluid in motion. This set of partial differential equations is often simplified in accordance with the specifications of the problem to be solved. The NSE are derived from four different conservation principles: mass, energy, momentum and angular momentum.

The principal of conservation of mass states that: for a system closed to all transfers of matter and energy, the mass of the system must remain constant over time. This is described by Equation 2.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2}$$

Where ρ represents the density of the fluid and $\mathbf{u}$ is the fluid velocity.

The principal of conservation of momentum is described by Equation 3.

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} - \nabla \left(p + \frac{2}{3} \mu \nabla \cdot \mathbf{u} \right) + \nabla \cdot \left[\mu \nabla \mathbf{u} + \mu \nabla^T \mathbf{u} \right] \tag{3}$$

Where g represents the gravity acceleration, p is the pressure and ν is the dynamic viscosity of the fluid.

Additionally, since one of the objectives of this work is to study and evaluate the performance of conjugated problems using the IB Method, there is the need of using the energy conservation law – Equation 4.

$$\frac{\partial\left(\rho C_v T\right)}{\partial t}+\nabla\cdot(\rho C_v T\mathbf{u})=\rho C_v T\mathbf{u}-\nabla\cdot(p\mathbf{u})-\nabla\cdot\left(\frac{2}{3}\mu(\nabla\cdot\mathbf{u})\mathbf{u}\right)+\nabla\cdot\left[\mu\left(\nabla\mathbf{u}+\nabla^T\mathbf{u}\right)\right]+\nabla\cdot(\lambda\nabla T)+\rho Q \quad (4)$$

Where C_v is the volumetric heat capacity, T is the fluid temperature, λ is the heat conductivity coefficient and Q is the volumetric heat source.

For the purpose of this work, several assumptions were conducted, namely, it was assumed that the fluid to be simulated is incompressible and Newtonian. With these assumptions, the principle of conservation of mass is simplified to yield Equation 5.

$$\nabla\cdot\mathbf{u}=0 \quad (5)$$

Combining this equation with the momentum equation stated in Equation 3, the Navier-Stokes Equation is obtained in a simplified form – Equation 6.

$$\frac{\partial\mathbf{u}}{\partial t}+\nabla\cdot(\mathbf{u}\otimes\mathbf{u})=\nabla\cdot(\nu\nabla\mathbf{u})-\frac{1}{\rho}\nabla p \quad (6)$$

Equation 6 represents the behaviour of the fluid flow. However, this equation can be transformed to yield the generalized transport equation – Equation 7.

$$\frac{\partial(\rho\phi)}{\partial t}+\nabla\cdot(\rho\phi\mathbf{u})=\nabla\cdot(\Gamma_\phi\nabla\phi)+S_\phi \quad (7)$$

Here, ϕ corresponds to the transported variable (fluid velocity in case of Equation 6 but it can also be temperature, for example). Γ_ϕ is the scalar diffusivity and S_ϕ is the transport equation source term.

The equations presented above can be solved analytically for some simple cases, however, most of the problems require numerical approaches to solve these equations. Three numerical approaches to solve the differential equations are available: the finite differences method, the finite elements method and the finite volume method. Usually, CFD are solved using the finite elements or finite volume methods. In this work, the finite volume method was used since it was already implemented in *SOL*.

Using the finite volume method, the differential equations are integrated within the volume of the cell. Considering Ω as the finite volume with $\partial\Omega$ and applying this method to the generalized transport equation (Equation 7) the following equation is obtained:

$$\underbrace{\int_\Omega\frac{\partial(\rho\phi)}{\partial t}dV}_{\text{temporal term}}+\underbrace{\int_\Omega\nabla\cdot(\rho\phi\mathbf{u})dV}_{\text{convective term}}=\underbrace{\int_\Omega\nabla\cdot(\Gamma_\phi\nabla\phi)\,dV}_{\text{diftusive term}}+\underbrace{\int_\Omega S_\phi dV}_{\text{source term}} \quad (8)$$

From the Divergence-Theorem, it is known that:

$$\int_{\Omega} \nabla \cdot \mathbf{u} dV = \int_{\partial\Omega} \mathbf{u}.d\mathbf{S} \tag{9}$$

$$\int_{\Omega} \nabla \phi dV = \int_{\partial\Omega} \phi.d\mathbf{S} \tag{10}$$

The application of this theorem in Equation 8 results in final form of the generalized transport equation after the numerical treatment using the finite volume method – Equation 11.

$$\underbrace{\int_{\Omega} \frac{\partial(\rho\phi)}{\partial t} dV}_{\text{temporal term}} + \underbrace{\int_{\partial\Omega} \rho\phi\mathbf{u}.d\mathbf{S}}_{\text{convective term}} = \underbrace{\int_{\partial\Omega} \rho\Gamma_{\phi}\nabla\phi.d\mathbf{S}}_{\text{diftusive term}} + \underbrace{\int_{\Omega} S_{\phi} dV}_{\text{source term}} \tag{11}$$

Each of these terms presented in the above equation are treated using different numerical discretization schemes, which will be explained in the following section.

2.2.3 Discretization Schemes

SOL has several discretization schemes available. For the purpose of this work, discretization schemes suitable for unstructured grids were considered as they are capable of solving any type of meshes, offering a more robust response independently of the mesh type.

A brief explanation on the discretization methods for each of the terms represented in Equation 11 is presented below.

2.2.3.1 Temporal Scheme

During this work, all the simulations were performed considering steady-state. That implies that the transported variables are time independent – do not alter within time. For that reason, the temporal term in Equation 11 is equal to zero (Equation 12), simplifying that same equation into Equation 13:

$$\underbrace{\int_{\Omega} \frac{\partial(\rho\phi)}{\partial t} dV}_{\text{temporal term}} = 0 \tag{12}$$

$$\underbrace{\int_{\partial\Omega} \rho\phi\mathbf{u}.d\mathbf{S}}_{\text{convective term}} = \underbrace{\int_{\partial\Omega} \rho\Gamma_{\phi}\nabla\phi \cdot d\mathbf{S}}_{\text{diftusive term}} + \underbrace{\int_{\Omega} S_{\phi} dV}_{\text{source term}} \tag{13}$$

2.2.3.2 Convective Scheme

The convective scheme is responsible for the discretization of the convective term in Equation 11. For structured meshes, this scheme is a simple two-point linear interpolation, as described in Equation 14.

$$\phi_f = (1 - \eta)\phi_{P_0} + \eta\phi_{P_1} \tag{14}$$

Where, ϕ_f is the transported variable at the face centre of the cell f; ϕ_{P0} and ϕ_{P1} are the transported

variable at cells centroid Pe_0 and P_1, respectively and η is a blending factor based on geometry distances given by:

$$\eta = \frac{\|\mathbf{f} - \mathbf{P}_0\|}{\|\mathbf{P}_1 - \mathbf{P}_0\|} \tag{15}$$

Equation 14 is only suitable for structured meshes. As mentioned, unstructured meshes suffer from the presence of skewed elements. To counter this issue, the convective scheme used for structured grids is altered by changing the blending factor and by adding to Equation 14 a skewness correction factor – Equation 16. [24, 44]

$$\phi_f = \eta_{TRI}\phi_{P_0} + (1 - \eta_{TRI})\,\phi_{P_1} + \underbrace{(\mathbf{f} - \mathbf{P}_0 - \eta_{TRI}(\mathbf{P}_1 - \mathbf{P}_0))\,(\overline{\nabla\phi})_f}_{\text{skewness correction}} \tag{16}$$

$$\eta_{TRI} = \frac{(\mathbf{f} - \mathbf{P}_0)\cdot(\mathbf{P}_1 - \mathbf{P}_0)}{(\mathbf{P}_1 - \mathbf{P}_0)\cdot(\mathbf{P}_1 - \mathbf{P}_0)} \tag{17}$$

Where $(\overline{\nabla\phi})_f$ is the average cell centred gradients of the two considered cells.

With this correction term, the scheme becomes second order accurate for most unstructured grids. When the skewness factor of the grids gets close to zero, the convective scheme equation reverts to Equation 14.

2.2.3.3 Diffusive Schemes

Analogously to the convective scheme, the diffusive scheme is needed for the solution of the diffusive term presented in Equation 11. For structured grids, the diffusive scheme is obtained using the following equation – Equation 18.

$$(\nabla\phi)_f = \frac{\phi_{P_1} - \phi_{P_0}}{\|\mathbf{P}_1 - \mathbf{P}_0\|} \tag{18}$$

For unstructured grids, the diffusive scheme is given by Equation 19. In this equation, a correction for the warp angle based on tangential correction (TC) is included. This correction is required since the desired value is in a direction parallel to the face surface vector (S_f). [45, 46]

$$(\nabla\phi)_f = (\phi_{P_1} - \phi_{P_0})\,\mathbf{p}_{TC} - \underbrace{\left[(\overline{\nabla\phi})_f - \left((\mathbf{P}_1 - \mathbf{P}_0).(\overline{\nabla\phi})_f\right)\mathbf{p}_{TC}\right]}_{\text{warp angle correction}} \tag{19}$$

Where:

$$\mathbf{p}_{TC} = \frac{\mathbf{S}_f}{\mathbf{S}_f\cdot(\mathbf{P}_1 - \mathbf{P}_0)} \tag{20}$$

In the above stated equations, ϕ_{P1} and ϕ_{P0} represent the transported variable at the centroid of the cells P_1 and P_0, respectively, and $(\nabla\phi)_f$ is the gradient of the transported variable at the centre of the face f. Once again, here, the $(\overline{\nabla\phi})_f$ is the average of the gradients of the transported variable ϕ at the centroids P_1 and P_0.

If the mesh is structured, the warp angle correction goes to zero and the diffusive scheme is simpli-

fied to Equation 18.

2.2.3.4 Gradient Scheme

For the computation of the cell centred gradients required in the pressure term of Equation 11, the Gauss method was used:

$$(\nabla\phi)_P = \frac{1}{V_P}\sum_{f\in\mathcal{F}(P)}\phi_f \mathbf{S}_f \tag{21}$$

Here, V_p denotes the volume of cell P. In this equation, ϕ_f was calculated using the convective scheme for unstructured grids, already presented. The calculation is performed using three iteration steps by starting the calculation using the convective scheme without the skewness correction factor.

2.2.4 Pressure Velocity Coupling Algorithm

The coupling between the momentum and mass equations requires a coupling algorithm. In *SOL*, the SIMPLE and PISO algorithms are available. During this work, the SIMPLE algorithm was used in all simulations.

2.2.5 Immersed-Boundary Method

As already mentioned, the immersed-boundary method is becoming very popular among the CFD community as it allows for the solution of more complex problems. This complexity can come from the geometry of the solids – the more complex the geometry, the harder it is to compute the mesh for the solid – or the movement of the solid body – if the body is moving within time, its boundaries are changing position within time requiring a remeshing process at each time step iteration. [5, 8, 47]

When using the IB method, cells that contain only fluid are isolated from the rest of the domain. This division of the fluid domain forms the IB boundary – a delimitation of the shape of the solid following the trends of the mesh, independently of its type. The IB boundary will outline the IB cells and the solid domain. IB cells are cells that contain the interface solid-fluid. Figure 2.2 depicts a generic IB boundary considering a random solid geometry in 2D.

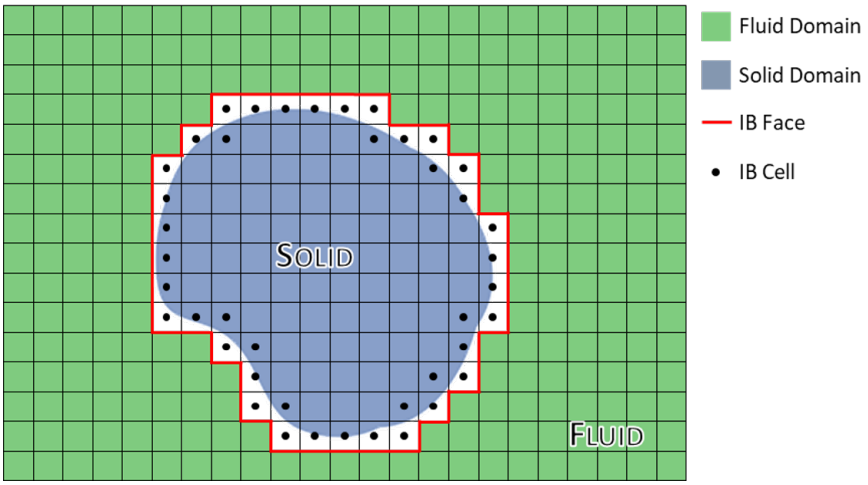


Figure 2.2: *Generic representation of a 2D random solid geometry and its correspondent IB boundary.*

The implementation of the IB method requires two major steps. The first step is when the entire mesh is analysed and the IB boundary is determined. Here, the fluid cells are isolated from IB cells (cells that contain vertices both in the fluid and solid domains) and solid cells. After the determination of the IB boundary, comes the second step of the IB method which is the transport variables interpolation from the solid surface boundary to the IB boundary. When using the IB method, the boundary conditions needed for solving the fluid bulk must be imposed at the IB boundary. Interpolation methods have to be used for the boundary conditions of the solid.

This methodology is repeated at each time iteration step. However, if the solid interface is not changing its position nor its shape, the first step is not required. On the contrary, if the body changes its position or shape throughout time, the IB boundary will be determined at each time step iteration and the boundary conditions will be reinterpolated according to the new body position/shape. Afterwards, the simulation proceeds normally.

In *SOL*, the IB method was already implemented for isothermal systems, meaning that both continuity and momentum equations could be solved using the IB method. This project focused on implementing IB method for non-isothermal systems, in order to be possible to solve heat transfer problems and energy balances using this method.

2.2.6 Boundary Conditions

For solving a differential equation, either ordinary or partial, there is a need to indicate initial and boundary conditions. Both initial and boundary conditions exist in the form of mathematical equations and constrain the problems to the specified boundaries.

Initial conditions are responsible for indicating the initial state of the system, generally for $t = 0$. For a time-dependent problem of order k and n dimensions, generally kn initial conditions will be required in order to solve the differential equation.

Regarding boundary conditions, they generally express the behaviour of the equation at the boundary of the domain that is being solved. A boundary value problem is applied in several branches of physics and fluid mechanics and, consequently, CFD is no exception. There are five types of boundary conditions: Dirichlet, Neumann, Robin, Mixed and Cauchy, within which Dirichlet and Neumann are the most used. Throughout this project and document, several allusions to Dirichlet and Neumann boundary conditions are presented.

In the case of boundary conditions for the generalized NSE, the Dirichlet boundary conditions prescribes a value $BC_{DIRICHLET}$ for the transport variable ϕ at the boundary (Equation 22). For example, in case the transport variable being evaluated is the fluid temperature, the Dirichlet boundary condition will stipulate a temperature value at the boundary of the fluid. This is also called a boundary condition of the first kind.

$$\phi_b = BC_{DIRICHLET} \tag{22}$$

On the other hand, a Neumann condition $BC_{NEUMANN}$ will define the derivative of the transport variable ϕ at the boundary (Equation 23), which following the example when $\phi = T$, the Neumann condition will stipulate a heat flux at the fluid boundary. This is also called a boundary condition of the second kind.

$$\left(\frac{\partial \phi}{\partial n}\right)_b = BC_{NEUMANN} \tag{23}$$

2.2.7 Order of Accuracy

The order of accuracy refers to the rate at which a numerical approximation of a differential equation converges to the analytical solution. Considering the characteristic length h, as given by Equation 1 to be a parameter that characterizes the approximation, a simulation is said to be of order n if the analytical error is proportional to h to the nth power.[48, 49, 50]

2.3 SOL

SOL is a computational software to study, solve and analyse CFD problems. As mentioned before, this software was developed (and it is still under development) by *LASEF – Laboratory of Simulation in Energy and Fluids*, a research group in CFD at Instituto Superior Técnico.

The main idea of *SOL* is to provide *LASEF* researchers a common platform for their CFD studies. Each researcher works in their branch of the code, making the software richer and more collaborative.

The software is built using C language and is capable of importing meshes from other CFD software like *STAR CCM* or *OpenFoam*. It can export data in formats compatible with *Tecplot* or other visualization tools. All in all, *SOL* is a complete and powerful CFD software.

Several thesis and papers using *SOL* were published and are available. [24, 25, 23]

3 Interpolation Methods' Study

In this section, an extensive study on the different interpolation methods at the IB boundary is carried out. Here, the methodology used is described and the results are systematized, compared to the literature and discussed.

These interpolation methods are applied to a physical problem with a known analytical solution - a Taylor-Couette Flow Problem. This constitutes the first thermal verification performed on the IB method implemented in *SOL*.

Firstly, a linear interpolation method is evidenced and tested. The same process is employed to describe and test a least squares interpolation method.

Lastly, a comparison of the analytical error of both Neumann and Dirichlet boundary conditions for the same simulation is performed and discussed.

3.1 Taylor-Couette Flow Problem

This section details the physical problem used to verify the linear thermal interpolation method detailed in the following chapters.

In order to perform the required tests, a physics problem geometry consisting of the two-dimensional flow between two cylinders was defined in *SOL*. This problem is also called a Taylor-Couette flow problem. All simulations in this chapter were conducted in stationary conditions, and meshes used in this chapter are hexagonal meshes.

The fluid domain in this problem corresponds to the space between the outer most cylinder with radius $R_0 = 1$ and and the inner most cylinder with radius $R_{SOLID} = 0,5$. The outer most cylinder is stationary while the inner most cylinder is rotating with a constant angular velocity, $u_\theta = 1rad/s$ in an anti-clockwise direction. A no-slip condition is enforced on all boundaries. Figure 3.1 is a graphical representation of this problem.

Regarding the temperature field; a Dirichlet boundary condition is applied on the outer wall, which is maintained at $T_0 = 0$. The boundary condition at the inner circle varies between different simulations. When a Dirichlet condition is applied, it has a value of $T_{SOLID} = 1$. When a Neumann boundary condition is applied it has a value of $\left[\frac{dT}{dr}\right]_{r=R_{SOLID}} = 1$.

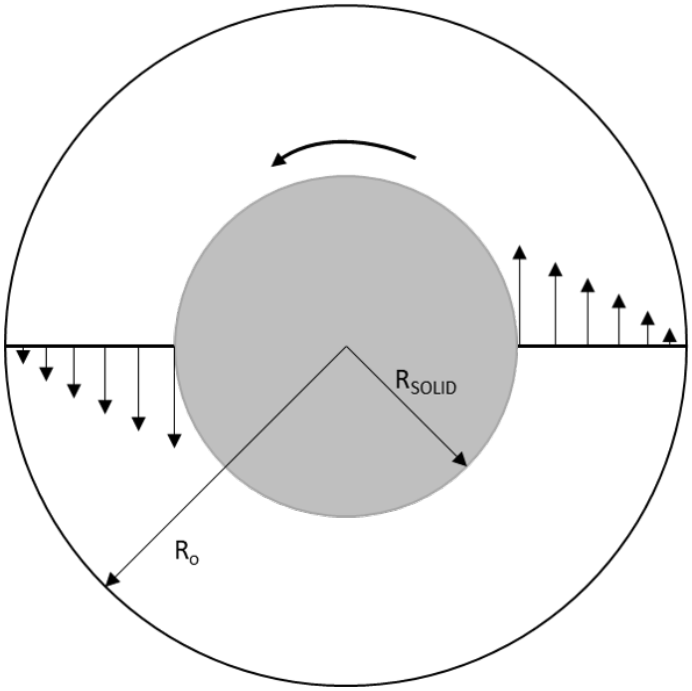


Figure 3.1: *Coannular cylinder geometry used for code verification. Figure adapted from Fig 10 in [51].*

All Taylor-Couette simulations conducted use $Pr = 1$ and $Re = 0,25$ where Prandtl and Reynolds number are calculated as follow. [52]

$$Re = \frac{u_\theta R_{SOLID}\,(R_0 - R_{SOLID})}{\nu} \tag{24}$$

$$Pr = \frac{\nu}{\alpha} \tag{25}$$

Where ν represents the kinematic viscosity and α the thermal diffusivity.

After this physical domain is fully defined, the analytical solution for both velocity and temperature fields are implemented in *SOL*. This allows for the calculation of analytical errors for both the velocity and temperature fields given by each simulation. These solutions are as follow: [51]

For the velocity field:

$$u_r = 0; \quad u_\theta(r) = \begin{cases} 0 & \text{for } R_i < r < R_{SOLID} \text{ (solid)} \\ -\frac{R_0 R_{SOLID}^2}{R_o^2 - R_{SOLID}^2} \cdot \frac{1}{r} + \frac{R_0}{R_o^2 - R_{SOLID}^2} \cdot r & \text{for } R_{SOLID} < r < R_o \text{ (fluid)} \end{cases} \tag{26}$$

For the temperature field with a Dirichlet boundary condition at the IB:

$$T(r) = \begin{cases} - & \text{for } R_i < r < R_{SOLID} \text{ (solid)} \\ T_0 - \frac{(T_0 - T_{SOLID})\log(R_o/r)}{\log(R_o/R_{SOLID})} & \text{for } R_{SOLID} < r < R_o \text{ (fluid)} \end{cases} \tag{27}$$

For the temperature field with a Neumann boundary condition at the IB:

$$T(r) = \begin{cases} - & \text{for } R_i < r < R_{SOLID} \text{ (solid)} \\ \frac{T_{SOLID}\log(R_o/r) + T_0\log(r/R_{SOLID})}{\log(R_o/R_{SOLID})} & \text{for } R_{SOLID} < r < R_o \text{ (fluid)} \end{cases} \tag{28}$$

Where $T_{SOLID} = T_o - q''_{SOLID} \log\left(R_o / R_{SOLID}\right)$.

The errors are obtained by computing the difference between the simulation results and the relevant analytical solution. The results obtained show consistency with the analytical solution presented, in both velocity and temperature fields for both boundary conditions considered.

Regarding the dimensions of several variables in these simulations, lengths are made dimensionless with $(R_0 - R_{SOLID})$, velocity with $(u_\theta R_{SOLID})$ and temperature with T_{SOLID}.

3.2 Interpolation Methods

This section details all the interpolation methods developed and implemented in *SOL* for the calculation of the thermal boundary condition on the IB boundary. Both the linear interpolation methods and the least squares interpolation methods presented in this Chapter were already implemented in SOL when the work presented in this documented began. However, no documentation about these methods was available. After exploring and testing the available code, the methods were found to be analogous to those implemented by Jonas Pereira in [23] for the velocity interpolation, and are presented in the following sections.

3.2.1 Linear Method

This section exposes the linear methods developed to interpolate Dirichlet and Neumann boundary conditions from the solid boundary to the IB boundary. Both methods have a theoretical order of accuracy of 1.

3.2.1.1 Dirichlet Boundary Condition

This subsection details the first order temperature interpolation, this is, using a Dirichlet boundary condition at the solid boundary.

To linearly interpolate the temperature at the immersed boundary a ratio of distances must first be defined:

$$\eta = \frac{d_1}{d_1 + d_2} \tag{29}$$

Where d_1 and d_2 are defined as the distance from the IB face center to the fluid cell center of the cell containing the IB face and as the distance between the IB face center to the closest solid material point, respectively. These distances are schematized in Figure 3.2.

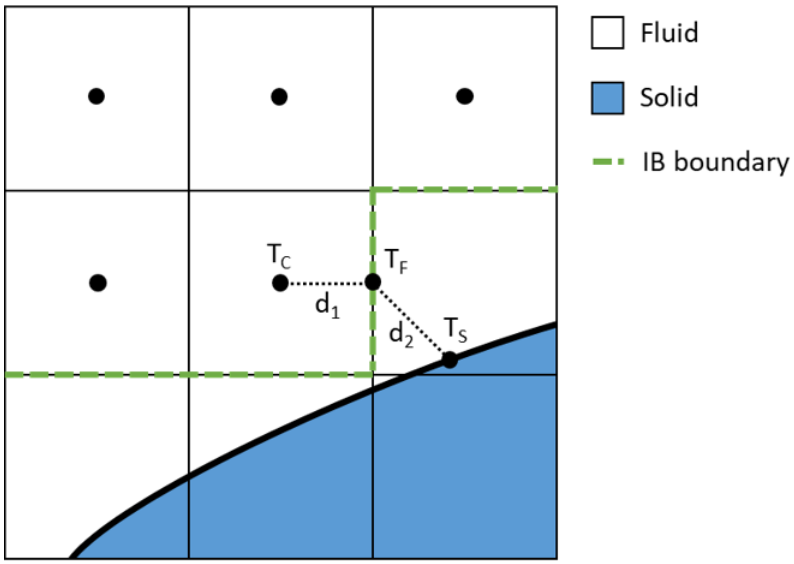


Figure 3.2: *Definition of distances and location of interpolated values used in linear interpolation method for Dirichlet boundary condition.*

The temperature at the IB face center can be obtained with a weighted average using the calculated distance ratio:

$$T_f = T_c(1-\eta) + \eta T_s \tag{30}$$

Where T_f is the interpolated temperature at the IB face center, T_c is the temperature at the fluid cell center which includes the IB face and T_s is the temperature at the closest solid point. These temperatures are also schematized in Figure 3.2. The calculated temperature is now the Dirichlet boundary condition applied at the IB face center, allowing the bulk solver to simulate the entire fluid domain.

3.2.1.2 Neumann Boundary Condition

This subsection details the first order temperature gradient interpolation employed at the immersed boundary. This method is utilized when a Neumann boundary condition is imposed at the solid boundary.

To linearly interpolate the temperature gradient at the immersed boundary the same ratio of distances defined on Equation 29 is utilized.

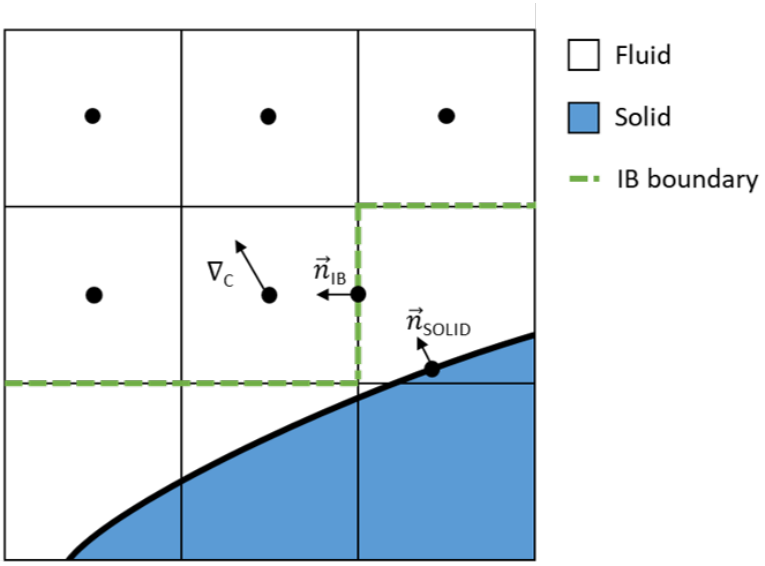


Figure 3.3: *Definition of vectors and locations used in the linear interpolation method for Neumann boundary condition.*

Figure 3.3 shows the vectors required for this interpolation. ∇_c represents the temperature gradient at the fluid cell center, $\vec{n}_{IB}$ is the unit length vector normal to the IB face pointing outwards from the solid body, and $\vec{n}_{SOLID}$ is the unit length vector normal to the solid body pointing outwards from the solid body.

Furthermore, two intermediary variables are needed, ϕ_s and ϕ_c , which are defined as follows:

$$\phi_s = \nabla_T \cdot \vec{n}_{IB} \tag{31}$$

$$\phi_c = (\vec{n}_{IB} \cdot \vec{n}_{SOLID}) \times \left[\frac{dT}{dn}\right]_{SOLID} \tag{32}$$

Where $\left[\frac{dT}{dn}\right]_{SOLID}$ is the value of the normal temperature gradient at the solid boundary, this is, the Neumann condition being imposed at the solid boundary.

Finally, the value for the temperature gradient at the IB face center is calculated with a weighted average using the calculated distance ratio in an analogous manner to the linear temperature interpolation of Equation 30:

$$\phi_f = \phi_c(1-\eta) + \eta\phi_s \tag{33}$$

Where ϕ_f is the temperature gradient for the IB face center. This value is now the Neumann boundary condition applied at the IB face center, allowing the bulk solver to simulate the entire fluid domain.

3.2.2 Least Squares Interpolation Method

A second order interpolation scheme for the immersed boundary using the least squares method is developed, implemented and tested in this section. The scheme differs from the previously explained linear method in that the temperature interpolation is not calculated considering only one fluid cell and one body point, but a group of neighbouring fluid cells and geometrical body points. These points are

arranged in a stencil built using the same algorithm developed by Jonas Pereira [23] for the interpolation of the velocity values in the IB calculation. The method employed differs when imposing a temperature or a temperature gradient at the solid boundary, and as such will be explained separately.

3.2.2.1 Dirichlet Boundary Condition

This section details the methodology employed to interpolate a boundary condition for the IB face from an imposed temperature (Dirichlet BC) on the solid boundary, using a least squares methodology. The number of points considered varies depending on the local geometry. However, a generic situation is presented in Figure 3.4, where 5 neighbouring fluid cell centers, 4 IB face centers and 3 solid points are shown.

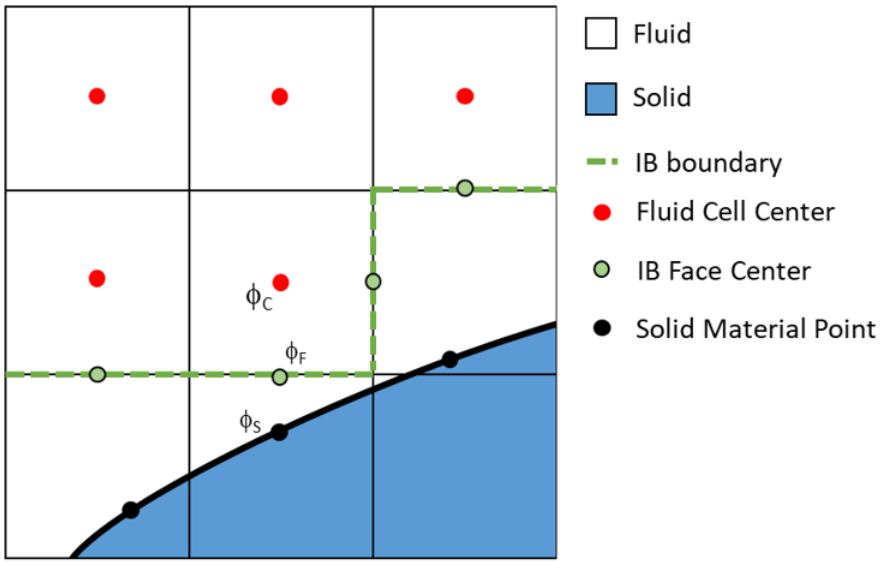


Figure 3.4: *Schematic of relevant points for creation of the stencil in a generic face f.*

In this case, the calculation of the value in face f, ϕ_f, uses a stencil consisting of 5 fluid cell (red dots) and 3 solid points (black dots). Each of these points represents a value for the temperature. No values from IB face centres are used for the interpolation.

A quadratic polynomial is now created for each point in the stencil, following the form:

$$\phi = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy \tag{34}$$

Where the coefficients β_n are determined by the least squares method and x and y represent the points' coordinates in the simulation's referential. In matrix form, this equation can be written as: $\phi = M\vec{\beta}$ where $\vec{\beta} = [\beta_0, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5]^T$ and M :

$$M = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2 y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & y_n^2 & x_n y_n \end{bmatrix} \tag{35}$$

Following the least squares method, minimizing the square of the difference between the values considered at each point corresponds to minimizing the value of $\|\phi - M\vec{\beta}\|^2$. Following the works of

Kariya and Kurata [53] this minimization occurs when the calculation of the vector β is perfomed as follows:

$$\beta = \left(M^T \cdot M\right)^{-1} M^T \phi \tag{36}$$

Following the determination of the β vector, the temperature at the IB face center is directly calculated using Equation 34. This calculated temperature is now the Dirichlet boundary condition applied at the IB face center, allowing the bulk solver to simulate the entire fluid domain.

3.2.2.2 Neumann Boundary Condition

This section details the methodology employed to interpolate a boundary condition for the IB face from an imposed temperature gradient (Neumann BC) on the solid boundary, using a least squares methodology.

The same generic detail of the IB cut presented in Figure 3.4 is considered. However, a new referential is used, centered on the solid point and using the solid points normal direction (vector $\vec{n}_{SOLID}$) pointing outwards as the x direction and the tangential body direction (vector $\vec{t}_{SOLID}$) as the y direction, as schemetized in Figure 3.5. All points considered in this section use this referential.

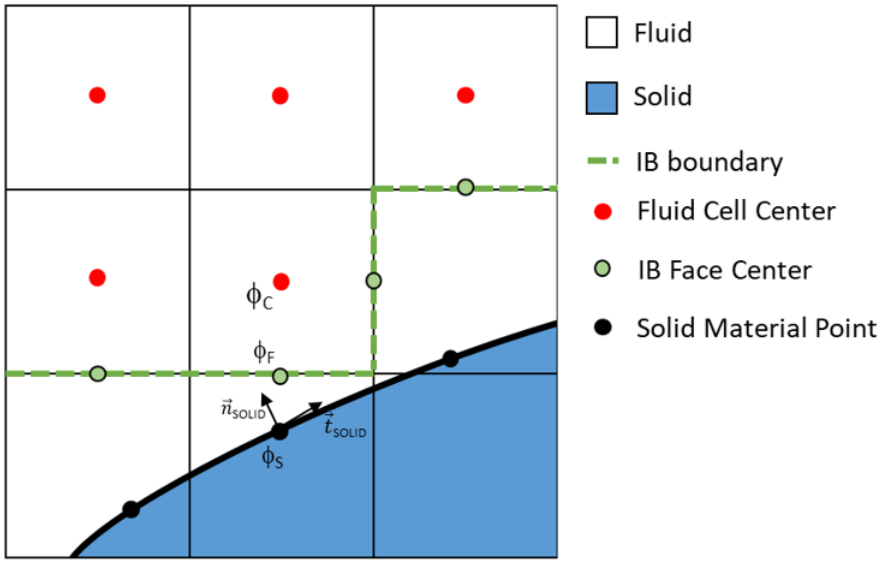


Figure 3.5: *Schematic of relevant points for creation of the stencil in a generic face f for Neumann boundary condition.*

In the generic case presented, 5 neighbouring fluid cell centers and 3 solid points are considered. The same quadratic polynomial as presented in Equation 34 is used for all fluid cell center points. However, the value inherited from solid points is a temperature derivative, forcing that the quadratic polynomial in Equation 34 must be differentiated with respect to x , the normal body direction, resulting in Equation 37.

$$\frac{d\phi}{dx} = \beta_1 + 2x\beta_3 + \beta_5 y \tag{37}$$

Since the matrix includes rows relative to points with thermal information in the form of temperature, and points with information in the form of temperature gradients, Equation 37 is multiplied by the char-

acteristic length h prior to it's introduction in the matrix as a way to ensure the same dimensions across both types of rows of matrix $M_{Neumann}$.

Thus, the main differences from the methodology employed in Section 3.2.2.1 lies in the construction of matrix $M_{Neumann}$ as seen in Equation 38.

$$M_{Neumann} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2 y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & y_n^2 & x_n y_n \\ 0 & h & 0 & 2x_{n+1}h & 0 & y_{n+1}h \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & h & 0 & 2x_{n+m}h & 0 & y_{n+m}h \end{bmatrix} \tag{38}$$

Where n is the number of fluid cell centers in the stencil and m is the number of solid points in the stencil. After Equation 36 has been solved and the value for the β coefficients is known, the temperature gradient at the IB cell centered is calculated by first finding the gradient at this point, in the considered coordinate system, by using Equation 37 as well as the derivative in the second direction, presented in Equation 39.

$$\frac{d\phi}{dy} = \beta_2 + 2y\beta_4 + \beta_5 x \tag{39}$$

The Neumann condition imposed at the IB face center simply becomes the temperature gradient that occurs in the direction of the face normal.

3.3 Results and Discussion

In this section the results from mesh refinement studies for the Taylor-Couette flow problem simulations using the various interpolation methods are exposed, evaluated and discussed.

3.3.1 Velocity

This section relates to the results for the velocity field obtained. Figure 3.6 shows the observed order of accuracy for these values and Figure 3.7 represents the velocity field in the fluid domain.

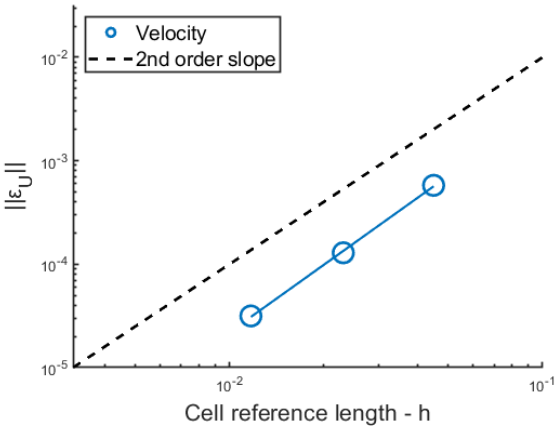


Figure 3.6: *Convergence of the mean velocity error for a Taylor-Couette flow simulation.*

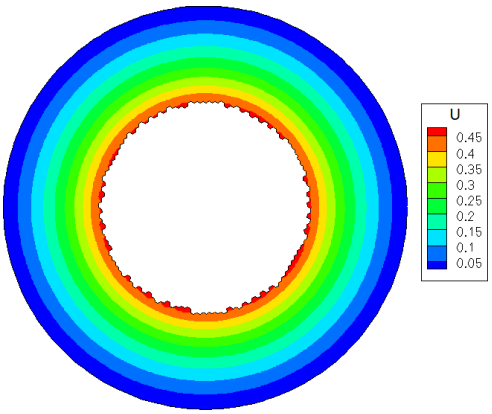


Figure 3.7: *Velocity field for Taylor-Couette flow simulations.*

The observed error decay was obtained by computing a trend line of the different mean errors associated to different mesh refinements. This error decay corresponds to the slope of the trend line obtained in a log-log space. For the velocity field, the slope obtained is equal to 2,16.

Since the simulation of the velocity field, and consequently the pressure field, are decoupled from the calculation of the temperature field, the velocity and pressure fields above are valid for all Taylor-Couette simulations present in this document.

3.3.2 Linear Interpolation Method

This section presents the results for the temperature field obtained using the linear interpolation methods described in Chapter 3.2.1. Figure 3.8 shows the observed behaviour of the mean error for a mesh refinement study.

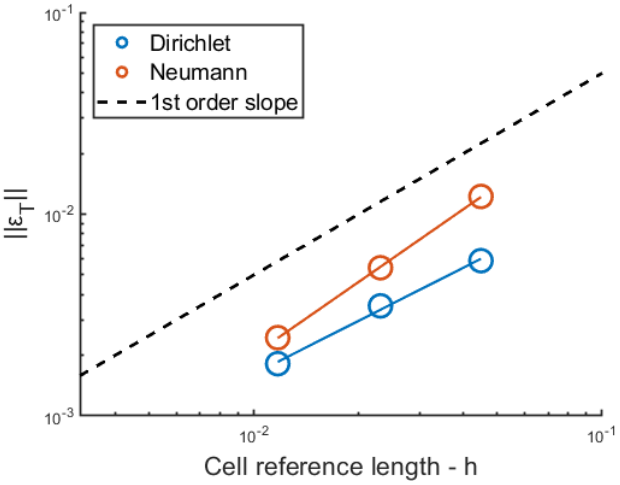


Figure 3.8: *Convergence of the mean temperature error for a Taylor-Couette flow simulation.*

Table 3.1: *Observed order of convergence using the linear interpolation method for both boundary conditions.*

	Linear
Imposed Temperature (Dirichlet BC)	0,88
Imposed Temperature Gradient (Neumann BC)	1,20

Table 3.1 represents the observed orders of accuracy for the linear interpolation methods, obtained in an analogous manner to trend line methodology used for the velocity field.

Both schemes' observed order of accuracy are within an acceptable variance of the expected value of 1. The method for imposed temperature, despite having a deviation of 12% from the expected value, has lower analytical error in all considered meshes than the imposed temperature gradient method, despite its considerably higher observed order of accuracy. Since these methods present first degree order of accuracy, a mesh refinement study for a physical problem with coupling of the thermal and velocity fields would lead to a decrease of the velocity field calculation's order of accuracy. To address the need for a higher order thermal interpolation method, the least squares method is studied in Section 3.3.3.

3.3.3 Least Squares Interpolation Method

This section presents the results obtained using the least squares interpolation method for the calculation of the thermal boundary conditions at the IB boundary. The values presented were obtained using both considered boundary conditions.

These results are used for the mesh refinement study using both Dirichlet and Neumann boundary conditions. The methodology used is explained below.

Using the analytical expression of the temperature field, Equations 27 and 28, it is possible to calculate both the imposed temperature and the imposed heat flux at the immersed boundary for which the resultant temperature field is the same.

The solution for the Dirichlet boundary condition, Equation 27, can be differentiated in regards to the radius, resulting in:

$$\frac{dT}{dr} = \frac{T_{SOLID} - T_0}{r \log \frac{R_0}{R_{SOLID}}} \tag{40}$$

From this result any pair of values for Dirichlet and Neumann boundary conditions can be obtained, considering a Dirichlet boundary condition defined by T_{SOLID} and a Neumann boundary condition by $\left[\frac{dT}{dr}\right]_{r=R_{SOLID}}$. Thus, the resultant temperature field from imposing a Dirichlet boundary condition of $T_{SOLID} = 1$ is the same as the obtained from imposing the Neumann boundary condition $\left[\frac{dT}{dr}\right]_{r=R_{SOLID}} = 2.885390$. The physical problem is then simulated for both these conditions.

Figure 3.9 shows the observed behaviour of the mean temperature error, Table 3.2 shows the observed order of accuracy and Figure 3.10 shows the resultant temperature field for a Dirichlet boundary condition of $T_{SOLID} = 1$.

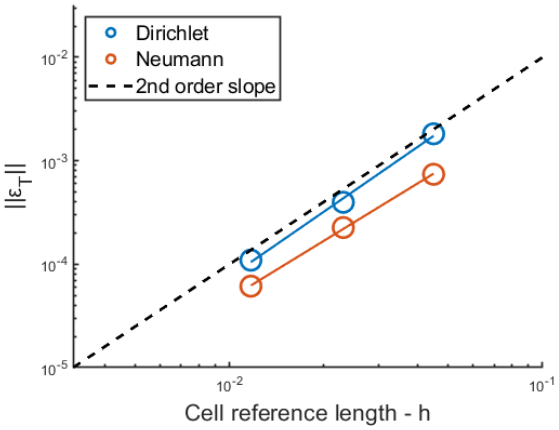


Figure 3.9: *Convergence of the mean temperature error for a Taylor-Couette flow simulation with least squares interpolation method at the IB for Neumann and Dirichlet boundary conditions.*

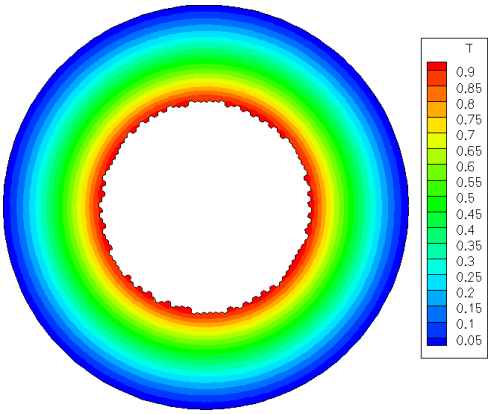


Figure 3.10: *Temperature field for Taylor-Couette simulation.*

Table 3.2: *Observed order of convergence using the least squares interpolation method for both boundary conditions.*

	Least Squares
Imposed Temperature (Dirichlet BC)	2,09
Imposed Temperature Gradient (Neumann BC)	1,85

The results obtained with the least squares interpolation methods present a similar situation to the results obtained with linear methods since both boundary conditions observed order of convergence has a maximum deviation of 7,5% from the theoretical value of 2. This maximum deviation occurs for the Neumann boundary condition. The overall analytical error is an entire order of magnitude below that of the linear interpolation methods for most of the considered meshes, as is expected for higher order schemes. The least squares interpolation method also has a reasonably lower error when imposing temperature gradient rather than temperature. This difference is believed to be due to the physical problem considered, a Taylor-Couette flow problem, since it results in a purely diffusive one dimensional problem, if only considering the temperature field. Thus, it is preferable to impose a temperature gradient in the solid body, and thus in the IB boundary, than to impose a temperature, since the gradient can immediately be imposed in the diffusive flux at the face. Further testing of this method is presented in Chapter 4.

4 Mesh Geometry Robustness Study

In this chapter, the interpolation methods implemented and discussed in Chapter 3 are tested and verified on both structured and hybrid meshes, both for imposed temperature and imposed temperature gradient at the solid boundary. The results are then presented and discussed.

Testing the current capabilities of *SOL* at handling heat exchange over an immersed boundary using several different types of mesh and levels of refinement presents an invaluable opportunity to assess the robustness of both the methods developed and implemented in Chapter 3 and of *SOL* as a whole.

SOL's response to this kind of testing can indicate both limitations in some methods' mesh handling and other methods' capabilities at solving the same problem. This knowledge enables an advantageous starting point for future simulations being employed using the optimal approach and optimal interpolation methodology selection.

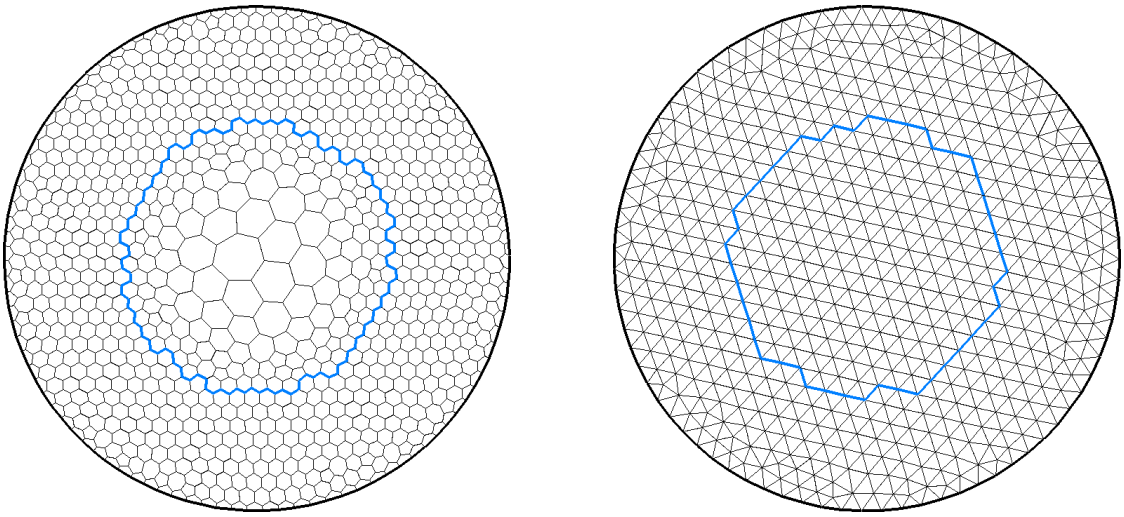
All simulations in this chapter follow the same methodology described in section 3.1 of a Taylor-Couette physical problem.

4.1 Structured Meshes

In this section the structured meshes selected for the tests are presented.

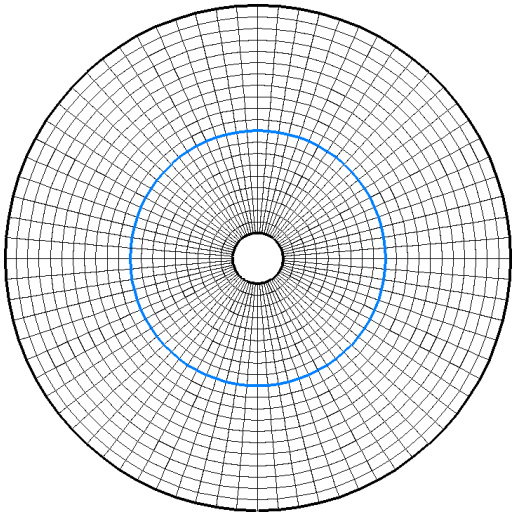
Three different types of structured meshes are considered. The first type of meshes used are hexagonal, strictly structured in the fluid domain. The second type of structured mesh employed are triangular meshes, with some small variance in cell connectivity near the outer circle due to the body fitted nature of the mesh in regards to this outer boundary. The third and last type of mesh considered are fully structured radial quadrilateral mesh. For all types of meshes employed, three different levels of refinement are considered, corresponding to three different characteristic lengths, calculated using Equation 1.

The 3 different types of structured meshes used are represented in Figure 4.1.



(a) *Hexagonal Mesh.*

(b) *Triangular Mesh.*



(c) *Radial Quadrilateral Mesh.*

Figure 4.1: *Different types of structured meshes used for robustness testing. The IB boundary is represented in blue. Body-fitted boundaries are represented in black.*

The boundary found inside the IB boundary of meshes type *(c)* - Radial Quadrilateral has no relevance to the current simulations since it is entirely inside the solid boundary and consequently the IB boundary, being however necessary for the meshing process of such grids.

It is also worth noting that meshes of the type *(c)* - Radial Quadrilateral are extremely well suited to the Taylor-Couette flow problem, with cells perfectly aligned along the only direction on which heat diffusion occurs - radial. This must be taken into consideration when comparing the results using this mesh type to others.

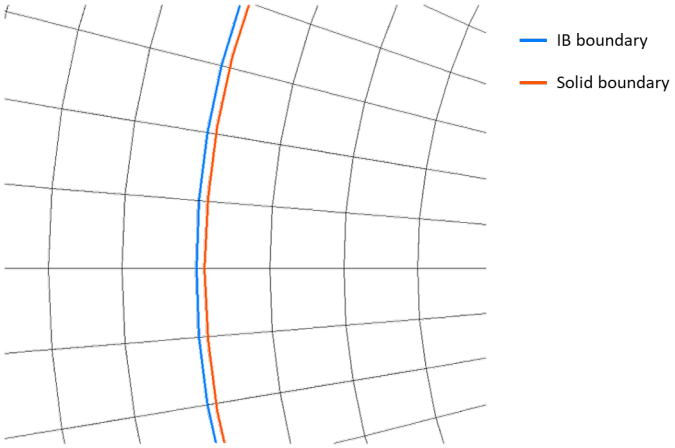
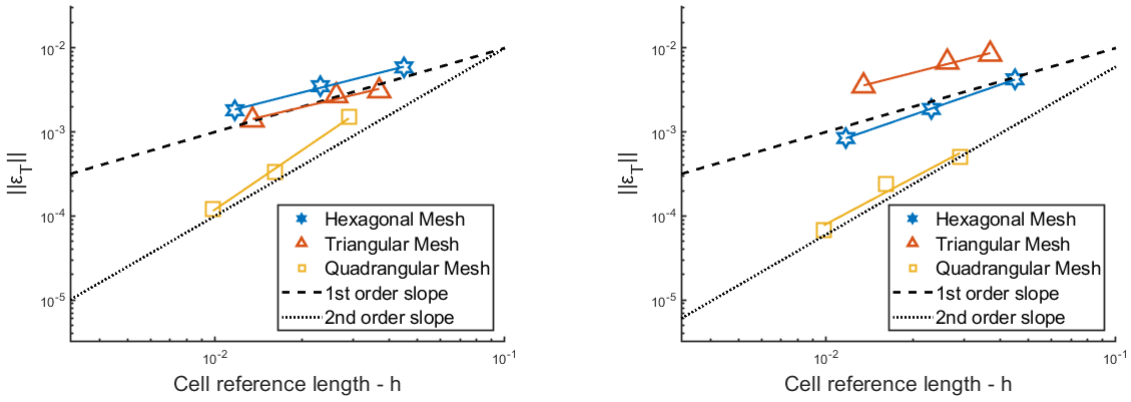


Figure 4.2: *Detail of the IB cut and solid boundary on the coarsest radial quadrilateral mesh.*

This geometry also presents advantages in regards to the IB cut, since the IB boundary has the same geometry as the solid body, a circle. However, the IB boundary is not coincident with the solid boundary, as shown in Figure 4.2.

4.1.1 Linear Interpolation Methods Results and Discussion

This section exposes the results from the Taylor-Couette simulations using the structured meshes described in the previous section (Chapter 4.1). Considering the known analytical solution (Equations 27 and 28) the resultant temperature field can be translated into mean analytical errors, which are presented in this section. All results are obtained using the linear interpolation method described in Section 3.2.1.



(a) *Error decay with Dirichlet boundary condition.* ***(b)*** *Error decay with Neumann boundary condition.*

Figure 4.3: *Observed error decay with linear interpolation method for various mesh types.*

Table 4.1: *Observed order of convergence with linear method for three different meshes and boundary conditions.*

	Triangular	Quadrilateral	Hexagonal
Imposed Temperature (Dirichlet BC)	0,82	2,35	0,88
Imposed Temperature Gradient (Neumann BC)	0,88	1,83	1,20

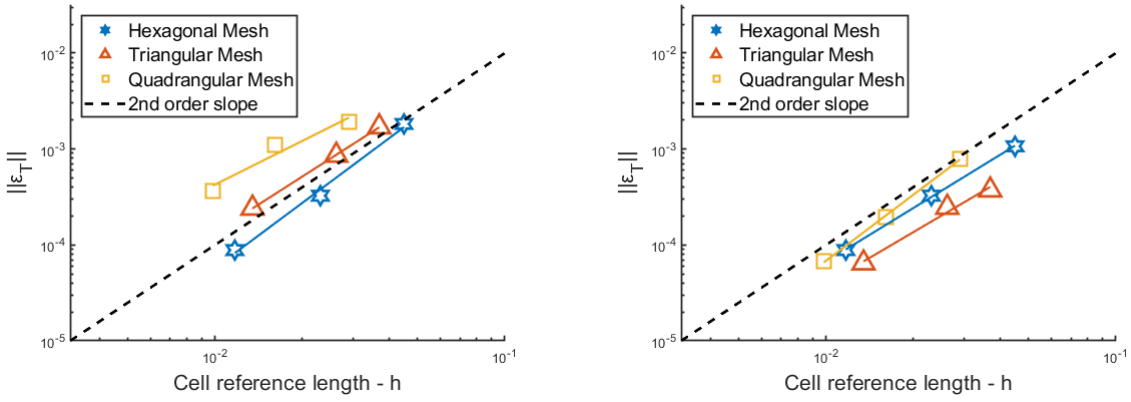
Figure 4.3 shows the observed behaviour for the considered methods, across both bondary conditions considered. Table 4.1 presents the observed order of accuracy for all studies presented in this section.

The mesh refinement studies conducted for the linear interpolation methods in this Section demonstrate a slight decrease in the observed order of accuracy when using triangular structured meshes compared to hexagonal meshes, which is an expected result. The higher computational power required to mesh a geometry using hexagonal cells is well documented to improve accuracy of schemes over the easier to mesh triangular geometries. The observed order of accuracy for triangular meshes has a maximum deviation of 18% from the theoretical value of 1, which while considerable is compensated by a much easier meshing process, as mentioned. [40]

The major outliers are the results obtained when using quadrilateral radial meshes, where an observed order of accuracy much higher than the theoretical value of 1 is verified. Considering the exceptional quality these meshes possess for the current problem, as described in Chapter 4.1, these values do not point to a particularly capable scheme, but instead to a high sensibility to mesh quality. This hypothesis is tested in Chapter 4.2.

4.1.2 Least Squares Interpolation Method Results and Discussion

Analogously to the previous Section 4.1.1, this section exposes the results from the Taylor-Couette simulations using the structured meshes described in Section 4.1. Considering the known analytical solution (Equations 27 and 28) the resultant temperature field can be translated into mean analytical errors, which are presented here. All results are obtained using the least squares interpolation method described in Section 3.2.2.



(a) *Error decay with Dirichlet boundary condition.* ***(b)*** *Error decay with Neumann boundary condition.*

Figure 4.4: *Observed error decay with least squares interpolation method for various mesh types.*

Table 4.2: *Observed order of convergence with with least squares interpolation method for three different meshes and boundary conditions.*

	Triangular	**Quadrilateral**	**Hexagonal**
Imposed Temperature (Dirichlet BC)	1,93	1,51	2,25
Imposed Temperature Gradient (Neumann BC)	1,78	2,28	1,85

Table 4.2 presents the observed order of convergence for the Dirichlet boundary condition refinement study shown in Figure 4.4a and for the Neumann boundary condition shown in Figure 4.4b. Figure 4.5 shows the temperature field resultant from a Taylor-Couette flow simulation with unit Neumann boundary condition.

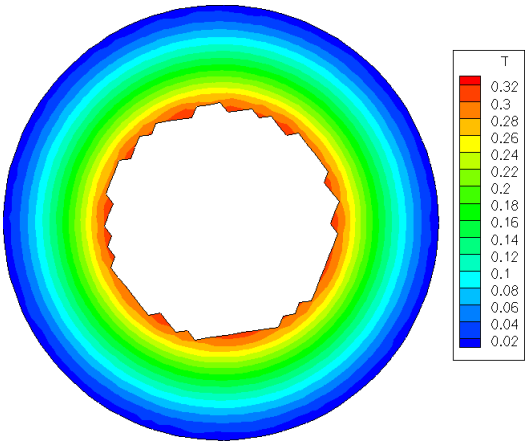


Figure 4.5: *Temperature field for a triangular mesh using least squares interpolation method and a unitary Neumann boundary condition.*

The mesh refinement studies conducted for the least squares interpolation method in this Section demonstrate a slight decrease in the observed order of accuracy when using triangular structured meshes compared to hexagonal meshes, which is an expected result which was also observed in the behaviour of the linear interpolation methods.

The major point to note is the result obtained when using quadrilateral radial meshes, where an observed order of accuracy half an order lower than the theoretical value of 2 is verified. This order of accuracy is in fact lower than that registered for the same meshes when utilizing a linear interpolation method. Considering the exceptional quality these meshes possess for the current problem, as described in Chapter 4.1, this behaviour points to a mesh geometry particularly optimal to implement the linear interpolation method, which resulted in an exceedingly higher order of accuracy than expected when using this method. Overall, these quadrilateral meshes observed behaviour is concordant with what was expected, with deviations due to it's particularly good fitting to the current physical problem.

It is worth noting that all three structured mesh types considered can be reused for conjugate heat transfer problems, with varying advantages. The higher coarseness inside the IB boundary for the hexagonal meshes would be of great value in conjugate heat transfer problems since the solid domain would represent a purely diffusive problem, which coupled with the much higher typical thermal conductivity of solids compared to that of fluids leads to the lower need for mesh refinement in this area. Complementarily, the radial quadrilateral meshes' internal boundary would be appropriate to represent, for example, an imposed heat flux typically associated with an electrical resistance, running through a

solid submerged in a fluid onto which it is dissipating heat.

4.2 Hybrid Meshes

In this section the hybrid meshes employed for testing are presented.

These hybrid meshes have a much wider range of connectivity compared to the previously considered structured meshes (Section 4.1). While most elements on these grids are quadrilaterals, triangles also occur, which means each vertex belongs to a varying number of cells, ranging from 3 to 5. The characteristics of these meshes lead to a higher difficulty for the interpolation methods at the IB boundary to function at the expected order of accuracy.

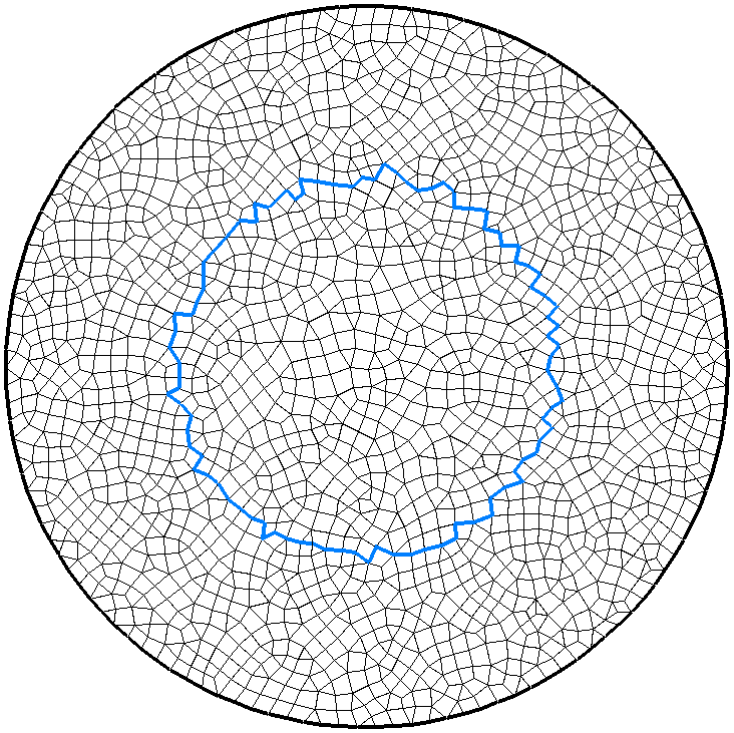
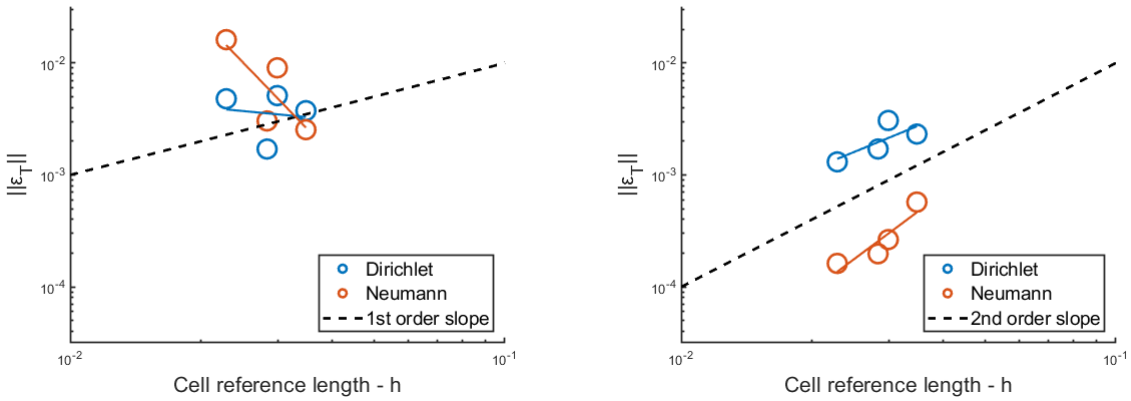


Figure 4.6: *Example of a hybrid mesh used for robustness testing. The IB boundary is represented in blue. The outer circle, a body-fitted boundary, is represented in black.*

Five hybrid meshes of similar geometry but varying levels of refinement were considered. Figure 4.6 shows one such mesh, the second coarsest. This figure also demonstrates the higher complexity of the IB boundary when compared to structured meshes in Section 4.1.

4.2.1 Results

This section presents all the results of the mesh refinement studies conducted on hybrid meshes. Figure 4.7 shows the observed behaviour, and Table 4.3 presents the observed order of accuracy for all used methods and boundary conditions.



(a) *Observed error decay with linear interpolation method for hybrid meshes.* ***(b)*** *Observed error decay with least squares interpolation method for hybrid meshes.*

Figure 4.7: *Observed error decay for hybrid meshes.*

Table 4.3: *Observed order of convergence in hybrid meshes.*

	Linear	Least Squares
Imposed Temperature (Dirichlet BC)	-0,37	1,59
Imposed Temperature Gradient (Neumann BC)	-4,02	2,89

Table 4.8 shows the temperature field obtained using a hybrid mesh and Neumann boundary condition.

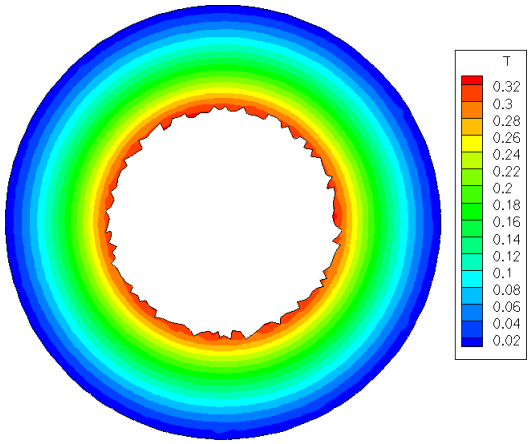


Figure 4.8: *Temperature field for a hybrid mesh using least squares interpolation method and a unitary Neumann boundary condition.*

4.2.2 Discussion

This section discusses the results obtained in Chapter 4.2, in which mesh refinement studies are conducted using both linear and least squares interpolation method using hybrid meshes. Firstly, the results obtained when using the linear interpolation method show a complete lack of sensitivity of the mean temperature error in regards to mesh refinement, pointing to a high degree of mesh sensitivity, as postulated in Section 4.1.1. These methods are not considered capable of handling hybrid mesh geometries, and are discouraged from being used outside of structured mesh types.

The least squares interpolation method shows variable robustness when handling hybrid meshes. When imposing a temperature gradient at the solid boundary, the observed order of accuracy is in fact higher than the theoretical value of 2. However, when imposing temperature, the observed order of accuracy experiences a decrease of 21% regarding the expected second order behaviour. It is known that when the maximum error for the temperature field has no variation across a mesh refinement study, the order of accuracy of the mean error lowers by at least 1. The hypothesis that this phenomenon was occurring was verified and is presented in Figure 4.9.

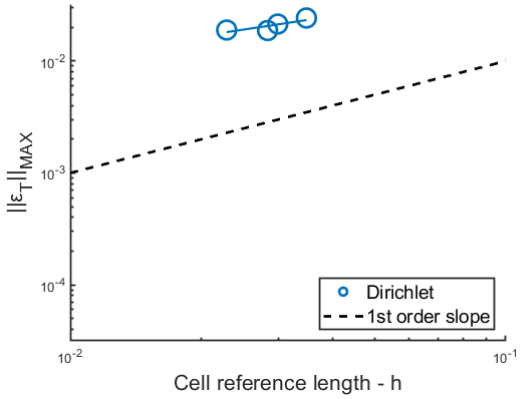


Figure 4.9: *Temperature maximum error for least squares interpolation method using Dirichlet boundary condition in hybrid meshes.*

It is verified that the maximum error remains mostly constant for all meshes, with a observed order of accuracy of only $0,3$. This shows that the error for the temperature field when using least squares interpolation method for Dirichlet boundary condition is following a much higher order of accuracy than the observed value of $1,59$, in almost the entire domain, except for the region on which the maximum error occurs which is responsible for this drop. Overall, the developed least squares interpolation method for thermal interpolations at the immersed boundary is robust and capable of handling hybrid meshes. A boundary condition of imposed temperature gradient (Neumann) appears to be the optimal way to impose boundary conditions using this method.

5 Heat Transfer on the Surface of a Cylinder

This chapter describes the simulation of heat transfer problem involving flow over a stationary cylinder using the immersed boundary method and the least squares interpolation method developed in Chapter 3 using *SOL*. A body fitted approach is also detailed and simulated using *SOL*, in an effort to compare the obtained data from the IB method to the more commonly used body fit approach.

The rate of heat transfer across the surface of the cylinder is studied and the results discussed in this chapter.

5.1 Definition of the physical problem

This section details the physical problem used in all simulations in Chapters 5 and 6.

The fluid domain comprises a rectangular geometry with one inlet on which flow enters the geometry with horizontal free stream velocity $U_0 = 1$, null vertical velocity and temperature $T_0 = 0$, and one outlet with $p = 0$ and null velocity derivatives. Both bottom and top walls have temperature $T_{WALL} = 0$ as a Dirichlet boundary condition and no permeability, this is, null vertical velocity. Figure 5.1 schematizes this problem. The computational domain size is 25,6 x 12,8 and the circle center is located at (6,4;6,4) and has diameter $D = 1$. [54]

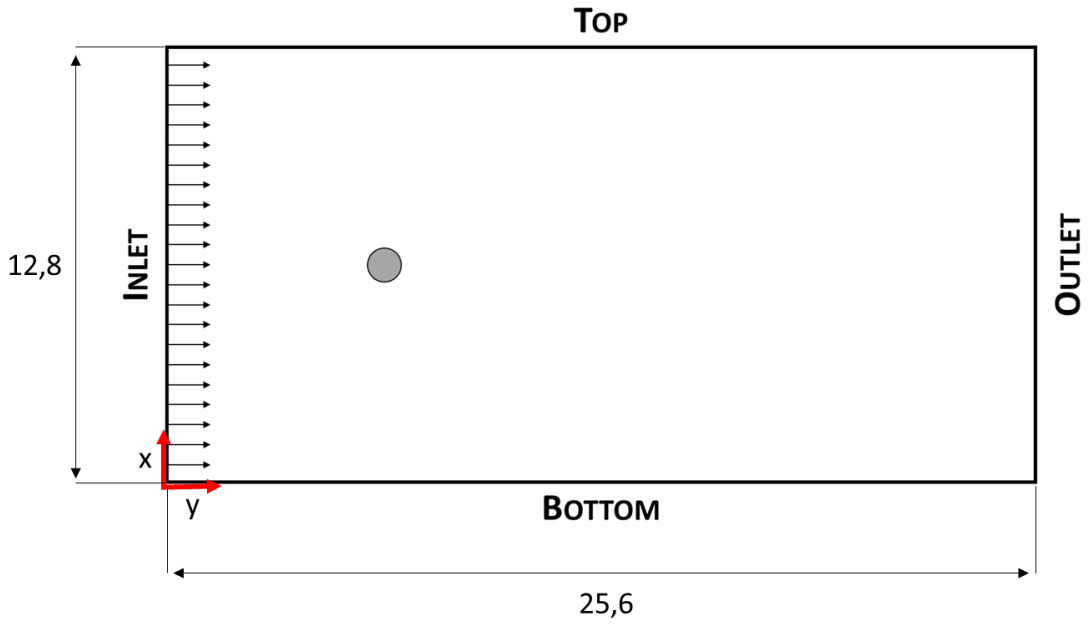


Figure 5.1: *Schematic of physical problem.*

All simulations conducted use $Pr = 0.7$ and $Re = 20$, where Prandtl and Reynolds number are, for this simulation, calculated as shown in Equations 41 and 42, as demonstrated by Zhang in [54].

$$Re = \frac{U_0 D}{\nu} \tag{41}$$

$$Pr = \frac{\nu}{\alpha} \tag{42}$$

In the following simulations in Chapters 5 and 6, when an isothermal boundary condition is applied

at the solid boundary, a condition of $T_{Solid} = 1$ is considered.

Regarding the iso-heat-flux boundary condition at the solid boundary, and following the work of Zhang2008 [54] where the heat flux is defined as shown in Equation 43, the temperature gradient imposed at the solid boundary is $\frac{dT}{dn} = -1$.

$$q = -(\frac{dT}{dn})_{IB} \tag{43}$$

5.2 Immersed Boundary Method Meshes

This Section details the meshes utilized in the simulations of the physical problem detailed in Section 5.1, when utilizing the Immersed Boundary method. All simulations using the IB method in Chapters 5 and 6 utilize these meshes.

All meshes considered have square elements of constant size across the entire mesh. Figure 5.2 shows the coarsest mesh considered. All boundaries are highlighted in blue, including the boundary created by the Immersed Boundary method. Figure 5.3 shows a detail of this same mesh, where the IB boundary is highlighted in blue and the solid boundary is highlighted in black.

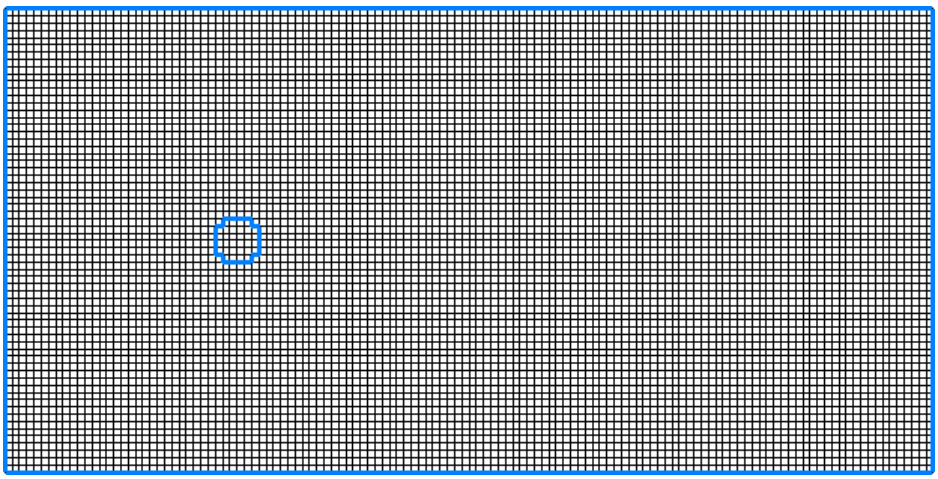


Figure 5.2: *Coarsest mesh utilized with highlighted boundaries.*

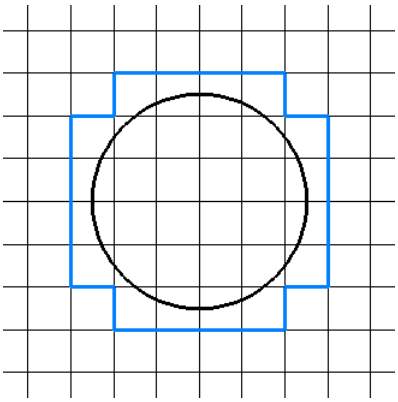


Figure 5.3: *Detail of coarsest mesh utilized where the IB boundary is highlighted in blue and the solid boundary is highlighted in black.*

Eight different refinement levels are considered. Table 5.1 shows the grid size, number of cells and overall domain size as a function of cells for all meshes.

Table 5.1: *Characteristics of meshes used for IB method.*

h	Length (number of cells)	Height (number of cells)	Total Domain (number of cells)	Diameter (number of cells)
0,20	128	64	8192	5,00
0,19	132	66	8712	5,16
0,10	256	128	32768	10,00
0,10	260	130	33800	10,16
0,08	316	158	49928	12,34
0,07	362	181	65522	14,14
0,06	422	211	89042	16,48
0,05	508	254	129032	19,84

5.3 Nusselt Sampling

In this physical problem, the most important variable analysed was the distribution of the Nusselt number over the cylinder surface. Figure 5.4 illustrates the definition of the referential used, using θ measured concentric with the cylinder. The Nusselt number variation over the cylinder's body is presented as a function of θ, a angle defined in the clockwise direction starting from the surface point closest to the inlet.

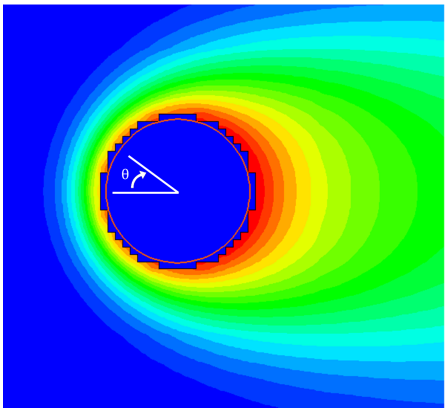


Figure 5.4: *Detail from the temperature field for a Dirichlet BC simulation showing the IB cut and the definition of angle* θ

The method employed to calculate the local Nusselt number differs depending on the boundary condition being imposed and the method, whether it is IB or bodyfitted. For the immersed boundary simulations, the calculations are performed as follows, as shown by Zhang in [54].

For the isothermal boundary condition, Nusselt is calculated using Equation 44 for each IB face.

$$Nu = -\frac{dT}{dn} \tag{44}$$

where n refers to the normal direction at the main solid point, pointing away from the solid body.

This temperature gradient is obtained after solving the matricial problem detailed in Section 3.2.2.2, which provides the full vector β. Considering that the coordinate system used is centered on the main solid point and as such $x = 0$ and $y = 0$, Equations 37 and 39 can be simplified to obtain Equations 45 and 46.

$$\frac{d\phi}{dx} = \beta_1 \tag{45}$$

$$\frac{d\phi}{dy} = \beta_2 \tag{46}$$

After obtaining the temperature gradient at the solid point, a vector dot product is then used to determine the gradient acting on the direction of the vector normal to the solid body, pointing outwards, as evidenced in Equation 47, thus allowing for the calculation of the Nusselt number.

$$\frac{dT}{dn} = \nabla T \cdot \vec{n} = (\beta_1; \beta_2) \cdot (n_x; n_y) \tag{47}$$

For the iso-heat-flux boundary condition, Nusselt is expressed using Equation 48.

$$Nu = \frac{1}{T} \tag{48}$$

Where T is the dimensionless temperature at the main solid point.

Analogously to the Dirichlet boundary case, this temperature is obtained after solving the matricial problem detailed in Section 3.2.2.2, which provides the full vector β. Since the referential used is centered at the main solid point, Equation 34 simplifies into Equation 49 allowing for the calculation of the Nusselt number.

$$\phi = \beta_0 \tag{49}$$

Section 6.1 will detail the use of a changed referential system centered on the IB face center for the Neumann interpolation mode, which requires the use of the full Equation 34 to find the temperature at the main solid point.

The following section demonstrates how Nusselt is calculated for Body Fitted mesh simulations.

5.4 Body Fit Approach

This section details the body fit approach utilized in this work. A body fit approach is the most used methodology for handling boundaries in CFD and is defined by having the mesh fitted and defined by the geometry boundaries themselves. Several different meshes using a body fit approach to the solid body are utilized, using varying degrees of refinement. This section details their geometry and how the necessary data, namely the Nusselt number, is locally calculated to provide a valid comparison to the IB method.

The body fitted meshes utilized follow the geometry shown in figure 5.5, where the solid boundary is represented by a mesh boundary, indistinguishable from the remaining mesh boundaries. The solid boundary conditions, such as non-permeability, no slip condition and thermal Dirichlet or Neumann are applied directly to this mesh boundary.

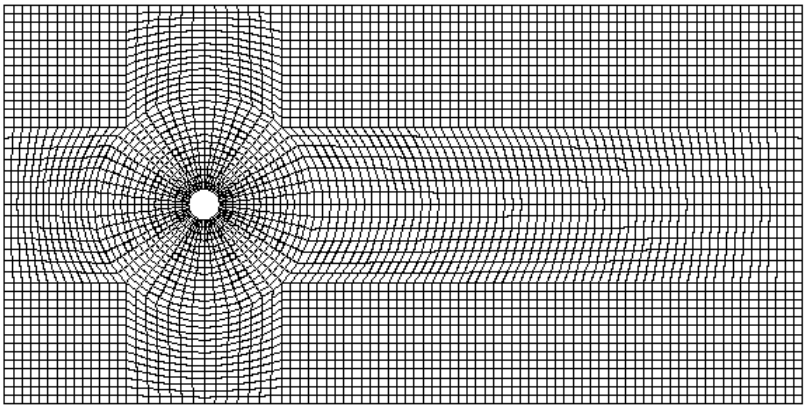


Figure 5.5: *Example of a Quadrilateral Mesh using a body fit approach to the cylinder boundary.*

Table 5.2 shows the different refinement levels considered for this mesh geometry, and the corresponding total number of cells.

Table 5.2: *Mesh size and characteristic length of all Body Fit meshes considered.*

h	Total number of cells
0,17	11194
0,14	16563
0,12	20562
0,08	47037
0,07	56680

A calculation of the local Nusselt number is performed in order to compare with the IB method results. Figure 5.6 schematizes the geometric points relevant to the calculation.

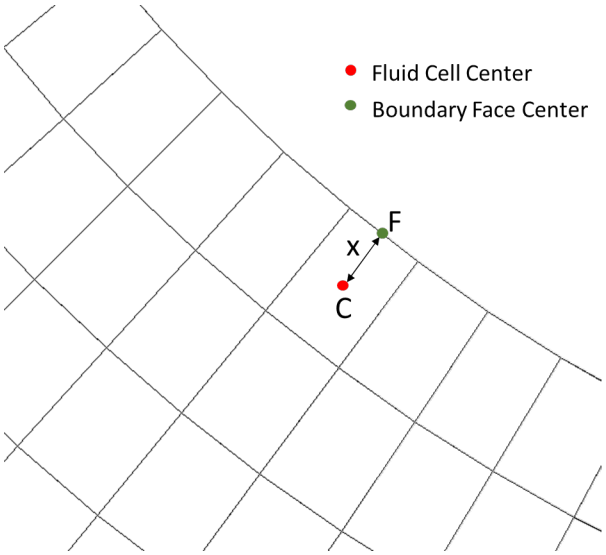


Figure 5.6: *Detail of a Quadrilateral Mesh using a body fit approach to the cylinder boundary.*

For the isothermal boundary condition, Nusselt is defined by Equation 44, where the derivative is estimated by utilizing the finite difference evidenced in Equation 50.

$$\frac{dT}{dn} = \frac{T_F - T_C}{d} \tag{50}$$

For the iso-heat-flux boundary condition, Nusselt is calculated using Equation 48, where the dimensionless temperature is calculated via Equation 51.

$$T = \frac{dT}{dn} d + T_C \tag{51}$$

5.5 Results

This section presents the results obtained for the simulations defined in Section 5. The results will be analysed separately when using Dirichlet or Neumann boundary conditions.

5.5.1 Dirichlet Results

This section presents the results obtained for the simulations defined in Section 5 using a Dirichlet boundary condition, this is, an isothermal condition imposed on the cylinder.

A refinement study was conducted which considered all grid sizes defined in Section 5.2 using the IB method. Figure 5.7 shows the resultant temperature field on the entire fluid domain for the IB mesh with $h = 0.1$.

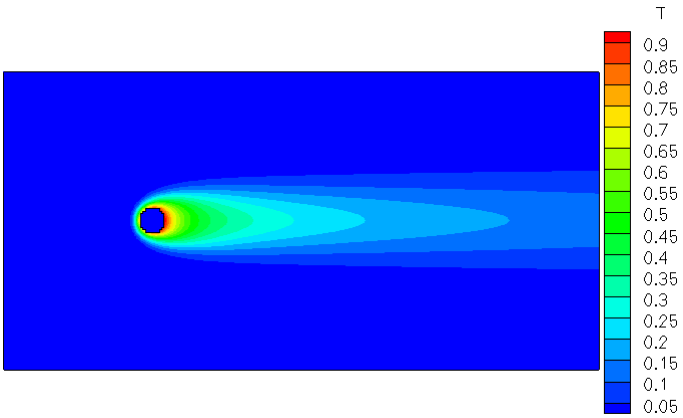


Figure 5.7: *Temperature field for whole domain with Dirichlet BC.*

Figure 5.8 shows a detail of the resultant temperature field around the cylinder for a Dirichlet boundary condition using the IB method, while Figure 5.9 shows the same detail of the resultant temperature field while using a body fit approach.

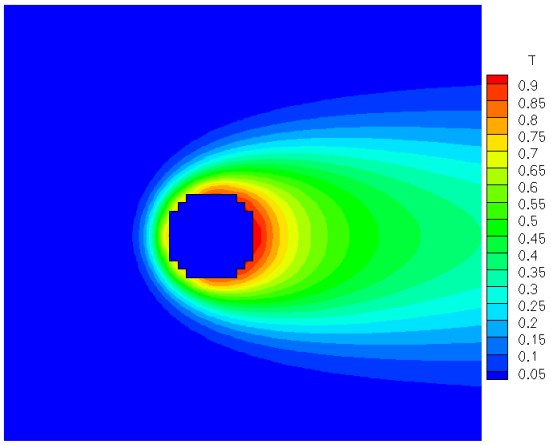


Figure 5.8: *Detail of temperature field for cylinder with Dirichlet BC using IB method and a mesh with $h = 0.1$.*

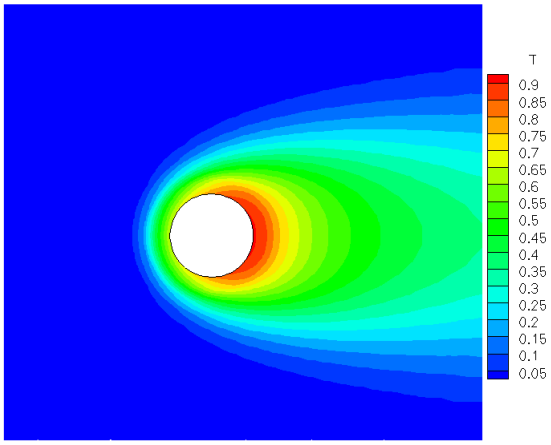


Figure 5.9: *Detail of temperature field for cylinder with Dirichlet BC using Body Fit method.*

Figure 5.10 and Figure Figure 5.11 refer to the Nusselt distribution for the coarsest and finest mesh considered for the IB method and the body fit method, respectively. Further results for simulations conducted on all mesh refinements considered can be found on Annex A.

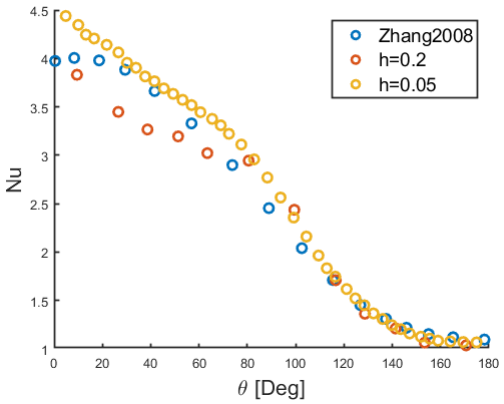


Figure 5.10: *Nusselt number distribution over surface of cylinder for the coarsest and most refined mesh using the IB method with a Dirichlet boundary condition.*

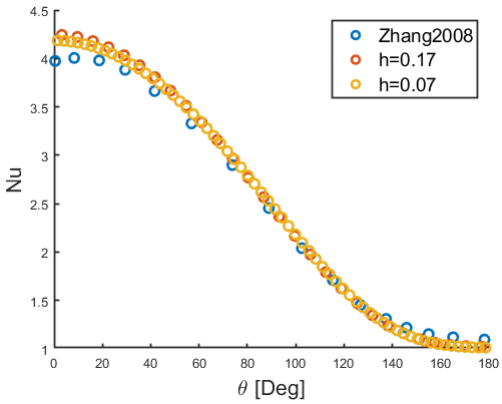


Figure 5.11: *Nusselt number distribution over surface of cylinder for the coarsest and most refined mesh using a body fit approach with a Dirichlet boundary condition.*

Figure 5.12 compares the results from the most refined meshes to the results obtained by Zhang in [54] and to results obtained in *SOL* using the body fit methodology presented in section 5.4.

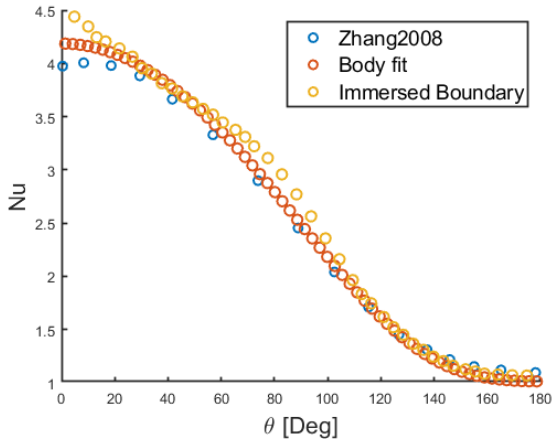


Figure 5.12: *Comparison of Nusselt number distribution over surface of cylinder using IB method, Body Fit method and literature data.*

5.5.2 Neumann Results

This section presents the results obtained for the simulations defined in Section 5 using a Neumann boundary condition, this is, a iso-heat-flux condition imposed at the solid boundary.

Figure 5.13 shows a detail of the resultant temperature field around the cylinder for a Neumann boundary condition using the IB method, while Figure 5.14 shows the same detail of the resultant temperature field while using a body fit approach.

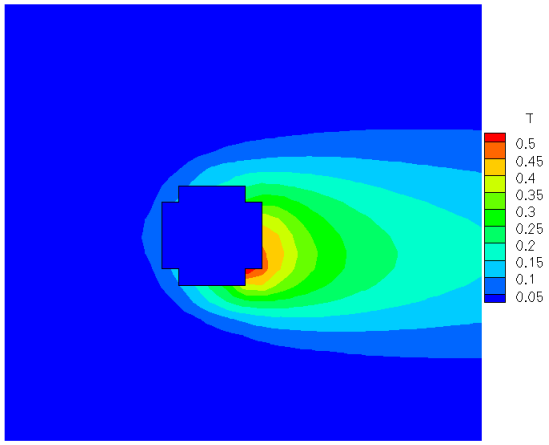


Figure 5.13: *Detail of temperature field for cylinder with Neumann BC using IB method with coarsest mesh.*

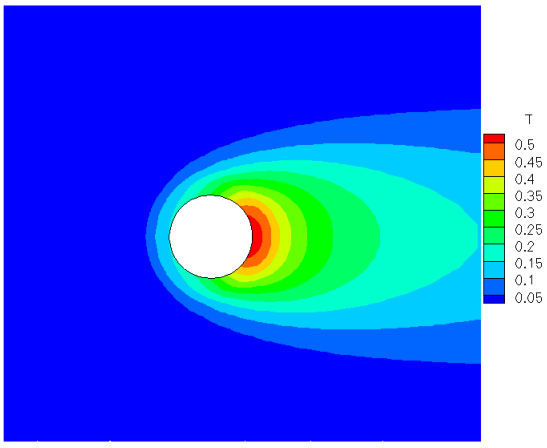


Figure 5.14: *Detail of temperature field for cylinder with Neumann BC using Body Fit method.*

Considering the unexpected nature of these results for all mesh refinements considered, no further results that utilize this Neumann interpolation method for the IB boundary are presented. These results indicate an error is occurring in the Neumann least squares interpolation method used. Chapter 6 addresses this limitation.

Figure 5.15 shows the Nusselt distribution for the coarsest and finest mesh considered utilizing the body fit method. Further results for simulations conducted on all mesh refinements considered can be found on Annex A.

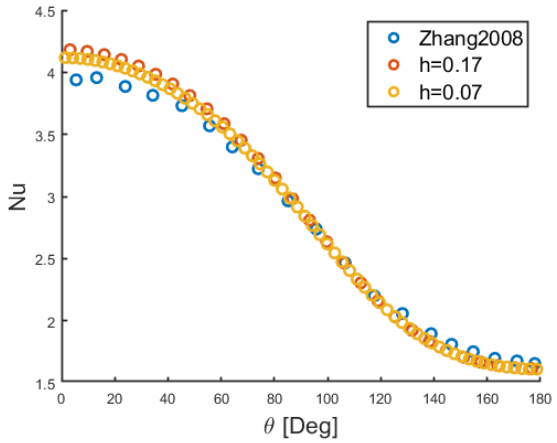


Figure 5.15: *Nusselt number distribution over surface of cylinder for the coarsest and most refined mesh using a body fit approach with a Neumann boundary condition.*

5.6 Discussion

The Nusselt number, a metric of the ratio of convective heat transfer over conductive heat transfer across a surface, is an adimensional number often used in solid-fluid heat transfer simulations. Using *SOL* and the methods developed in Chapter 3.2 it is possible to conduct heat transfer simulations of the flow over a cylinder with surface sampling of Nusselt numbers, allowing for a more in depth analysis. The physical problem as detailed in Section 5.1 represents the simulation chosen to study the behaviour of the presented interpolation methods. A simulation utilizing a standard body fit approach serves as a baseline comparison, as do the results obtained by Zhang in [54].

Considering the study of the Dirichlet least squares interpolation method, Figure 5.10 demonstrates a two fold behaviour for the isothermal simulations conducted. When considering the half of the cylinder facing the wake, a coarse mesh is sufficient for *SOL* to achieve data highly consistent with the numerical results from Zhang [54]. However, in the leading half of the cylinder and in particular at the leading stagnation point, the Nusselt number for both methods utilized in *SOL*, IB and bodyfit, overshoot the considered data by approximately 13% and 6%, respectively, representing a considerable deviation. This phenomenon can occur due to several factors, including the confining effects of the fluid domain. This hypothesis was tested by simulating the same exact flow case in a bigger computational domain. Figure 5.16 shows the comparison between the literature, and the two domain sizes considered.

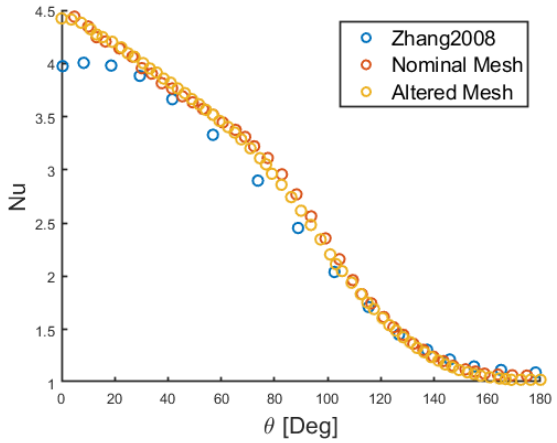


Figure 5.16: *Detail from the temperature field showing the IB cut and the definition of angle θ.*

No relevant variation was detected, reducing the possibility that confinement effects are the cause of the detected overshoot.

Considering that the deviation from the literature data occurs for both methods considered, the underlying cause cannot be easily attributed, and may be due to a difference in the geometry or boundary conditions of the domain imposed by Zhang in [54], to those imposed in this work.

It is known from the literature, including from the work done by Zhang in [54] that for the considered Reynolds number, $Re = 20$, the expected results have stationary wakes, resulting in a symmetric temperature field in regards to $y = 6.4$, as is verified for the Dirichlet results in Chapter 5.5.1. However, Figure 5.13 demonstrates that the interpolation method for Neumann boundary conditions does not produce a valid temperature field, since it does not in fact produce a symmetric temperature wake, revealing that the current interpolation method for Neumann boundary conditions in incapable of producing coherent results for the physical problem considered. The results for the Neumann boundary condition utilizing a body fit approach, presented in Figure 5.14 do demonstrate the correct behaviour. As such, modifications to the Neumann interpolation method were developed, implemented and tested, and are presented in Chapter 6.

6 Modified Neumann Interpolation Methods

This Chapter details the modifications applied to the interpolation method presented in section 3.2.2.2 in search of methodologies capable of adequately handling simulations such as the one presented in Chapter 5. All simulations used in this Chapter correspond to the physical problem described in Section 5.1

6.1 IB Face Centered Referential

This section details the first modification employed, a change of the coordinate system used in section 3.2.2.2. The previously defined method defines the main point in the stencil as the main solid point. This point is used as the origin of the referential, from which all distances are measured and, more importantly, it dictates the main directions, defined as it's normal and tangent vectors.

Figure 6.1 demonstrates the new coordinate system used, where $\overrightarrow{V_n}$ and $\overrightarrow{V_t}$ represent the base vectors of the new coordinate system, corresponding in the relevant equations to, respectively, x and y. Variables represented in this referential are identified by the subscript $Local$.

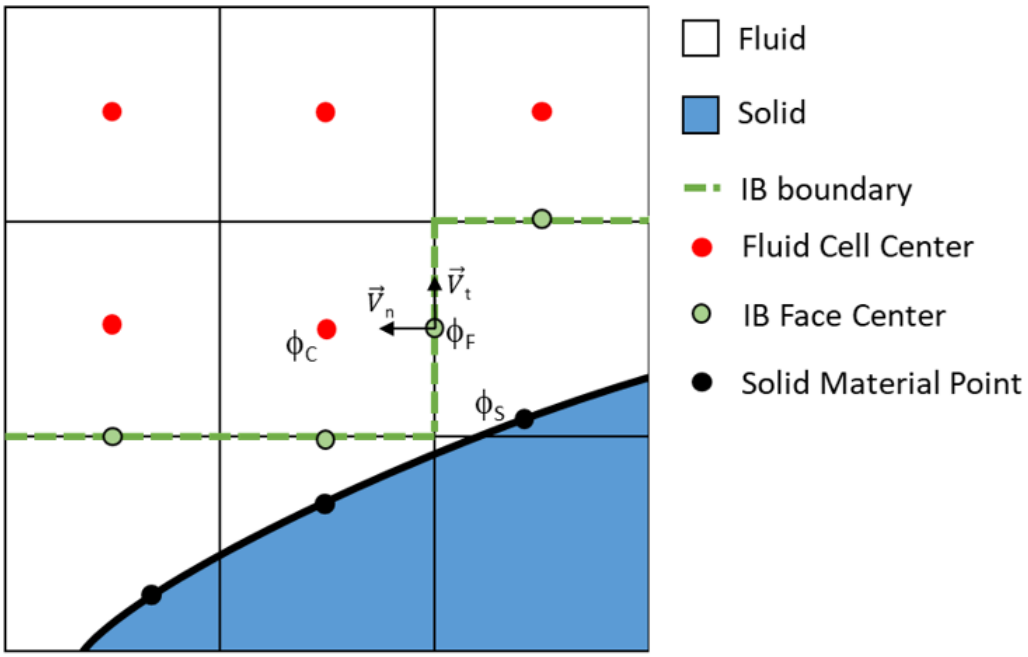


Figure 6.1: *Schematic of points and vectors for modified Neumann interpolation method.*

This reference system does not address the inherent error associated with assuming the solid point normal vectors and the IB face normal vector share the same direction. However, the main advantage in utilizing an IB face centered referential lies in the much simpler calculation of the desired variable: the temperature gradient at the IB face center. Since the referential used has the main direction defined by the normal unit vector to the IB face center, the gradient at this point is directly calculated utilising Equation 37, since direction x is the direction of the IB face center normal.

6.1.1 Results and Discussion

This section presents and discusses the results from the interpolation method proposed in Section 6.1.

Figure 6.2 shows the temperature field for the entire domain obtained by utilizing the new interpolation method, with the altered coordinate system, with a Neumann boundary condition.

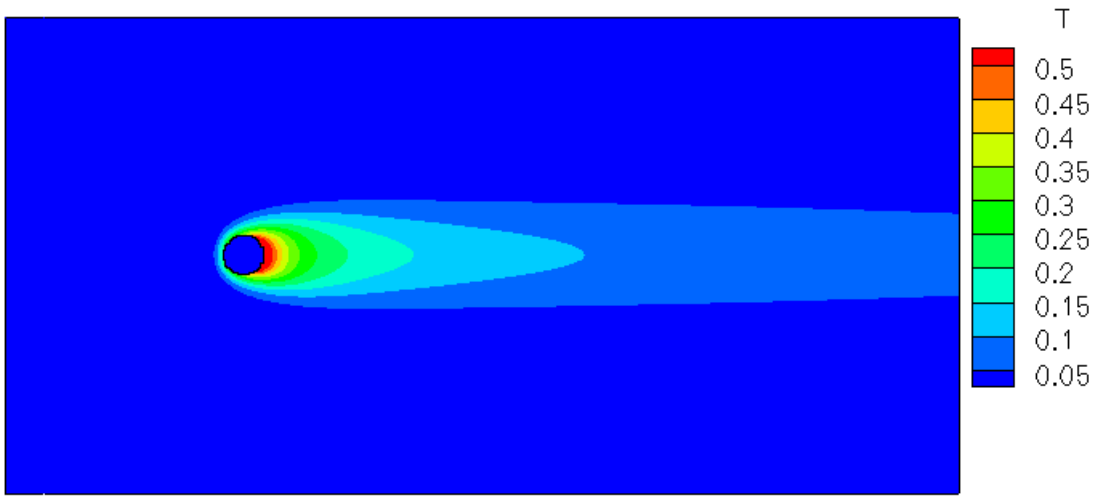


Figure 6.2: *Temperature field using a Neumann BC and Immersed Boundary method with the most refined mesh considered.*

Figure 6.3 shows a detail of the temperature field around the immersed boundary, utilizing the coarsest mesh considered. Figure 5.14 shows the same detail of the temperature field but obtained utilizing a body fit approach, as presented in the previous chapter. The obtained temperature field is macroscopically concordant with the solution obtained with the body fit approach, and as such the altered method proves much more capable of handling this geometry than was presented previously.

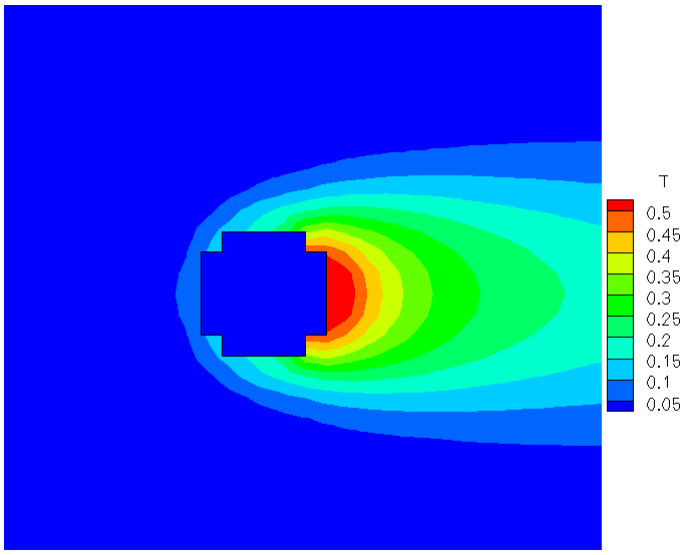


Figure 6.3: *Detail of temperature field utilizing a Neumann boundary condition with IB method.*

Figure 6.4 shows the Nusselt number distribution across the IB boundary on the coarsest and finest IB meshes considered and compares them to the literature data, while Figure 6.5 compares the results obtained utilizing both the IB method, the body fit method and the literature data. Further results for simulations conducted on all mesh refinements considered can be found on Annex A.

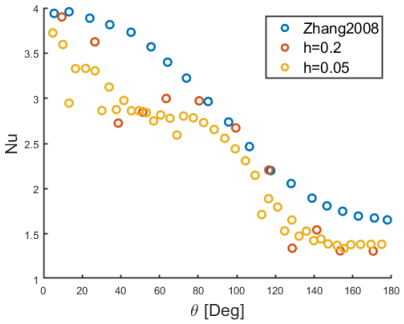


Figure 6.4: *Nusselt distribution using IB method with altered coordinate system.*

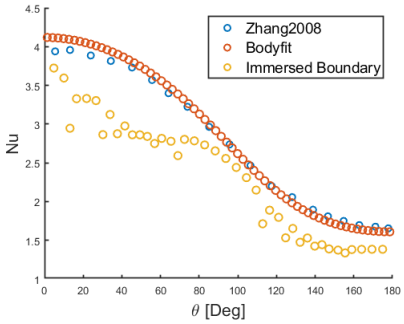


Figure 6.5: *Comparison of Nusselt distribution obtained using IB method, body fit approach and literature data.*

Despite the method proving capable of achieving a solution similar to that of the body fit approach, a more thorough analysis of the heat transfer across the surface of the cylinder, presented as the Nusselt distribution, demonstrates that even for the most refined meshes a substantial deviation from the expected results presented by the literature occurs. This method's inadequacy is attributed to the change of referential not addressing the variation of each solid point's normal vector, upon which the temperature gradient is imposed, and the IB face center normal, which defines the main direction of the stencil. This inadequacy of the interpolation method for Neumann boundary conditions is addressed in the following section.

6.2 Combination of Directional Derivatives

This Section details further modifications applied to the interpolation method presented in section 6.1. The new method is presented and the results are exposed and discussed. The need to compensate for the variation between the solid face normal vectors and the IB face center normal vector, which are assumed to be linear in the previous methodology, lead to the development of the methodology presented in this section. Previously, for each solid point introduced into the least squares problem, in the form of either Equation 37 or 39, a single direction is used to impose the temperature gradient. As such, each gradient occurs along either one of the main directions in the utilized coordinate system, which is rarely the case.

To address this limitation, a weighted combination of both derivatives is utilized. Each derivative's relative weight is defined as the value of the unit length vector upon which the temperature gradient occurs, when defined in the local coordinate system as defined in Section 6.1. Using this approach, each polynomial imposes the correct temperature gradient, at the correct relative location, imposed in the correct direction. Equation 52 defines this process.

$$\frac{dT}{d\overrightarrow{n}} = n_x \frac{d\phi}{dx} + n_y \frac{d\phi}{dy} \tag{52}$$

Where $\overrightarrow{n}$ is the unit length vector upon which the temperature gradient is applied. The variables n_x and n_y correspond to the coordinates of vector $\overrightarrow{n}$ in the local coordinate system as defined in Section 6.1, as shown in Equation 53.

$$\overrightarrow{n} = (n_x; n_y)_{Local} \tag{53}$$

For the solid points considered in this work, $\overrightarrow{n}$ is simply the outwards pointing unit length body normal.

Utilizing Equation 52 to define the temperature gradient contribution at each solid point, the new matrix $M_{Neumann}$ is defined in Equation 54, for a stencil with n fluid points and m solid points.

$$M_{Neumann} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2 y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & y_n^2 & x_n y_n \\ 0 & n_{x_{n+1}} h & n_{y_{n+1}} h & 2x_{n+1} n_{x_{n+1}} h & 2y_{n+1} n_{y_{n+1}} h & y_{n+1} n_{x_{n+1}} h + x_{n+1} n_{y_{n+1}} h \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & n_{x_{n+m}} h & n_{y_{n+m}} h & 2x_{n+m} n_{x_{n+m}} h & 2y_{n+m} n_{y_{n+m}} h & y_{n+m} n_{x_{n+m}} h + x_{n+m} n_{y_{n+m}} h \end{bmatrix} \tag{54}$$

6.2.1 Results and Discussion

This section presents the results obtained by utilizing the interpolation method presented in Section 6.2 for Neumann boundary conditions at the Immersed Boundary in simulations of the physical problem described in Section 5.1.

Macroscopically, the results obtained utilizing this interpolation method are practically indistinguishable from those obtained in the previous section. Figure 6.6 compares the Nusselt distribution in literature data with the distribution obtained using the IB method in *SOL* for three of the considered meshes. Further results for simulations conducted on all mesh refinements considered can be found on Annex A.

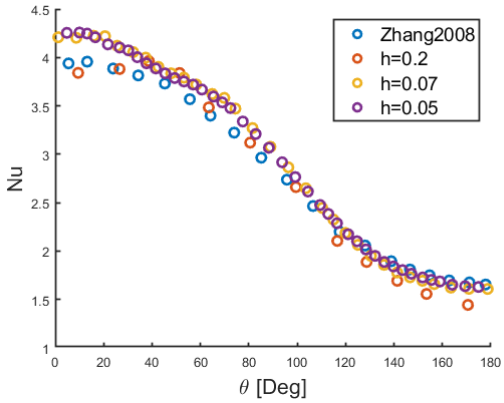


Figure 6.6: *Nusselt distribution along Immersed Boundary for 3 different mesh sizes utilizing linear combination of directional derivatives.*

The behaviour achieved with the addition of a combination of directional derivatives to the Neumann interpolation method grants a response of the local heat transfer across the surface of the cylinder with a much higher concordance to the literature data, even when considering lower refinement levels, when compared to the results presented in Figure 6.4, proving the postulated hypothesis that the deviation from the literature data and the body fit data was mostly due to the previous interpolation not addressing the difference in normal vectors direction.

Figure 6.7 compares the Nusselt distribution in literature data from Zhang2008 [54], with the distribution obtained using the IB method in *SOL* and with the results achieved using a body fit approach also using *SOL*.

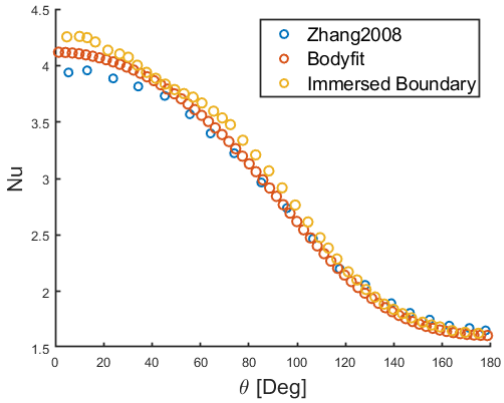


Figure 6.7: *Nusselt distribution comparison of the results from the IB method, a body fit approach and literature data.*

Figure 6.7 demonstrates that the Immersed Boundary results utilizing the combination of direction derivatives achieve a very high concordance with the results also obtained in *SOL* utilizing a body fit approach. However both these methodologies differ from the literature data at the leading half of the cylinder, culminating in the leading stagnation point, where a slight overshoot, of approximately 12% for the IB method and 8% for the body fit method, occurs. This effect is also observed for the Dirichlet results, and is believed to occur due to differing effects of confinement when comparing the simulation obtained in *SOL* from those conducted by Zhang in [54], be it due to different domain size, or slightly different boundary conditions.

The results indicate there are benefits in utilizing Neumann boundary conditions instead of Dirichlet. Figures 5.12 and 6.7 show a higher concordance with the data obtained with the body fit methodology when imposing heat flux. As is mentioned in Section 3.3, in terms of the bulk solver (fluid region), the gradient can immediately be imposed in the diffusive flux at the face, while the temperature requires an additional calculation.

A mesh refinement study, presented in Appendix B, is also conducted utilizing this Neumann interpolation method and the Taylor-Couette flow simulation detailed in Section 3.1 and radial quadrilateral meshes. The observed order of accuracy is measured at $1,82$, verifying the developed method. However, it is lower than the expected value of 2 and represents a deviation of 9%.

7 Conclusions and Future Work

The present work focused on the thermal simulation capabilities of *SOL* when using the immersed boundary method. In particular, the work focused on the interpolation of the relevant variables from the original solid boundary to the IB boundary. After this work, *SOL* is capable of handling both imposed temperature and imposed heat flux temperature boundary conditions across physical problems of varying complexity and using different mesh geometries.

Both linear and least squares interpolation methods were verified in both boundary cases using analytical data for low-Reynolds 2D Taylor-Couette flow problems with heat transfer, while also verifying their theoretical order of accuracy.

An extensive study of the developed methods was conducted using various types of structured grids. The code showed robustness to variable mesh geometries for both methods using Dirichlet and Neumann boundary conditions.

Several simulations using hybrid meshes for Taylor-Couette flow problems are also presented, revealing an inability of the linear interpolation method in handling grids with arbitrary connectivity. The same methodology was also used to demonstrate the least squares interpolation method's ability to handle these mesh geometries.

The least squares interpolation method was used for flow over a cylinder with heat transfer and an algorithm was implemented to plot the Nusselt number along the surface of the solid boundary. The same physical problem was also simulated using a body fit methodology, to obtain data also provided by *SOL* to use as benchmark for the IB method's results.

The Dirichlet boundary condition proved capable of achieving results that were in accordance with both the literature and the body fit approach. However, this study revealed that the least squares interpolation method for Neumann boundary conditions was not capable of handling this more complex simulation. This Neumann interpolation method limitation was addressed in two ways: by changing the coordinate system used while constructing the stencil, and by enabling the imposition of temperature gradients upon any direction by utilizing a combination of directional derivatives. These alterations proved capable of resolving the method limitations discovered, with the Neumann interpolation method achieving results with high concordance with the literature data and with the body fit approach. Simulations with imposed temperature and imposed heat flux in more complex geometries and flows would further verify the software used and would be of utmost importance.

In the process of understanding the existing interpolation methods and implementing the new one, comments were made in the presented interpolation methods in order to facilitate future work and allow for a better/faster understanding of the code by a new user. Hence, this work also contributes to the future developments of SOL.

Overall, *SOL* was verified using analytical data for heat transfer Taylor-Couette flow simulations using the immersed boundary method for up to second order across both Dirichlet and Neumann boundary conditions, opening the possibility for future work to implement fully conjugate heat transfer problems.

Further analysis of the Nusselt distributions along the solid cylinder surface in IB and body fit simulations could help find the reason for the differing Nusselt overshoot at the leading stagnation point and as such could prove the be valuable future work.

A combined study of the thermal results of the physical problem detailed in Chapter 5 and the viscous force locally sampled in the same manner around the cylinder is also now possible, and would

further expand the work of the IB method implementation in *SOL*.

A study on the implications of imposing different types of boundary conditions on the IB faces from those present in the solid body would also be promising, since the results present in this work show apparent benefits in utilizing Neumann boundary conditions, as was discussed in Sections 3.3 and 6.2.1. As such, there may be advantages to imposing heat flux at the IB face even when temperature is imposed at the solid boundary.

Such continuations of this work would enable *SOL* to test a wider range of problems, for example a simulation of oscillating heated solid bodies inside a stationary fluid domain, a difficult problem to simulate without IB method due to the moving nature of the solid body.

The current work furthers the capabilities of *SOL* such that future work may continue to expand upon the IB method range of operations.

References

[1] Frank M. White, “Fluid Mechanics,” 2011.

[2] J. Blazek, “Computational fluid dynamics: principles and applications,” 2015.

[3] Hyoung Woo Oh, “Applied Computational Fluid Dynamics,” 2012.

[4] “Web of Science - CFD & IB Method Research,” 2022.

[5] T. Kajishima and K. Taira, “Immersed Boundary Methods,” *Computational Fluid Dynamics*, 2017.

[6] C. S. Peskin, “Flow Patterns Around Heart Valves: A Numerical Method,” *Journal of Computational Physics*, vol. 10, 1972.

[7] C. S. Peskin, “The immersed boundary method,” *Acta Numerica*, vol. 11, jan 2002.

[8] S. Roy, A. De, and E. Balaras, “Immersed Boundary Method: Development and Applications,” 2020.

[9] S. Péron, T. Renaud, M. Terracol, C. Benoit, and I. Mary, “An immersed boundary method for preliminary design aerodynamic studies of complex configurations,” *23rd AIAA Computational Fluid Dynamics Conference, 2017*, 2017.

[10] J. P. Johnson, G. Iaccarino, K. H. Chen, and B. Khalighi, “Simulations of high reynolds number air flow over the NACA-0012 airfoil using the immersed boundary method,” *Journal of Fluids Engineering, Transactions of the ASME*, vol. 136, no. 4, 2014.

[11] R. Ghias and A. Khondge, “Implementation of Immersed Boundary Method for Rapid and Reliable External Flow Simulations in Automotive,” 2009.

[12] O. Antepara, R. Borrell, O. Lehmkuhl, I. Rodríguez, and A. Oliva, “Parallel Adaptive Mesh Refinement of Turbulent Flow Around Simplified Car Model Using an Immerse Boundary Method,” *11th World Congress on Computational Mechanics (WCCM XI) 5th European Conference on Computational Mechanics (ECCM V) 6th European Conference on Computational Fluid Dynamics (ECFD VI)*, 2014.

[13] D. T. Hoang, S. V. Pham, K. N. Tran, C. D. Nguyen, and K. P. Nguyen, “Aeroelastic analysis on wing structure using immersed boundary method,” *Lecture Notes in Mechanical Engineering*, 2018.

[14] T. Behrens, “Simulation of Moving Trailing Edge Flaps on a Wind Turbine Blade using a Navier-Stokes based Immersed Boundary Method,” *DTU Mechanical Engineering*, 2011.

[15] S. Jafari, N. Chokani, and R. S. Abhari, “Terrain Effects on Wind Flow: Simulations with an Immersed Boundary Method,” 2011.

[16] S. Jafari, N. Chokani, and R. S. Abhari, “An immersed boundary method for simulation of wind flow over complex terrain,” *Journal of Solar Energy Engineering, Transactions of the ASME*, vol. 134, no. 1, 2012.

[17] F. Sbrizzai, V. Lavezzo, R. Verzicco, M. Campolo, and A. Soldati, “Direct numerical simulation of turbulent particle dispersion in an unbaffled stirred-tank reactor,” *Chemical Engineering Science*, vol. 61, no. 9, 2006.

[18] R. Verzicco, M. Fatica, G. Iaccarino, and P. Orlandi, “Flow in an impeller-stirred tank using an immersed-boundary method,” *AIChE Journal*, vol. 50, jun 2004.

[19] Y. Guo, C. Y. Wu, and C. Thornton, “Modeling gas-particle two-phase flows with complex and moving boundaries using DEM-CFD with an immersed boundary method,” *AIChE Journal*, vol. 59, mar 2013.

[20] M. Miyoshi, T. Ishii, A. Hashimoto, K. Mori, and Y. Nakamura, “Computational Analysis of Parachute Motions Using the Immersed Boundary Method,” 2008.

[21] Y. Kim and C. S. Peskin, “2-D parachute simulation by the immersed boundary method,” *SIAM Journal on Scientific Computing*, vol. 28, no. 6, pp. 2294–2312, 2006.

[22] F. Favre, O. Antepara, C. Oliet, O. Lehmkuhl, and C. D. Perez-Segarra, “An immersed boundary method to conjugate heat transfer problems in complex geometries. Application to an automotive antenna,” *Applied Thermal Engineering*, vol. 148, feb 2019.

[23] J. Alexandre, “A 2-D Immersed Boundary Method on Low Reynolds Moving Body,” *MSc thesis*, 2014.

[24] D. Martins, “On the suppression of spurious pressure oscillations in immersed boundary methods with unstructured grids,” *MSc thesis*, 2016.

[25] H. Daniel, “Error Estimation Criteria to Couple the Immersed Boundary Method with an Automated Adaptive Grid Algorithm,” *MSc thesis*, 2017.

[26] R. Singh, “CFD simulation of NASCAR racing car aerodynamics,” *SAE Technical Papers*, vol. 2008, no. 724, 2008.

[27] S. Thabet and T. H. Thabit, “CFD Simulation of the Air Flow around a Car Model (Ahmed Body),” *International Journal of Scientific and Research Publications (IJSRP)*, vol. 8, no. 7, 2018.

[28] R. B. Sharma, “CFD Simulation for Flow over Passenger Car Using Tail Plates for Aerodynamic Drag Reduction,” *IOSR Journal of Mechanical and Civil Engineering*, vol. 7, no. 5, 2013.

[29] K. Gore, A. Gote, A. Govale, A. Kanawade, and S. Humane, “Aerodynamic Analysis of Aircraft Wings Using CFD,” 2018.

[30] A. Consulting and S. Peter, “Mathematical model of hot-air balloon steady-state vertical flight performance,” vol. 25, no. 3, 2021.

[31] J. Loganathan, K.-M. Lim, H. P. Lee, and B. C. Khoo, “Computational Fluid Dynamics Study of Balloon System Tethered to a Stratosail,” *Journal of High Altitude Ballooning*, vol. 1, no. 1, 2021.

[32] D. Šimurda, L. Popelka, L. Zelený, and M. Matějka, “Wing Body Interaction : Cfd and in-Flight Testing,” *The 19th International Symposium on Transport Phenomena*, no. December 2013, 2008.

[33] P. Witting, “Computational fluid dynamics in a traditional animation environment,” *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999*, 1999.

[34] R. Mittal, H. Dong, M. Bozkurttas, A. Von Loebbecke, and F. Najjar, “Analysis of flying and swimming in nature using an immersed boundary method,” *Collection of Technical Papers - 36th AIAA Fluid Dynamics Conference*, vol. 1, no. June, 2006.

[35] Y. Tamaki and T. Imamura, “Turbulent flow simulations of the common research model using immersed boundary method,” *AIAA Journal*, vol. 56, no. 6, 2018.

[36] R. Mittal, H. Dong, M. Bozkurttas, F. M. Najjar, A. Vargas, and A. von Loebbecke, “A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries,” *Journal of Computational Physics*, vol. 227, no. 10, 2008.

[37] B. Blais, M. Lassaigne, C. Goniva, L. Fradette, and F. Bertrand, “A semi-implicit immersed boundary method and its application to viscous mixing,” *Computers and Chemical Engineering*, vol. 85, 2016.

[38] A. Cervone, A. Chierici, L. Chirco, R. Da Via, V. Giovacchini, and S. Manservisi, “CFD simulation of turbulent flows over wire-wrapped nuclear reactor bundles using immersed boundary method,” *Journal of Physics: Conference Series*, vol. 1599, no. 1, 2020.

[39] W. J. Zhu, T. Behrens, W. Z. Shen, and J. N. Sørensen, “Hybrid immersed boundary method for airfoils with a trailing-edge flap,” *AIAA Journal*, vol. 51, no. 1, 2013.

[40] I. Sadrehaghighi, “Mesh Generation in CFD,” 2020.

[41] J. E. Castillo and Society for Industrial and Applied Mathematics., “Mathematical aspects of numerical grid generation,” 1991.

[42] R. Peyret, “Handbook of Computational Fluid Mechanics,” 1996.

[43] J. Sack and J. Urrutia, “Handbook of computational geometry,” 2000.

[44] F. Juretić and A. D. Gosman, “Error analysis of the finite-volume method with respect to mesh type,” *Numerical Heat Transfer, Part B: Fundamentals*, vol. 57, no. 6, 2010.

[45] H. Jasak, *Error Analysis and Estimation for the finite volume method with applications to fluid flows.* PhD thesis, 1996.

[46] J. P. Magalhaes, *An adaptive framework for the numerical simulation of environmental flows in ˜ complex geometries*. PhD thesis, 2011.

[47] C. Santarelli, T. Kempe, and J. Fröhlich, “Immersed boundary methods for heat transfer,” *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 26, mar 2016.

[48] R. J. LeVeque, “Finite Difference Methods for Ordinary and Partial Differential Equations,” 2007.

[49] P. G. Ciarlet, “The Finite Element Method for Elliptic Problems,” 2002.

[50] J. C.Strikwerda, “Finite Difference Schemes and Partial Differential Equations,” 2004.

[51] K. Nagendra, D. K. Tafti, and K. Viswanath, “A new approach for conjugate heat transfer problems using immersed boundary method for curvilinear grid based solvers,” *Journal of Computational Physics*, vol. 267, 2014.

[52] R. Kedia, M. L. Hunt, and T. Colonius, “Numerical simulations of heat transfer in taylor-couette flow,” *Journal of Heat Transfer*, vol. 120, no. 1, 1998.

[53] T. Kariya and H. Kurata, “Generalized Least Squares,” *Generalized Least Squares*, vol. 7, 2004.

[54] N. Zhang, Z. C. Zheng, and S. Eckels, “Study of heat-transfer on the surface of a circular cylinder in flow using an immersed-boundary method,” *International Journal of Heat and Fluid Flow*, vol. 29, no. 6, 2008.

Appendices

A Nusselt distributions for all IB meshes

The results presented show the Nusselt distribution on the cylinder boundary for all meshes simulated for each respective method and boundary condition. Figure A.1 refers to the results obtained using the IB method with a Dirichlet boundary condition, Figure A.2 also refers to a Dirichlet boundary condition but utilizing a body fit approach. Figures A.3 , A.4 and A.5 all refer to simulations with Neumann boundary condition, respectively utilizing a body fit approach, an IB method with altered coordinate system and an IB method with combination of directional derivatives.

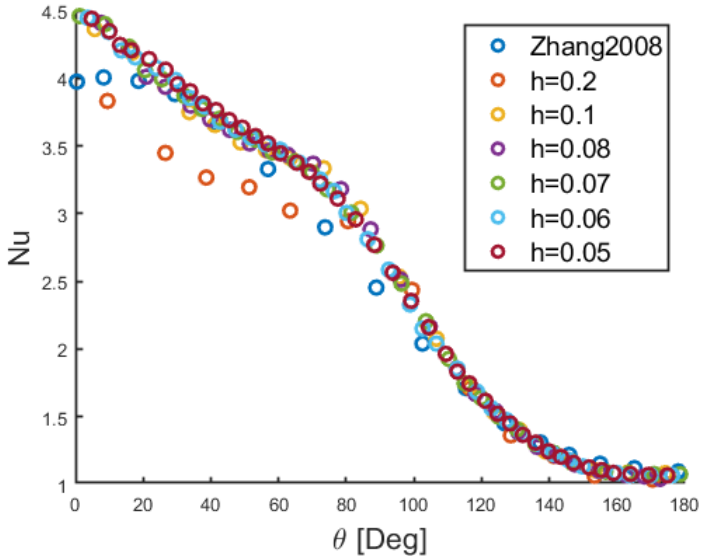


Figure A.1: *Nusselt distribution along Immersed Boundary for all meshes using Dirichlet boundary condition.*

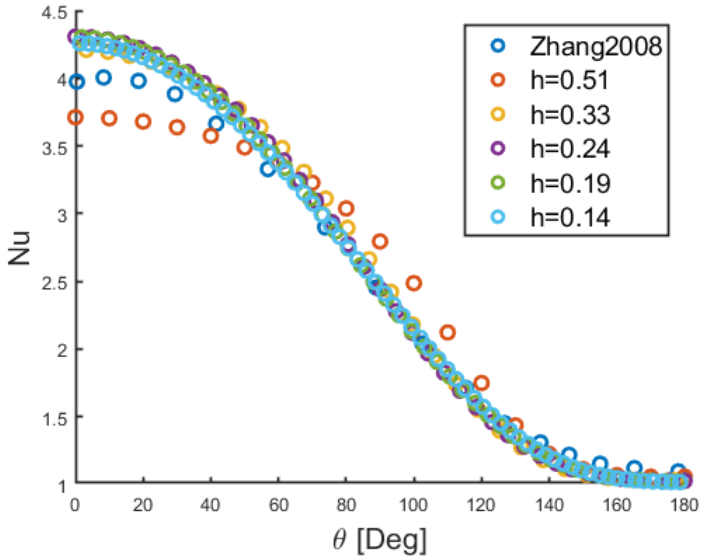


Figure A.2: *Nusselt distribution for all meshes used in body fit approach with Dirichlet boundary condition.*

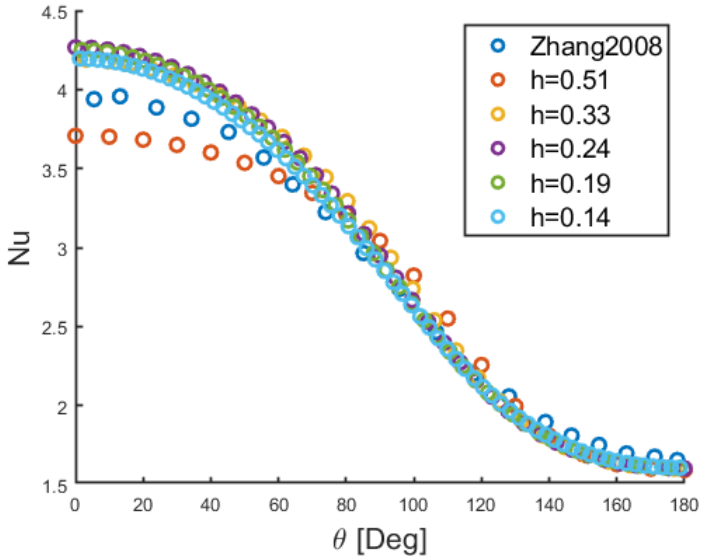


Figure A.3: *Nusselt distribution for all meshes used in body fit approach with Neumann boundary condition.*

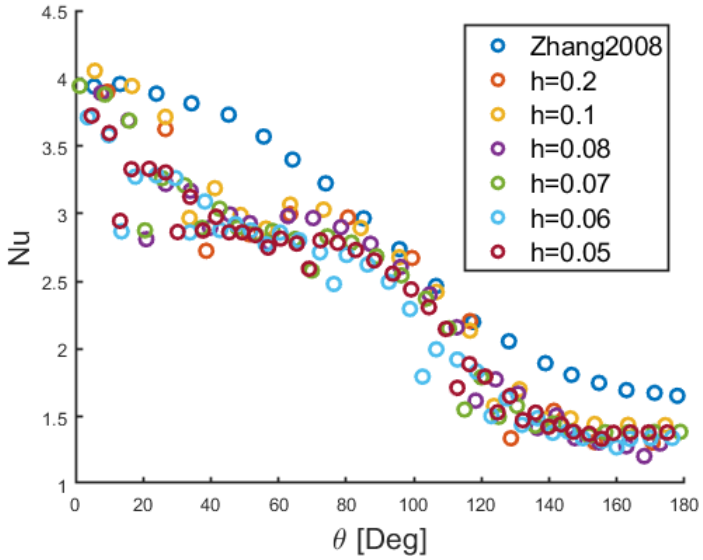


Figure A.4: *Nusselt distribution along Immersed Boundary for all meshes using Neumann boundary condition with alternate coordinate system.*

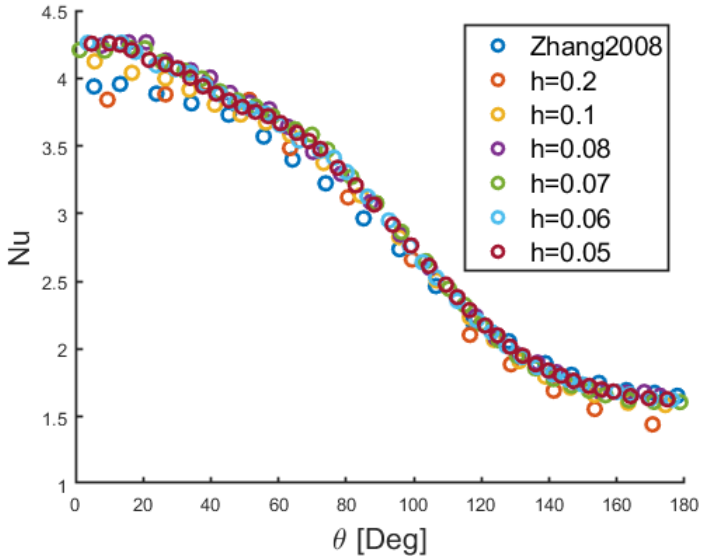


Figure A.5: *Nusselt distribution along Immersed Boundary for all meshes using Neumann boundary condition with alternate coordinate system and combination of directional derivatives.*

B Verification of Neumann interpolation method with combination of directional derivatives

Figure B.1 presents the mesh refinement study for a Taylor-Couette simulation using the modified Neumann interpolation method. For this study 3 different radial quadrilateral meshes are used.

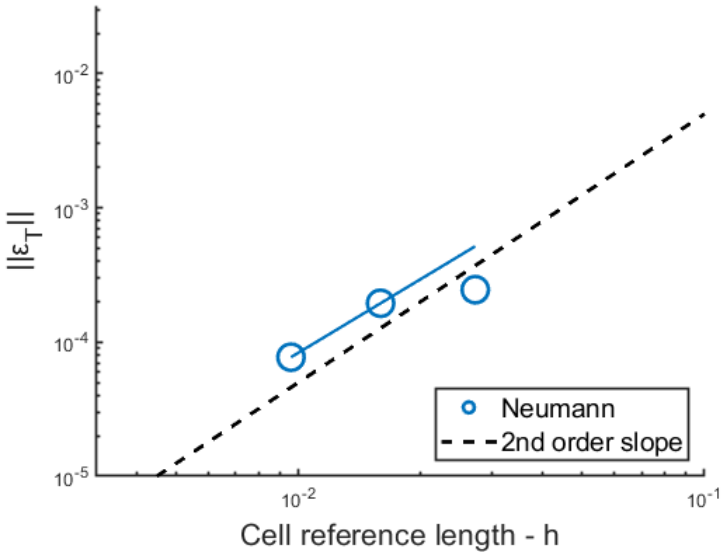


Figure B.1: *Error decay for Taylor-Couette simulation using Neumann boundary condition with alternate coordinate system and combination of directional derivatives.*