

Error Estimation Criteria to Couple the Immersed Boundary Method with an Automated Adaptive Grid Algorithm

Hugo Daniel da Silva Domingos

Thesis to obtain the Master of Science Degree in
Aerospace Engineering

Supervisors: Prof. José Carlos Fernandes Pereira
Dr. Duarte Manuel Salvador Freire Silva de Albuquerque

Examination Committee

Chairperson: Dr. Filipe Szolnoky Ramos Pinto Cunha
Supervisor: Dr. Duarte Manuel Salvador Freire Silva de Albuquerque
Member of the Committee: Dr. José Manuel da Silva Chaves Ribeiro Pereira

May 2017

Acknowledgments

Firstly, I would like to thank Professor José Carlos Pereira, one of the Thesis' supervisors, without whom I wouldn't have been able to work on a scientific field that brought me great pleasure.

I would also like to thank Doctor Duarte Albuquerque who had a key role in helping me through the process of completing this Thesis, taking the time to explain, aguide me and clarify any doubts I had.

To all my family and friends, that were essential, giving me support whenever I needed throughout my life and without whom I would not be here today.

A special thanks to the mosqueteiros, for being such a special group that kept together for all these years.

Resumo

Neste trabalho foram verificados e validados os resultados do Método de Fronteira Imersa em casos bi-dimensionais de escoamento incompressível e foi implementado um algoritmo automático de malhas adaptativas baseado em estimadores de erro. Foram usados dois estimadores de erro, o critério de Taylor e o de Resíduo.

Estudou-se um escoamento em torno de um cilindro imerso num fluído permitindo assim uma comparação directa entre os resultados obtidos e os de outros trabalhos, estudando-se a influência do tamanho do domínio. Foi também estudado um caso com solução analítica que permitiu a comparação entre três pares de esquemas numéricos.

Após concluídas as etapas de verificação e de perceber a influência dos esquemas numéricos nos resultados, foi altura de testar o método com o algoritmo de malhas adaptativo em casos sem solução analítica. Para isso, utilizou-se novamente o caso do cilindro imerso num escoamento, com três números de Reynolds diferentes. Revisitar este caso permitiu também perceber se a implementação do algoritmo de refinamento adaptativo aumentou a eficiência relativamente ao critério uniforme e também perceber a precisão dos resultados quando comparados com os obtidos anteriormente. Testou-se também o método num caso de um perfil NACA 0012.

Os resultados obtidos confirmam que a implementação destes estimadores de erro reduz muito o poder computacional necessário para obter resultados com uma maior precisão.

Palavras-chave: Método da Fronteira Imersa, Escoamento Incompressível, Gerador malhas adaptativo, estimadores de erro, critério de Taylor, critério do Resíduo

Abstract

In this work, the results of the Immersed Boundary Method were validated and verified for bi-dimensional cases of incompressible flow. An automated adaptive grid algorithm based on error estimators was also tested. Two different error estimators were used, the Taylor criterion and the Residual one.

A study of an immersed cylinder in a flow was also performed, allowing for a comparative study between the obtained results with others from the literature, allowing to understand how the domain size influences the numerical results due to the blockage effect. A case with an analytical solution was also studied, which was ideal for the comparison of three different pairs of numerical schemes and to choose the best one to be used throughout the rest of the work.

After completing the verification of the method and the influence of numerical schemes on the results' accuracy, the IBM coupled with the automated adaptive grid algorithm had to be tested in cases with no analytical solution. To achieve this, the immersed cylinder case was studied once again, demonstrating the computational power savings that the adaptive grid brought when compared to the uniform refinement case. Finally, this method was also tested in a NACA 0012 airfoil.

By comparing the different results it was possible to conclude that the error estimators implementation would effectively reduce the computational power needed to achieve accurate results.

Keywords: Immersed Boundary Method, Incompressible flow, Adaptive grid generator, Error estimators, Taylor criterion, Residual criterion

Contents

Acknowledgments	iii
Resumo	v
Abstract	vii
List of Tables	xi
List of Figures	xiii
Nomenclature	xvii
Glossary	xix
1 Introduction	1
1.1 Motivation	1
1.2 Historical context	2
1.3 Immersed Boundary Method	3
1.3.1 Continuous forcing approach	4
1.3.2 Discrete forcing approach	6
1.3.3 Applied Method	9
1.4 Error Estimators	9
1.5 SOL code	10
1.6 Objectives	11
1.7 Thesis Outline	11
2 Numerical Methods	13
2.1 Governing Equations	13
2.2 Finite Volume Discretization	14
2.2.1 Convective Scheme - Structured grid	15
2.2.2 Diffusive Scheme - Structured grid	16
2.2.3 Cell Centred Gradient Scheme	16
2.2.4 Weighted Least Squares Method	17
2.2.5 Face WLS - Adaptive refinement criteria	18
2.3 Coupling Velocity and Pressure	19
2.3.1 SIMPLE	19
2.4 IBM algorithm	21

2.5	Calculating the Viscous and Pressure Forces	21
2.6	Error Estimation Criteria	22
2.6.1	Taylor criterion	22
2.6.2	Residual criterion	23
2.7	Adaptive meshing algorithm	24
3	Verification and Validation	27
3.1	Flow Around a 2D Cylinder	27
3.1.1	Computational Domain Confinement Issue - Blockage Effect	28
3.1.2	Geometry and Domain Characteristics	30
3.1.3	Immersed Boundary Method Grids and Convergence Study	31
3.1.4	Body-fit Grids and Convergence Study	33
3.1.5	Reynolds 10	35
3.1.6	Reynolds 20	36
3.1.7	Reynolds 40	37
3.1.8	2D Cylinder - Discussion	38
3.2	Analytical solution with Immersed Boundary Method and adaptive refinement	39
3.2.1	Analytical Singularity	39
3.2.2	Analytical singularity - Discussion	45
4	IBM with automatic adaptive refinement	49
4.1	Flow Around a 2D Cylinder with adaptive criteria	49
4.1.1	Reynolds 10	50
4.1.2	Reynolds 20	53
4.1.3	Reynolds 40	55
4.1.4	2D Cylinder with adaptive criteria	58
4.2	Flow Around the NACA 0012 airfoil	58
4.2.1	NACA 0012 - Discussion	67
5	Conclusions	73
5.1	Achievements	73
5.2	Future Work	74
Bibliography		75
A Literature Table		79

List of Tables

3.1	Drag coefficient with the respective viscous and pressure components variation with the blockage effect	30
3.2	2D Cylinder Domain Dimensions.	30
3.3	Reynolds and kinematic viscosity values	31
3.4	IBM Mesh refinement	32
3.5	IBM convergence study	33
3.6	Body-fit Mesh refinement	34
3.7	C_D results for the different body-fit grids.	35
3.8	Results from literature survey and from the present work for $Re = 10$	36
3.9	Results from literature survey and from the present work for $Re = 20$	37
3.10	Results from literature survey and from the present work for $Re = 40$	38
3.11	Analytical Singularity - Error decay with Uniform Refinement	40
3.12	Analytical Singularity - Error decay with Analytical criterion.	41
3.13	Analytical Singularity - Error decay with Taylor criterion.	43
3.14	Analytical Singularity - Error decay with Residual criterion.	44
3.15	Analytical Singularity: Efficiency Values for the most refined grid of each criteria.	46
4.1	$Re = 10$ - Convergence study for the Taylor adaptive criterion.	51
4.2	$Re = 10$ - Convergence study for the Residual criterion.	52
4.3	$Re = 20$ - Convergence study for the Taylor criterion.	53
4.4	$Re = 20$ - Convergence study for the Residual criterion.	54
4.5	$Re = 40$ - Convergence study for the Taylor criterion.	56
4.6	$Re = 40$ - Convergence study for the Residual criterion.	57
4.7	NACA 0012 results for $\alpha = 0^\circ$	61
4.8	NACA 0012 results for $\alpha = 10^\circ$	65

List of Figures

1.1	Structured and unstructured body-fit grids [3].	2
1.2	Cartesian grids with and without local refinement [3].	3
1.3	(a) Distribution of forcing F_k from X_k to its surrounding. (b) Various distributions functions that can be used from [8], [9] and [10], taken from [7].	5
1.4	Different types of cell nodes. F1, F2, F3 and F4 are the fluid nodes, G is a ghost cell node and P1, P2 are points on the solid boundary. [3]	7
1.5	Cut made at a solid boundary. [12]	8
1.6	Re-shaped cell near an embedded body with its respective 6-point stencil for determining the flux showed at the picture on the right [3].	8
1.7	Representation of the different type of cells. (Source [6])	9
1.8	Some of the available computational grids in the SOL repository. (Source [15])	10
2.1	Skewness factor. [14]	16
2.2	Possible stencils used in the WLS schemes [15].	19
2.3	Conservative cut around a solid body in a Cartesian grid [17].	21
2.4	Example of a cell's first and second neighbours by vertex on the right and by face on the left [19].	23
2.5	Grid generated without the grid correction steps on the left and a grid generated with the grid correction steps on the right. [14]	25
3.1	Expected streamlines for $Re = 10, 20$ and 40 [22].	28
3.2	Domain used to study the flow past a 2D cylinder [16].	28
3.3	Variation of the drag coefficient with the blockage effect B for $Re = 40$ [23].	29
3.4	Variation of the drag coefficient with the blockage effect for $Re = 40$ in a body-fit grid with the SOL code.	30
3.5	IBM grids. (a) Coarser discretization of the domain. (b) to (d) Close up of the three grids at study near the cylinder location.	32
3.6	Body-fit grids. (a) Coarser discretization of the domain. (b) to (d) Close up of the three gridss at study near the cylinder location.	34
3.7	Pressure distribution with velocity streamlines in the IBM(a) and in the Body-fit Method (b) for $Re = 10$	35

3.8 Pressure distribution with velocity streamlines in the IBM(a) and in the Body-fit Method (b) for $Re = 20$	36
3.9 Pressure distribution with velocity streamlines in the IBM(a) and in the Body-fit Method (b) for $Re = 40$	37
3.10 Analytical Singularity - analytical solution of the velocity magnitude.	39
3.11 Analytical Singularity - Velocity magnitude error field after four refinements.	40
3.12 Analytical Singularity - Decay of the velocity magnitude error with $1/\sqrt{N_c}$ for the uniform criterion.	41
3.13 Analytical Singularity - Velocity magnitude error field for the level 10 of the analytical cri- terion.	42
3.14 Analytical Singularity - Decay of the error with $1/\sqrt{N_c}$ for the analytical criterion.	42
3.15 Analytical Singularity - Velocity magnitude error field on the left and error estimation crite- rion values on the right for the level 8 of the Taylor criterion.	43
3.16 Analytical Singularity - Decay of the error with $1/\sqrt{N_c}$ for the taylor criterion.	44
3.17 Analytical Singularity - Velocity magnitude error field on the left and error estimation crite- rion values on the right for the level 7 of the residual criterion.	45
3.18 Analytical Singularity - Decay of the error with $1/\sqrt{N_c}$ for the Residual criterion.	45
3.19 Analytical Singularity: Final grids for the different criteria.	46
3.20 Analytical Singularity - Comparison between the different criteria of the error decay with $1/\sqrt{N_c}$	47
3.21 Analytical Singularity - Comparison between schemes for the analytical criterion.	47
4.1 Initial grid with a close up on the cylinder location on the right for the adaptive refinement simulation of the 2D cylinder case.	50
4.2 Pressure distribution for the level 7 of the Taylor criterion for $Re = 10$	51
4.3 Pressure distribution for the level 10 of the Residual criterion for $Re = 10$	52
4.4 Relative error between both adaptive criteria and the body-fit method for $Re = 10$	53
4.5 Pressure distribution for the level 7 of the Taylor criterion for $Re = 20$	54
4.6 Pressure distribution for the level 11 of the Residual criterion for $Re = 20$	55
4.7 Relative error between both adaptive criteria and the body-fit method for $Re = 20$	55
4.8 Pressure distribution for the level 10 of the Taylor criterion for $Re = 40$	56
4.9 Pressure distribution for the level 13 of the Residual criterion for $Re = 40$	57
4.10 Relative error between both adaptive criteria and the body-fit method for $Re = 40$	58
4.11 NACA 0012 airfoil shape.	59
4.12 Initial grid for NACA0012 airfoil.	60
4.13 Body-fit grid for NACA0012 airfoil.	60
4.14 Naca0012 grid resolution dependence for $\alpha = 0^\circ$ from [40].	61
4.15 Pressure distribution of the four grids at study for $\alpha = 0^\circ$	62
4.16 Distribution of the horizontal component of velocity for $\alpha = 0^\circ$	63

4.17 Distribution of the vertical component of velocity for $\alpha = 0^\circ$	64
4.18 Pressure distribution for $\alpha = 10^\circ$	65
4.19 Distribution of the horizontal component of velocity for $\alpha = 10^\circ$	66
4.20 Distribution of the vertical component of velocity for $\alpha = 10^\circ$	67
4.21 Fiva refinement steps of the Residual criterion.	68
4.22 Body-fit method convergence for the NACA 0012 airfoil.	69
4.23 C_d error relative to the body-fit method for $\alpha = 0^\circ$.	69
4.24 C_d error relative to the body-fit method for $\alpha = 10^\circ$.	70
4.25 C_l error relative to the body-fit method for $\alpha = 10^\circ$.	70

Nomenclature

A_{proj}	Projected area	α	Attack angle
B	Blockage factor	α_p	Pressure under relaxation factor
C_v	Volumetric heat capacity	α_u	Velocity under relaxation factor
C_{Dp}	Pressure drag coefficient	$\bar{\varepsilon}_u$	Mean error of the horizontal component of velocity
C_{Dv}	Viscous drag coefficient	$\bar{\varepsilon}_v$	Mean error of the vertical component of velocity
C_D	Drag coefficient	$\bar{\varepsilon}_{Umag}$	Mean error of the velocity magnitude
D	Diameter	\mathbf{F}_k	Nodal force
E_T	Taylor Criterion Error Estimation	\mathbf{S}	Surface vector of the cell
F_p	Pressure force	\mathbf{X}_k	Coordinate vector
F_v	Viscous force	\mathbf{d}	Distance vector
F_D	Drag force	\mathbf{u}_f	Velocity vector in the cell's face
Q	Volumetric heat source	δ	Dirac delta function
Re	Reynolds number	η	Blending factor
S_f	Cell's face length	λ	Heat conduction
S_ϕ	Transport Equation source term	$(M_{ij})_P$	Inertia Tensor of cell P
T	Fluid temperature	$\mathcal{F}(P)$	Set of all the cells' faces
U_f	Convective flux or Conservative velocity	μ	Dynamic viscosity
U_∞	Inlet velocity	ν	Kinematic viscosity
V_P	Cell Volume of cell P	ϕ	Transported variable
Γ_ϕ	Scalar diffusivity	ρ	Fluid density
Ω_b	Solid body domain	ε_u	Maximum error of the horizontal component of velocity
Ω_f	Fluid domain		

ε_v	Maximum error of the vertical component of velocity	k^{th}	Lagrangian point
$\varepsilon_{U_{mag}}$	Maximum error of the velocity magnitude	p	Pressure
a_l	Non-Diagonal term of momentum matrix	t	Time reference value
a_p	Diagonal term of momentum matrix	x	Horizontal cartesian coordinate
c	Chord	y	Vertical cartesian coordinate
g	Gravity acceleration vector	y_t	Half thickness

Glossary

CDS	Central Difference Scheme
CFD	Computational Fluid Dynamics
FLS	Face Least Squares
FVM	Finit-Volume Method
GPS	Ghost Point Scheme
GQS	Grid Quality Scheme
IBM	Immersed Boundary Method
ILV	Imposed Lateral Velocity
LS	Lateral Symmetry
MP	Material Points
NSE	Navier-Stokes Equations
Nc	Number of cells
PISO	Pressure Implicit with Splitting of Operator
PO	Pressure Outlet
SIMPLE	Semi-Implicit Method for Pressure Linked Equations
WLS	Weighted Least Squares

Chapter 1

Introduction

1.1 Motivation

The numerical analysis of mathematical problems has been a subject of study for thousands of years. More recently, names like Newton, Gauss, Euler and Lagrange presented various algorithms that used numerical methods in order to simplify the solution of several mathematical problems. Since there were no computers at that time, the computations were made by hand in a lengthy process. A turning point occurred when a few decades ago, computing power suffered an impressive development, allowing the introduction of new engineering fields (e.g. von Neumann in 1947). The numerical analysis of mathematical and physical problems could now be taken a step further and a new field named Computational Fluid Dynamics, commonly known as CFD, was created. CFD used algorithms to solve the governing equations of flows. With computers, it was possible to arrive to an approximate solution to these equations and with the continuous computer power increase, better solutions could be achieved. The current research in this field aims at improving the solutions accuracy and at reducing the time needed to perform the computational simulations.

A wide range of problems can now be solved using various algorithms. The potential this field has in the Aerospace Industry in order to diminish costs and development times is easily understood. The use of CFD in parallel with experimental work can also help to improve the aerodynamics of several components since there are numerous CFD tools that can be used to simulate the body-fluid interactions. Despite the number of tools available, the knowledge of the method used by the code is crucial in order to better understand how the governing equations are being solved and the inconveniences they have.

There has been an increasing demand for CFD techniques that can handle complex geometries, providing accurate results with low computational times. However, one of the core problems in CFD is the ability to handle complex geometries efficiently. Two different approaches can be used in order to simulate the interaction between a fluid and a body. One is using a body-fitted grid and the other one is using an Immersed Boundary Method, IBM.

In the Immersed Boundary Method, the grid is not conform with the immersed body. Instead, grid cells are divided into groups: solid, fluid and immersed cells. Immersed cells are the ones that have

both solid and fluid boundaries. This type of method allows for a simpler grid which will simplify the simulation. Body-fitted computations need a grid generator which will produce a grid of the fluid zone and of the immersed body. The generated grids can be structured or unstructured and allow for high-order accuracy results. The main disadvantage of this type of technique is the computational cost to generate a grid that conforms with a complex geometry. This disadvantage may be reduced if an automated grid generator coupled with error estimation criteria is implemented. The immersed-boundary method has some advantages in comparison. It is simple, more efficient and faster at solving the flow governing equations [1].

IBM techniques require a higher level of grid refinement in certain areas of the domain in order to achieve high accuracy results. The most important elements to refine are, typically, the cells in the body-fluid frontier since this boundary is approximated by Cartesian cells that do not follow the immersed boundary.

Consequently, it is obvious that the previous method could be combined with an adaptive grid algorithm. Its integration in the code allows for a selective grid refinement according to different parameters depending on the algorithm used. Increasing the number of cells only in specific areas assuring accurate results without the increase in computational time that would occur if all the domain was to be refined.

This Thesis' motivation was to combine the Immersed Boundary Method with the capabilities of an adaptive grid using the Residual Least-Squares Error Estimate and the Taylor Error Estimate in order to minimize this technique's disadvantages. After combining the IB method with an adaptive grid, the final objective of eliminating the grid generation by the user should be completed. This will also eliminate the need to have a non structured initial grid.

1.2 Historical context

Over the last decades, the Cartesian grid method has been gaining popularity. In this method, the governing equations are discretized in a Cartesian grid which does not adapt to the immersed boundaries. This allows for simpler and faster grid generation and maintains the simplicity of the governing equations in Cartesian coordinates [2]. In Figures 1.1 and 1.2 it is possible to compare several body-fit grids and Cartesian grids.

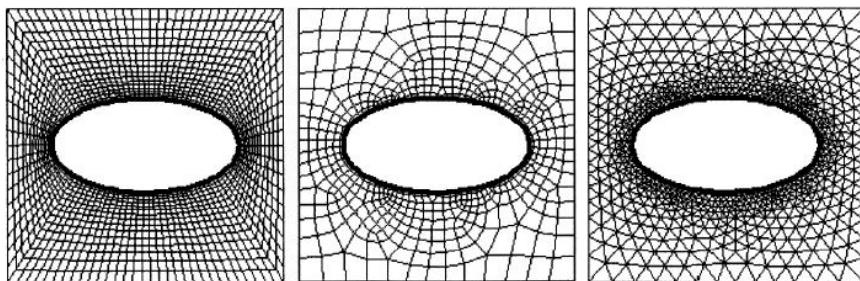


Figure 1.1: Structured and unstructured body-fit grids [3].

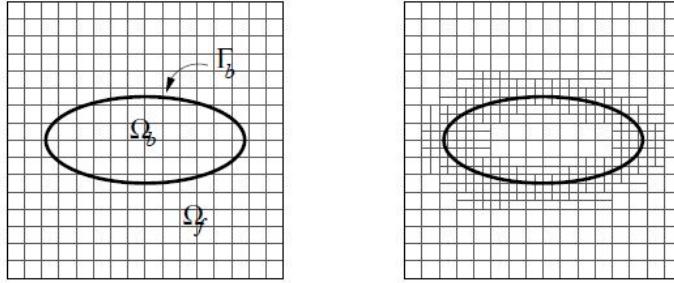


Figure 1.2: Cartesian grids with and without local refinement [3].

One example of a method that uses a Cartesian grid generator is the Immersed Boundary Method (IBM). The IBM was first introduced by Peskin [4] in order to study the blood flow in heart valves. This is a moving boundary problem where the blood flow simulation was performed on a Cartesian grid that did not follow the boundary geometry [5].

Because of this method's popularity, a number of variants have been introduced. Initially, the deformable heart valve boundary was characterized as a set of nodal forces which had their values incorporated in the fluid momentum equation as line forces. A smeared delta function was used to represent these forces resulting in a boundary dispersed over several cells [6].

From the information gathered, it becomes apparent that despite the high accuracy of body-fitted methods there has been a great effort to develop the IBM. Mainly, that comes from a need to reduce computational times since they play a huge role in the decision of how to discretize a domain. Having established the superiority of IBM in the grid generation process, there is a need to develop and improve this method's accuracy and also automate the solution process.

1.3 Immersed Boundary Method

The Immersed Boundary Method refers to the technique where the governing equations of the flow are discretized in a Cartesian grid that is not conform to the shape of the body immersed in the flow. Since the grid is not conform to the body, defining the boundary conditions cannot be made directly. Instead, a forcing function is included in the governing equations of the flow or the numerical scheme is altered close to the boundary.

At the time of this method's introduction, the formulation of the flow and moving boundaries were represented in different coordinates systems. The flow and its governing equations were solved in a Cartesian grid (Eulerian coordinate system) and the elastic boundaries were represented in a curvilinear grid (Lagrangian coordinate system). The force information is known on the Lagrangian coordinates and its effect is then spread to the Cartesian mesh with a Dirac delta function. This approach is known as continuous forcing approach.

Since the work of Charles S. Peskin [4], IBM techniques have been developed in different directions to deal with some of its weaknesses, such as defining the boundary conditions. The evolution of this method is well explained in [7, 3]. Thus, this explanation will follow the approach illustrated in the cited

sources.

What distinguishes one IB approach from another is the way each implements the boundary conditions. To better understand the different methods, we will now consider an incompressible flow with an embedded body. Equations 1.1 and 1.2 represent a simplified formulation of the governing equations and the boundary conditions.

$$\Lambda(U) = 0 \quad \text{in } \Omega_f \quad (1.1)$$

$$\text{with } U = U_\Gamma \quad \text{on } \Gamma_b, \quad (1.2)$$

where $U = (u, p)$ represents the unknowns. Equation 1.1 is discretized and solved on the Cartesian grid that covers the domain ($\Omega = \Omega_b + \Omega_f$).

For a body-fit method, the boundary condition 1.2 would be directly enforced but Immersed Boundary methods require an adaptation of equation 1.1 to allow the enforcement of the boundary condition. A new source term has to be added to the original governing equations so that the boundary conditions can be imposed. There are two different approaches to deal with this new source term. The continuous forcing and the discrete forcing approach.

1.3.1 Continuous forcing approach

The continuous forcing approach adds a new forcing term to equation 1.1 to simulate the immersed body presence in the fluid. Equation 1.3 can now enforce the boundary conditions.

$$\Lambda(U) = f_b \quad (1.3)$$

In order to solve the previous equation in the domain, it has to be discretized and equation 1.3 is transformed into 1.4.

$$[\Lambda]\{U\} = \{f_b\} \quad (1.4)$$

Equation 1.4 is now a system of discrete equations that can be solved on all the domain since the forcing term was already present in the set of equations previous to the discretization. This approach is called the continuous forcing approach and one of its key advantages is that it is independent from the spatial discretization scheme and separated solvers employed in the code.

Flows with Elastic Boundaries

This was the type of flow that Peskin (see [4]) aimed to solve when he first introduced the Immersed Boundary method. In this one, the Navier-Stokes equations are solved on a Cartesian grid, while the embedded body is represented by a set of elastic fibres. The location of these fibres can be traced in a Lagrangian coordinate system where the fibres move with local flow velocity. Equation 1.5 arises from the fact that the time derivative of the fibre location has to be equal to the fluid velocity at a certain point.

\mathbf{X}_k is the coordinate vector of the k^{th} Lagrangian point.

$$\frac{\partial \mathbf{X}_k}{\partial t} = \mathbf{u}(\mathbf{X}_k, t) \quad (1.5)$$

Using Hooke's Law, it is possible to make a relation between the deformation of these fibres and their stress force \mathbf{F} . The effect of the body has to be included, so a new local forcing term is included in the momentum equations. Equation 1.6 is responsible for distributing these forces to the Cartesian grid.

$$\mathbf{f}_m(\mathbf{x}, t) = \sum_k \mathbf{F}_k(t) \delta(|\mathbf{x} - \mathbf{X}_k|) \quad (1.6)$$

where δ is the Dirac delta function. As could be expected, the fibre location does not typically coincide with the Cartesian grid nodal points. Due to the disparity between the real and the nodal location, the forcing term has to be distributed in an area around each Lagrangian point (figure 1.3 (a)). It is then suitable to substitute the sharp dirac function by a smoother function to try and smooth the force distribution in the Cartesian grid.

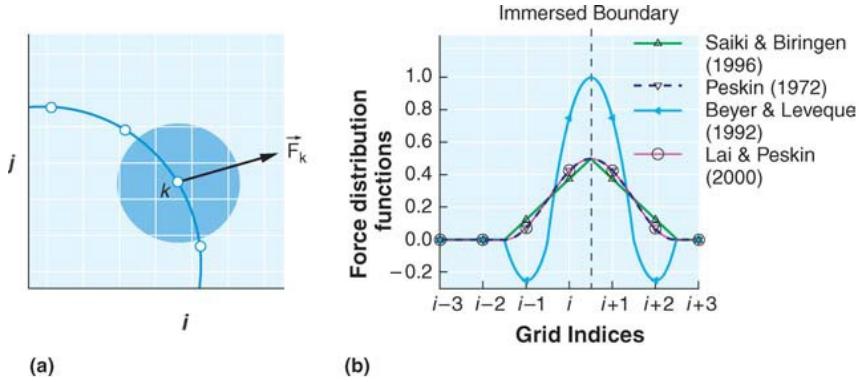


Figure 1.3: (a) Distribution of forcing \mathbf{F}_k from \mathbf{X}_k to its surrounding. (b) Various distribution functions that can be used from [8], [9] and [10], taken from [7].

Equation 1.7 is the discretized forcing function where d is the distribution function that can vary. In Figure 1.3 (b) several distribution functions are presented.

$$\mathbf{f}_m(\mathbf{x}_{i,j}, t) = \sum_k \mathbf{F}_k(t) d(|\mathbf{x}_{i,j} - \mathbf{X}_k|) \quad (1.7)$$

The d function is also used to calculate the fluid velocity in equation 1.5. Several papers [10, 9, 4, 8] have used different d functions to smooth the distribution of the stress through Cartesian nodes and the results can be summarized in figure 1.3.

Flows with Rigid Boundaries

The previous approach cannot handle rigid boundaries because small deformations of a very stiff boundary will cause large stresses. A possible solution to this problem is to assume the boundary of the body to be attached to an equilibrium position, held by a spring which will produce a restoring force.

Equation 1.8 represents the force the spring will produce

$$\mathbf{F}_k(t) = -\kappa(\mathbf{X}_k - \mathbf{X}_k^e(t)) \quad (1.8)$$

where κ is a positive spring constant and \mathbf{X}_k^e is the equilibrium position vector of the k^{th} Lagrangian point. In order to accurately portrait the rigid boundary, κ has to be very large to maintain the body in the equilibrium position, see [10, 9].

Despite the previous approach to try and solve the systems stiffness, its equations remain very stiff which will make it hard to maintain the stability of the system. This problem will limit the maximum possible time step when using explicit time integration methods which will increase the computation time. A similar and more general form of this approach can be seen in [11]. Unsteady flows and high Reynolds numbers represent a problem to the numerical stability of [10, 9, 11].

General Considerations

A continuous forcing approach is well suited to elastic boundaries since it is easy to implement. There have been successful examples of biological problems solved with this approach. A problem rises with rigid boundaries cases where the stiff systems of equations will only allow for acceptable results in low Reynolds flows with moderate unsteadiness.

The main problem of a continuous forcing approach is the way the force is distributed in a general area, not allowing for an accurate representation of the embedded body. This is more problematic at higher Reynolds numbers. One other disadvantage of this approach is the need to solve the flow inside the body.

1.3.2 Discrete forcing approach

The other approach that can be used to deal with the forcing term is the discrete forcing one. In this, equation 1.1 is discretized before including the forcing term and therefore, it takes no notice of the immersed body presence directly in the equations. Equation 1.9 is the set of discretized equations.

$$[\Lambda]\{U\} = 0 \quad (1.9)$$

Equation 1.9 is solved on Ω_f but in order to obtain the complete solution for the domain, a new equation 1.10 has to be solved on Ω_b .

$$[\Lambda]\{U\} = \{f'_b\} \quad (1.10)$$

This approach is called the discrete forcing approach and it has a strong dependence on the discretization scheme. This dependence allows for numerical accuracy control, stability and the conservation of the discrete properties.

1 - Indirect Boundary Conditions Imposition

In this case, the boundary condition is not applied directly in the numerical scheme. A new forcing term is added to the governing equations. This new term is different from the one in the Continuous forcing approach because it arises from the boundary conditions and not from the force vs displacement [3].

2 - Direct Boundary Conditions Imposition - DBCI

All approaches previously presented can produce problems at higher Reynolds numbers due to the force's distribution that causes a decrease of the solutions' local accuracy in the boundary where the force is smoothed. The solution to this problem is to alter the computational stencil to maintain a sharp boundary. The methods presented next in this Thesis respect this approach.

2.1 - Ghost-cell Finite-Difference (DBCI)

There is no need to calculate the flow inside of the body since it only costs computational time. In order to avoid this computational test, ghost-cells are created just inside the boundary in a way that it has at least one neighbour on the fluid zone. Since there is no information on these cells, the values of the variables are extrapolated from the values of the solid boundaries and the fluid cells around each ghost-cell.

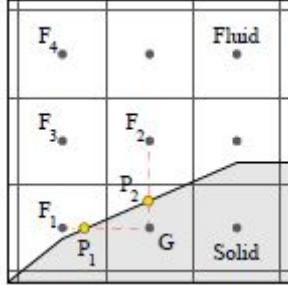


Figure 1.4: Different types of cell nodes. F_1 , F_2 , F_3 and F_4 are the fluid nodes, G is a ghost cell node and P_1 , P_2 are points on the solid boundary. [3]

An example of an extrapolation scheme is now presented to briefly explain how this method works. This linear extrapolation scheme is presented:

$$\phi = C_1xy + C_2x + C_3y + C_4 \quad (1.11)$$

where ϕ is a generic flow, variables C_1 , C_2 , C_3 , C_4 values will depend on an analysis of ϕ at the points F_1 , F_2 , P_1 and P_2 (see Figure 1.4). The type of extrapolation scheme to be used requires a previous analysis of the type of flow and its Reynolds number.

2.2 - Cut-cell Finite Volume (DBCI)

It is now time to take a finite-volume approach that can ensure both mass and momentum conservation

since none of the approaches above can ensure this conservation. To successfully implement a finite-volume approach, the frontier of the embedded body has to coincide with a cell face. For this to happen, the grid cells near the body have to be altered and part inside the frontier has to be cut. See figure 1.5 to easily understand how the cut is made.

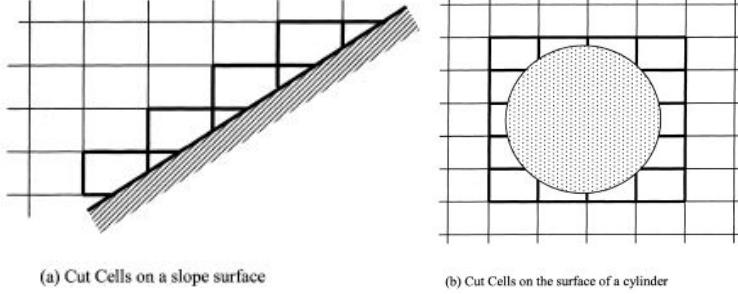


Figure 1.5: Cut made at a solid boundary. [12]

Mass, convective and diffusive flux integrals have to be estimated in this approach. This presents a problem in the reshaped cells [2].

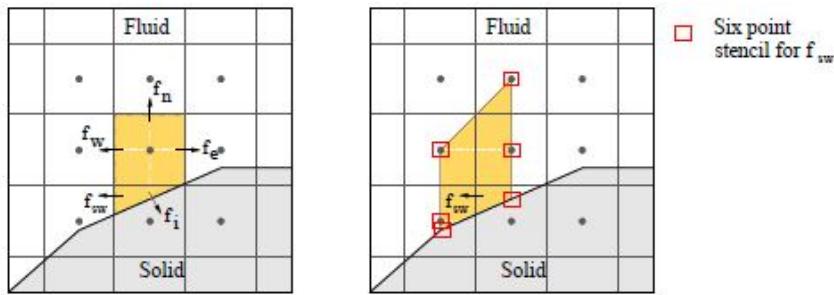


Figure 1.6: Re-shaped cell near an embedded body with its respective 6-point stencil for determining the flux showed at the picture on the right [3].

For instance, to calculate the flow on the southwest face f_{sw} from Figure 1.6 equation 1.12 can be used. This expression is based on a 6 point interpolation as shown in 1.6.

$$\phi = C_1xy^2 + C_2x^2 + C_3xy + C_4x + C_5y + C_6 \quad (1.12)$$

where C_1 to C_6 are coefficients whose values can be known evaluating ϕ at the 6 points shown in Figure 1.6.

The stencil and polynomial used in this approach retains a second order accuracy and it conserves the mass and momentum equations due to the use of a finite-volume approach. The cut-cell method is computationally advantageous because it discards the cells inside the body and it conserves the exact shape of the boundary instead of distributing the boundary force through close neighbour cells.

Two dimensional methods with this approach have been successfully used but extending the results to three dimensional problems represents a problem due to the complex geometry of the cells created with the boundary cut. Complex cell geometry makes for a complex discretization of the Navier-Stokes

equations. The 3D adaptation of this method is likely to be done with cell trimming techniques to generate a new grid that is conform to the body [13].

1.3.3 Applied Method

The method that is implemented and that will be used in all the results of this Thesis is based in the one utilized in reference [6] where in it, the immersed body is represented as a set of material points (MP) and flow calculations are done in the background grid.

Firstly, the vertices are classified as being in a fluid or solid zone according to their location. After classifying all vertices of the grid, the cells can be separated in different groups. If a cell has all its vertices in the fluid zone, the cell is classified as a fluid one. Similarly, if all cell's vertices are located in a solid zone, the cell is classified as a solid one. The IB cells make up the final group of cells which are characterized by having both solid and fluid vertices. Additionally, faces from these IB cells that are connected to a fluid cell are called IB faces. Examples of different types of cells and faces are represented in Figure 1.7.

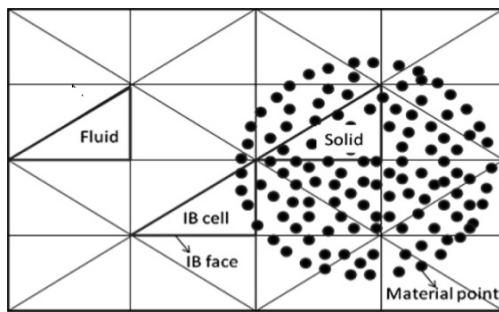


Figure 1.7: Representation of the different type of cells. (Source [6])

The value of the velocity used as boundary condition is calculated using an interpolation of the MP and neighbour cells' velocities. There are several approaches available to this interpolation, but the chosen approach was the least-squares method.

1.4 Error Estimators

The automatic mesh refinement has the potential to improve the accuracy of the numerical results, but for that to happen, the refinement algorithm requires some kind of indicator to determine where the refinement is needed. These sensors can be classified as error estimators. Error indicators are based on flow gradients of physical characteristics like vorticity or boundary layers and they do not provide information on the error levels, therefore they can not target directly the reduction in the discretization error. These error indicators also need the user to input problem related parameters and respective values. Error estimators, on the other hand, can provide a guess of the error distribution on the domain without the user having to input problem related parameters which would be dependent of the problem at study [14].

Since one of the main objectives of this work is to have a user independent algorithm that can automatically refine the domain, it has become obvious that error estimation is the key to achieve these objectives. Ideally, the use of error estimation techniques should allow for the creation of grids with an even error distribution. There are several error estimators that can be applied. This work will focus on the Taylor Series truncation error and the Residual Least Squares error estimator.

1.5 SOL code

The combination of IBM and adaptive refinement using error estimation criteria will be implemented in the SOL code. This code was created by the LASEF department and its main goal was to develop different approaches and methods of CFD in a common numerical code instead of having separate programs from each developer.

The objective of the code was to compute 3D flows on unstructured and adaptive grids. Several Thesis concerning the development of the SOL code were already concluded. It was written in C language and it has a common git repository that allows the transmission of code between the different developers. Figure 1.8 shows some of the grids available in the SOL repository.

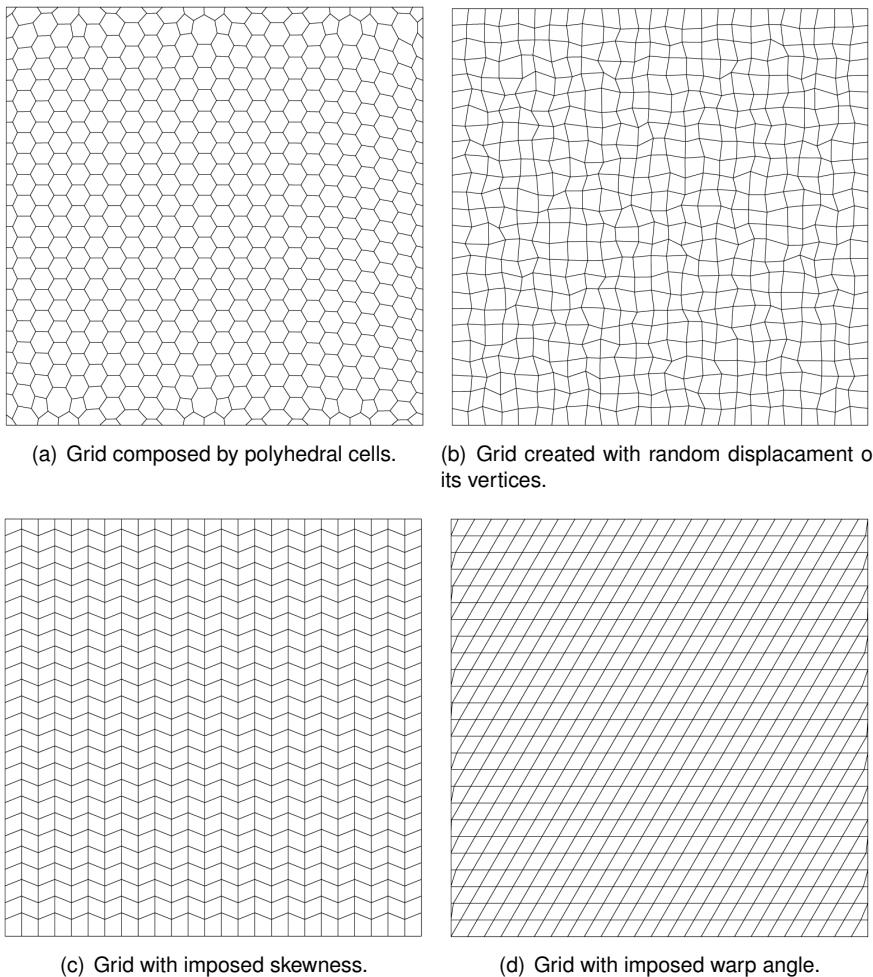


Figure 1.8: Some of the available computational grids in the SOL repository. (Source [15])

The code is being constantly improved and its capabilities developed by the several LASEF researchers.

1.6 Objectives

The present Thesis was guided by the following objectives:

- Validate the IBM;
- Implement the Taylor error estimation criterion on the IBM;
- Implement the Residual error estimation criterion on the IBM;
- Compare the results from both error estimation criteria.
- Validate the results from the different error estimation criteria;
- Demonstrate the advantages of the IBM and adaptive grid combination;
- Automatize the process of grid adaptation so that the user does not have to input an initial grid when a curve body is placed inside the fluid.

1.7 Thesis Outline

This Thesis is organized as follows:

- **Chapter 2** presents the equations of the fluid, the numerical schemes used in the Finite Volume discretization, the error estimation criteria, an overview of the IBM and the adaptive refinement algorithms;
- **Chapter 3** is dedicated to the verification and validation of all the methods and criteria used. Firstly, the IBM code is verified and only then, results using the adaptive refinement are presented and validated. After a literature survey, several case studies were selected and the results compared to ensure that the code behaved as expected;
- **Chapter 4** contains all the obtained results for the adaptive grids with the different error estimation criteria. This chapter is divided into two different cases. Firstly, the immersed cylinder case from chapter 4 is studied again after implementing the adaptive refinement, and then the flow past an immersed airfoil is studied for two different angles of attack;
- **Chapter 5** summarizes this Thesis with all the conclusions, achievements and suggestions for future research in this field.

Chapter 2

Numerical Methods

The numerical method present in this work was developed in the LASEF research laboratory of Instituto Superior Técnico. This work serves as a combination of [16] where the Immersed Boundary Method was implemented and of [14] where the error estimation criteria was implemented to body-fit grids.

2.1 Governing Equations

The Navier-Stokes equations are the four partial differential equations that are used to represent the behaviour of fluids, this set of equations is composed by Equation 2.1 which is the continuity equation and Equation 2.2 which represents the three momentum conservation equations.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = \rho \mathbf{g} - \nabla \left(p + \frac{2}{3} \mu \nabla \cdot \mathbf{u} \right) + \nabla \cdot [\mu \nabla \mathbf{u} + \mu \nabla^T \mathbf{u}] \quad (2.2)$$

In case of having to consider the compressibility effects, Equation 2.3 must also be solved to close the system of equations. This equation accounts for the conservation of energy.

$$\frac{\partial C_v T}{\partial t} + \nabla \cdot (\rho C_v T \mathbf{u}) = \rho C_v T \mathbf{u} - \nabla \cdot (p \mathbf{u}) - \nabla \cdot \left(\frac{2}{3} \mu (\nabla \cdot \mathbf{u}) \mathbf{u} \right) + \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla^T \mathbf{u})] + \nabla \cdot (\lambda \nabla T) + \rho Q \quad (2.3)$$

Compressibility properties won't be taken into account and the flow is considered to be isothermal and bi-dimensional for the purposes of this work. These simplifications result in Equations 2.4 and 2.5 which are the Navier-Stokes equations for incompressible flows.

$$\nabla \cdot \mathbf{u} = 0 \quad (2.4)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = \nabla \cdot (\nu \nabla \mathbf{u}) - \frac{1}{\rho} \nabla p \quad (2.5)$$

The two previous Equations can be written in a generalized transport equation represented by:

$$\frac{\partial \rho \phi}{\partial t} + \nabla \cdot (\rho \phi \mathbf{u}) = \nabla \cdot (\Gamma_\phi \nabla \phi) + S_\phi \quad (2.6)$$

The approach used by the SOL code to solve Equation 2.6 is the Finite-Volume method (FVM). To solve the Navier-Stokes equations with this method Equation 2.6 needs to be re-written in its integral form. Equation 2.7 represents this form after the volume integration and application of the Gauss or Divergence theorem to the convective and difusive terms.

$$\underbrace{\int_{\Omega} \frac{\partial(\rho\phi)}{\partial t} dV}_{\text{temporal term}} + \underbrace{\int_{\partial\Omega} \rho\phi \mathbf{u} \cdot d\mathbf{S}}_{\text{convective term}} = \underbrace{\int_{\partial\Omega} \rho \Gamma_\phi \nabla \phi \cdot d\mathbf{S}}_{\text{difusive term}} + \underbrace{\int_{\Omega} S_\phi dV}_{\text{source term}} \quad (2.7)$$

where $\partial\Omega$ is a closed boundary of the volume ω and $d\mathbf{S}$ represents a surface vector aligned with the normal of $\partial\Omega$.

In the case of Equation 2.2, applying the volume integration and using the Divergence-Theorem results in Equation 2.8

$$\underbrace{\int_{\Omega} \frac{\partial \mathbf{u}}{\partial t} dV}_{\text{temporal term}} + \underbrace{\int_{\partial\Omega} \mathbf{u} \mathbf{u} \cdot d\mathbf{S}}_{\text{convective term}} = \underbrace{\int_{\partial\Omega} \nu \nabla \mathbf{u} \cdot d\mathbf{S}}_{\text{difusive term}} + \underbrace{\int_{\Omega} -\frac{1}{\rho} \nabla p dV}_{\text{pressure term}} \quad (2.8)$$

Each term of Equation 2.8 is solved using different schemes. These schemes will be covered in section 2.2.

Since every case presented in this work is a steady-state flow, Equation 2.8 can be reduced to Equation 2.9 where the temporal term is null.

$$\underbrace{\int_{\partial\Omega} \mathbf{u} \mathbf{u} \cdot d\mathbf{S}}_{\text{convective term}} = \underbrace{\int_{\partial\Omega} \nu \nabla \mathbf{u} \cdot d\mathbf{S}}_{\text{difusive term}} + \underbrace{\int_{\Omega} -\frac{1}{\rho} \nabla p dV}_{\text{pressure term}} \quad (2.9)$$

2.2 Finite Volume Discretization

The numerical schemes used in this work have a second-order accuracy. The order of the schemes is a consequence of using the Gauss-Legendre quadrature to the terms of the Navier-stokes Equations in the integral form. To serve as an example, the source term of Equation 2.7 can be calculated using Equation 2.10.

$$\int_P S_\phi dV = S_\phi(\mathbf{P}) V_P \quad (2.10)$$

Equation 2.10 shows that the only source value required is the cell centroid value. This is how the the SOL code treats the source terms.

The surface integrals of Equation 2.9 can be calculated as a sum of the faces integrals since the cell surface ∂P is a set of its own faces. The Gauss-Legendre quadrature can then be applied using only one Gauss point at the face centroid f resulting in the convective term:

$$\int_{\partial P} \rho \phi \mathbf{u} \cdot d\mathbf{S} = \sum_{f \in \mathcal{F}(P)} \int_f \rho \phi \mathbf{u} \cdot d\mathbf{S} = \sum_{f \in \mathcal{F}(P)} \rho \phi_f \mathbf{u}_f \cdot \mathbf{S}_f = \sum_{f \in \mathcal{F}(P)} \rho \phi_f U_f \quad (2.11)$$

and resulting in the Diffusive term:

$$\int_{\partial P} \Gamma_\phi \nabla \phi \cdot d\mathbf{S} = \sum_{f \in \mathcal{F}(P)} \int_f \Gamma_\phi \nabla \phi \cdot d\mathbf{S} = \sum_{f \in \mathcal{F}(P)} (\Gamma_\phi)_f (\nabla \phi)_f \cdot \mathbf{S}_f \quad (2.12)$$

The set $\mathcal{F}(P)$ contains the faces of all the cells, \mathbf{S} is a vector of equal magnitude to the face area $A(f)$, normal to the face and pointing outwards from the cell. The conservative velocity U_f is the dot product between the velocity vector \mathbf{u}_f and the surface vector \mathbf{S}_f .

The diffusive scheme to solve Equation 2.12 consists in an algorithm that computes the gradient of the dependent variable $(\nabla \phi)_f$ from the surrounding cells.

2.2.1 Convective Scheme - Structured grid

The convective scheme is used to calculate the computational variable ϕ_f value at the face centroid from the neighbour cells. A first approach for the convective schemes can be defined by Equation 2.13.

$$\phi_f = \eta \phi_{P_0} + (1 - \eta) \phi_{P_1} \quad (2.13)$$

where η is the blending factor, ϕ_{P_0} and ϕ_{P_1} are the dependent variable values at P_0 and P_1 cells, respectively.

The linear interpolation was only used in the study-cases where the IBM with uniform criterion refinement was used. For the remaining adaptive criteria, the schemes used will be discussed in section 2.2.5. This interpolation is done by computing the intersection point \mathbf{z}_f between the surface plane of the cell face f with Equation 2.14.

$$\mathbf{z}_f = \mathbf{P}_0 + \eta_{LIN} (\mathbf{P}_1 - \mathbf{P}_0) = \mathbf{P}_0 + \eta_{LIN} \mathbf{d} \quad (2.14)$$

Vector \mathbf{d} is the distance between cells P_0 and P_1 centroids and the blending factor η_{LIN} is calculated using the projection of $(\mathbf{f} - \mathbf{P}_0)$ and \mathbf{d} vectors to the face normal which results in Equation 2.15.

$$\eta_{LIN} = \frac{\frac{(\mathbf{f} - \mathbf{P}_0) \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{S}_f}}{\frac{(\mathbf{P}_1 - \mathbf{P}_0) \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{S}_f}} = \frac{(\mathbf{f} - \mathbf{P}_0) \cdot \mathbf{S}_f}{(\mathbf{P}_1 - \mathbf{P}_0) \cdot \mathbf{S}_f} \quad (2.15)$$

The approach here presented does not use any information about the grid quality. The first grid quality parameter that can be accounted for is the skewness. This factor measures the deviation between

the distance vector d and the face centroid. A representation of this factor is shown in Figure 2.1 where $\sigma(f)$ represents the skewness.

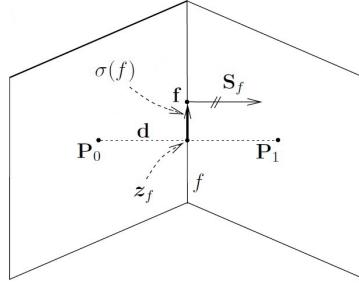


Figure 2.1: Skewness factor. [14]

Equation 2.14 can be altered to include a correction term that is directly proportional to the skewness, resulting in:

$$z_f = (1 - \eta)\phi_{P_0} + \eta\phi_{P_1} + \underbrace{(\mathbf{f} - \mathbf{P}_0 - \eta\mathbf{d}) \cdot (\nabla\phi)_f}_{\text{skewness correction}} \quad (2.16)$$

The scheme used did not include the correction factor because for the grids used, the skewness can be neglected.

2.2.2 Diffusive Scheme - Structured grid

The diffusive scheme consists in computing the value of the dependent variable gradient $\nabla\phi$ at the face centroid f using the information from surrounding cells.

Like the linear interpolation used in the convective scheme, the central difference scheme also has a second-order accuracy. This scheme was only used in the cases of the IBM with uniform refinement criteria.

The gradient $(\nabla\phi)_f$ is calculated with Equation 2.17 using two points.

$$(\nabla\phi)_f = (\phi_{P_1} - \phi_{P_0}) \frac{\mathbf{d}}{\mathbf{d} \cdot \mathbf{d}} \quad (2.17)$$

where d is the distance vector between the centroids of the cells adjacent to face f .

2.2.3 Cell Centred Gradient Scheme

To improve the accuracy of the schemes at non-structured or to compute the gradient terms of the governing equations, the variables gradients $\nabla\phi$ at the cell's centre have to be computed. The Gauss method was used in order to calculate these gradients.

The Gauss method consists of applying the Divergence theorem to the $\nabla\phi$ quantity and transforming it into a summation of the faces fluxes $\phi_f S_f$, as the following equation:

$$\int_P \nabla \phi dV = \int_{\partial P} \phi d\mathbf{S} \Rightarrow (\nabla \phi)_P = \frac{1}{V_P} \sum_{f \in \mathcal{F}(P)} \phi_f \mathbf{S}_f \quad (2.18)$$

Once again, both the second-order accuracy inside the cell and the Gauss-Legendre quadrature were considered. The computation of ϕ_f is made using the convective scheme presented in 2.2.1.

2.2.4 Weighted Least Squares Method

This method is very important for this Thesis since it is applied in the computation of the face WLS second-order schemes and in the error estimators that are explained in 2.6. WLS method is used to compute the polynomial fit of an arbitrary set of points with different values ϕ . To build this algorithm, the only requirement to be fulfilled is to have a number of polynomial to be computed, lower or equal to the number of cell values from the set of points chosen.

Considering a base polynomial with the form of a multidimensional Taylor series expansion of the variable ϕ centred at the cell P .

$$\phi(x, y, z; P) = \phi_P + \frac{\partial \phi}{\partial x}(x - x_P) + \frac{\partial \phi}{\partial y}(y - y_P) + \frac{\partial \phi}{\partial z}(z - z_P) + \dots \quad (2.19)$$

Equation 2.19 can be generalized into Equation 2.20.

$$Y = b_0 + \sum_{k=1}^n b_k X_k \quad (2.20)$$

where Y is a linear combination of n variable X_k plus a constant and b are polynomial coefficients that are going to be computed. Equation 2.21 represents a component y_i of the observation vector y .

$$y_i = (b_0 + b_1 X_{1,i} + b_2 X_{2,i} + \dots + b_n X_{n,i}) + e_i \quad (2.21)$$

Considering the set of points chosen, a new over-determined system of equations is formulated in Equation 2.22.

$$\mathbf{yW} = \mathbf{XbW} + \mathbf{eW} \quad (2.22)$$

where \mathbf{y} is a vector containing the m components of the observation of y , \mathbf{W} is a diagonal matrix formed by root of the weight function for each observation, \mathbf{X} contains the distances values for each observation, \mathbf{b} is the vector with the polynomial coefficients that are going to be computed and \mathbf{e} is the vector with the correlation error for each one of the observations.

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} & \dots & X_{n,1} \\ 1 & X_{1,2} & X_{2,2} & \dots & X_{n,2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1,m} & X_{2,m} & \dots & X_{n,m} \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad (2.23)$$

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_m \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} \sqrt{w_{1,1}} & \dots & 0 \\ 0 & \ddots & 0 \\ 0 & \dots & \sqrt{w_{m,m}} \end{bmatrix} \quad (2.24)$$

Each component of the vector e can be calculated using Equation 2.25 where the estimated value ($\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$) Vector b is obtained by the WLS method, where each component of the vector corresponds to the minimum of the sum in Equation 2.26.

$$e_i = y_i - \hat{y}_i \quad (2.25)$$

$$\min_{\mathbf{b}} \sum_{i=0}^m (w_{i,i})e_i^2 \Leftrightarrow \min_{\mathbf{b}} (\mathbf{W}^T \mathbf{e}^T)(\mathbf{W}\mathbf{e}) \quad (2.26)$$

The solution of this system can be obtained by Equation 2.27 when the residuals are neglected.

$$\mathbf{b} = \underbrace{(\mathbf{X}' \mathbf{W}^2 \mathbf{X})^{-1} \mathbf{X}' \mathbf{W} \mathbf{W} \mathbf{y}}_{(\mathbf{X}\mathbf{W})_{left}^{-1}} \quad (2.27)$$

where $(\mathbf{X}\mathbf{W})_{left}^{-1}$ is the left inverse of the rectangular matrix $\mathbf{X}\mathbf{W}$.

Applying the WLS method consists in using the Cramer's rule to compute the matrix $(\mathbf{X}\mathbf{W})_{left}^{-1}$ when the polynomial is linear and using the single value decomposition method in higher order polynomials.

2.2.5 Face WLS - Adaptive refinement criteria

The schemes used for the unstructured grids that resulted from the adaptive refinement were created using the WLS method. These schemes were built using linear polynomial centred in the face centroid with the computational values of the neighbour cells.

$$\phi(\mathbf{x}) = \phi_f + (\nabla\phi)_f \cdot (\mathbf{x} - \mathbf{f}) \quad (2.28)$$

where ϕ_f and $(\nabla\phi)_f$ are the convective and diffusive values, respectively, so both can be computed in the same regression. The cells included in this type of regression have at least one of the face vertices. Figure 2.2 shows three examples of cell's stencils that can be used with this scheme centred at three different faces. In this figure, each regression k is centred at the respective face S_k and the cells used

in each regression are marked with the respective regression number k .

1	1	2	2	
		S_2	2	
1	S_1	1	2	2
1	1	3	3	
		S_3	3	
		3	3	

Figure 2.2: Possible stencils used in the WLS schemes [15].

2.3 Coupling Velocity and Pressure

The continuity and momentum equations are coupled and there needs to exist an algorithm to handle the coupling of pressure and velocity. The choice of this algorithm is important because it can affect the accuracy and stability of the solution.

SIMPLE and PISO are the two different algorithms that are usually used in this situation. Despite PISO offering faster convergence speeds [17], the algorithm used in this work is the SIMPLE. The main advantage of this algorithm is being more robust which is important since the local refinement will cause a non-smooth grid locally which would cause the problem to diverge when using the PISO algorithm.

2.3.1 SIMPLE

SIMPLE stands for Semi-Implicit Method for Pressure Linked Equations and it is an iterative algorithm that allows to solve the incompressible Navier-Stokes equations.

The first step of the SIMPLE algorithm is to estimate the velocity field \mathbf{u}^* , which is estimated from the results of previous iterations and it has to satisfy the momentum equations. With the first step of the algorithm completed, the momentum equations for a steady state can be represented as Equation 2.29 in a linear system.

$$\frac{1}{\alpha_u} a_p \mathbf{u}_P^* + \sum_{l=1}^F a_l \mathbf{u}_l^* = -\frac{V_P}{\rho} (\nabla p^n)_P + \frac{1 - \alpha_u}{\alpha_u} a_p \mathbf{u}_P^* \quad (2.29)$$

where \mathbf{u}_P^0 is the velocity vector calculated in the previous time-step. It is essential to use an under-relaxation term to stabilize the numerical scheme. The value chosen for this term throughout this work is $\alpha_u = 0.8$. The next step is to calculate the face velocity U_f^* using a Rhie-Chow interpolation [15].

$$U_f^* = \mathbf{u}_f^* \cdot \mathbf{S}_f - \overline{\left[\frac{\alpha_u V_P}{\rho a_p} \right]_f} \left[(\nabla p^n)_f - (\overline{\nabla p^n})_f \right] \cdot \mathbf{S}_f \quad (2.30)$$

where \mathbf{u}_f^* is the interpolated velocity at the face centre and it is calculated using the convective scheme. The value for $(\nabla p^n)_f$ is calculated using the diffusive scheme to calculate the pressure field. The overlined terms are calculated by interpolating the values for the two adjacent cells of face f.

The previous calculated velocity-field satisfies the momentum equations but it has to be updated in order to also satisfy the continuity equation. To update the velocity field, the pressure correction field p' must be calculated using Equation 2.31.

$$\sum_{f=1}^F \overline{\left[\frac{\alpha_u V_P}{\rho a_p} \right]_f} (\nabla p')_f \cdot \mathbf{S}_f = \sum_{f=1}^F U_f^* \quad (2.31)$$

After solving Equation 2.31, it is possible to calculate the corrected face velocity, cell-centred velocity and pressure using Equations 2.32, 2.33 and 2.34 respectively.

$$U_f^{n+1} = U_f^* - \overline{\left[\frac{\alpha_u V_P}{\rho a_p} \right]_f} (\nabla p')_f \cdot \mathbf{S}_f \quad (2.32)$$

$$\mathbf{u}_P^{n+1} = \mathbf{u}_P^* - \frac{\alpha_u V_P}{\rho a_p} (\nabla p')_P \quad (2.33)$$

$$p^{n+1} = p^n + \alpha_p p' \quad (2.34)$$

Once again, there is a need to include an under-relaxed term to the pressure equation. The value used throughout this work is $\alpha_p = 0.2$

The convergence criteria used was based on [14] which stops the simulation once the maximum residual value is lower than 10^{-10} .

The SIMPLE algorithm can be summarized in a few steps.

1. Initial guess for the velocity and pressure field.
2. Compute the momentum equation using Equation 2.29.
3. Calculate velocities for the cell's faces using Equation 2.30.
4. Calculate the pressure correction term using Equation 2.31.
5. Calculate the corrected velocities at cells and faces and pressure fields using Equations 2.32, 2.33 and 2.34.
6. Return to step 2 using the values calculated in this iteration as the initial fields until the residual is lower than 10^{-10} .

2.4 IBM algorithm

The first step to implement the IB method is to divide the type of cell into three different groups according to the location of their vertices. Cells with all their vertices located in the fluid zone are called fluid cells, similarly, cells with all their vertices located in the solid zone are called solid cells, and finally, cells that have vertices located in both fluid and solid zones are called ib cells. An ib cell's face that is shared with a fluid cell is called an ib face.

The fluid cells are isolated from the remaining cells, creating an immersed boundary containing all ib faces. To this boundary is then applied a Dirichlet boundary condition for velocity and a Neumann boundary condition for the pressure correction.

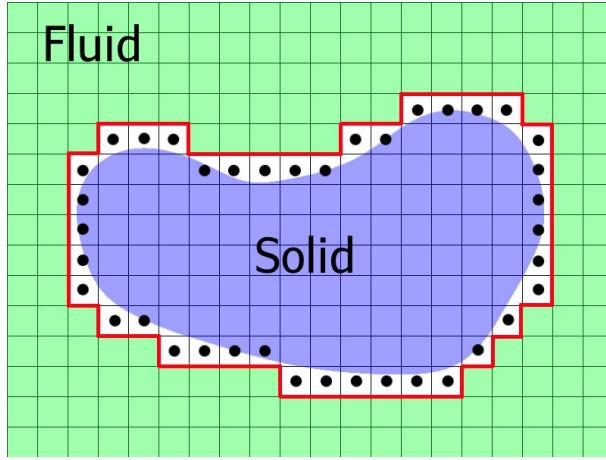


Figure 2.3: Conservative cut around a solid body in a Cartesian grid [17].

2.5 Calculating the Viscous and Pressure Forces

Accurately calculating the forces acting on an immersed body is crucial to compare different grids and methods. These forces are computed at the solid points that are associated with an ib face. Using the least square polynomials, it is possible to obtain the spatial derivatives $\partial u / \partial x$, $\partial u / \partial y$, $\partial v / \partial x$ and $\partial v / \partial y$ in the position of the solid point with the following equations:

$$\frac{\partial u}{\partial x} = a_1 + 2a_3x + a_5y + 2a_6xy \quad (2.35)$$

$$\frac{\partial u}{\partial y} = a_2 + 2a_4y + a_5x + a_6x^2 \quad (2.36)$$

$$\frac{\partial v}{\partial x} = b_1 + 2b_3x - 2b_3y - a_6y^2 \quad (2.37)$$

$$\frac{\partial v}{\partial y} = -a_1 - 2a_3x - a_5y - 2a_6xy \quad (2.38)$$

where a_i and b_i are the least-squares coefficients.

The viscous force can be calculated with:

$$d\mathbf{F}_v = \mu \frac{\partial U_T}{\partial n} S_S \mathbf{t} \quad (2.39)$$

where S_S is the area of the body surface associated with the solid point and where $\partial U_T / \partial n$ is the normal derivative of the tangent velocity, calculated by:

$$\frac{\partial U_T}{\partial n} = [\mathbf{t}]^T \cdot \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} \end{bmatrix} \cdot [\mathbf{n}] \quad (2.40)$$

where \mathbf{t} is the tangent versor and \mathbf{n} is the normal versor at the solid point position.

The total viscous force on the immersed body can finally be calculated by adding the contribution of the viscous force of each solid point.

The computation of the pressure force follows a similar approach to the viscous one. Using the least squares pressure polynomial, the pressure value at the solid point is obtained. The total pressure force can be calculated with:

$$d\mathbf{F}_p = (p_S S_S) \mathbf{n} \quad (2.41)$$

where p_S is the pressure value at the solid point location, computed with the least squares polynomial.

2.6 Error Estimation Criteria

The two different error estimation criteria that will be used in the course of this work are the Taylor and Residual ones. The purpose of these criteria is to estimate the error and the areas to be refined.

2.6.1 Taylor criterion

According to the Taylor series expansion theorem, a smooth function $\phi \in C^\infty$ in the vicinity of a point P can always be approximated by the sum of its derivatives at that point.

$$\phi(x_i; x_{i,P}) = \phi_P + \left(\frac{\partial \phi}{\partial x_i} \right)_P (x_i - x_{i,P}) + \frac{1}{2} \left(\frac{\partial^2 \phi}{\partial x_i \partial x_j} \right)_P (x_i - x_{i,P})(x_j - x_{j,P}) + \dots \quad (2.42)$$

The Finite Volume methods used in this thesis have a second-order accuracy, which means the discretization error is dominated by the second-order terms, so the average error equation can be computed with these terms.

$$E_T = \frac{1}{2V_P} \int_P \left(\frac{\partial^2 \phi}{\partial x_i \partial x_j} \right) (x_i - x_{i,P})(x_j - x_{j,P}) dV \quad (2.43)$$

where E_T is the Taylor series based error estimation and V_P is the cell P volume. Because of the second-order approximation, the Hessian matrix can be considered constant inside the cell P. Equation

2.43 integration results in:

$$E_T = \frac{1}{2V_P} \left| \left(\frac{\partial^2 \phi}{\partial x_i \partial x_j} \right)_P \right| (M_{ij})_P \quad (2.44)$$

where $(M_{ij})_P$ is the cell P inertia tensor.

The accuracy of the error estimation is dependent on the values of the Hessian matrix according to Equation 2.44. The components of this matrix were based on placing the values of the closest solid point in the face of the immersed boundary. This is a very simplistic approximation. To improve the accuracy of the criterion, the accuracy of the Hessian matrix must be improved. To do that, the values were changed to the values calculated by a second order interpolation introduced by [18] and derived from equations 2.35 to 2.38. The new values to be included in Hessian matrix can be calculated with:

$$\frac{\partial^2 u}{\partial x^2} = 2a_3 + 2a_6y \quad (2.45)$$

$$\frac{\partial^2 u}{\partial y^2} = 2a_4 \quad (2.46)$$

$$\frac{\partial^2 v}{\partial x^2} = 2b_3 \quad (2.47)$$

$$\frac{\partial^2 v}{\partial y^2} = -a_5 - 2a_6x \quad (2.48)$$

where, once again, a_i and b_i are the least-squares coefficients.

2.6.2 Residual criterion

A cubic order polynomial regression is computed with the information from the first and second neighbours by vertex. With this regression, new values are computed for the cell's face and gradients. These values computed with the regression will then be compared to the values obtained with the diffusive and convective schemes. Computing again the residual values using Equation 2.49 allows for this comparison.

		2		
	2	1	2	
2	1	*	1	2
	2	1	2	
		2		

2	2	2	2	2
2	1	1	1	2
2	1	*	1	2
2	1	1	1	2
2	2	2	2	2

Figure 2.4: Example of a cell's first and second neighbours by vertex on the right and by face on the left [19].

$$E_R = \frac{\sum_{f=1}^F U_f^n u_f - \nu_f \sum_{f=1}^F (\nabla u)_f \cdot S_f + (V_p/\rho) \nabla p^n}{a_p} \quad (2.49)$$

where u_f and $(\nabla u)_f$ values are computed with the cubic regression and a_p is the value of the matrix used for the momentum equations.

2.7 Adaptive meshing algorithm

Equation 2.50 is valid for a second order method in space for the relation between the local error E and the hydraulic diameter h . Equation 2.51 is also valid where h_i is the ideal hydraulic diameter and E_0 is the desired local error.

$$|E| \sim Kh^2 \quad (2.50)$$

$$|E_0| \sim Kh_i^2 \quad (2.51)$$

manipulating the previous equations and noting that a refinement step divides the hydraulic diameter in half, it is possible to achieve a relation between the error on two different refinement levels in:

$$E_0/E \sim (h_i/h)^2 = (1/2)^2 = 0.25 \quad (2.52)$$

This value can now be used to try and create adaptive grids with a constant error distribution. To do that, after running the simulation for each refinement step, the error is estimated in all cells using the chosen error estimation criterion from section 2.6. The maximum value for the error estimation is then chosen and multiplied by the error reduction factor of 0.25 calculated in 2.52. The result of this product is the threshold for whether or not the cells are refined. All cells with an error estimation higher than the threshold are selected and will be refined.

When applied to coarse grids, this type of refinement can result in a stiff cell distribution which can then cause numerical instabilities and divergence. This problem becomes more evident with the increasing number of refinement levels. To improve the general quality of the mesh and to try and avoid these problems, some measures are taken:

- The results for the criterion are smoothed in the domain;
- The selection of cells to be refined is expanded to their neighbour cells;
- If in a group of cells chosen to be refined, there are cells with different refinement levels, only the ones with the lowest level will be refined, preventing the accumulation of grid interface in the same location;
- The refinement directions are cross-checked. This causes incompatible directions of refinement to be excluded;
- As the anisotropic cells stretch, it is harder to refine them in counter direction. This prevents the degree of non-orthogonality to become severe.

This algorithm plays a vital role for this project because achieving an accurate solution is dependent on a grid that does not cause numerical instabilities or that causes the solution to diverge. Figure 2.5 shows the difference between a grid generated with and without the correction steps previously detailed in this section.

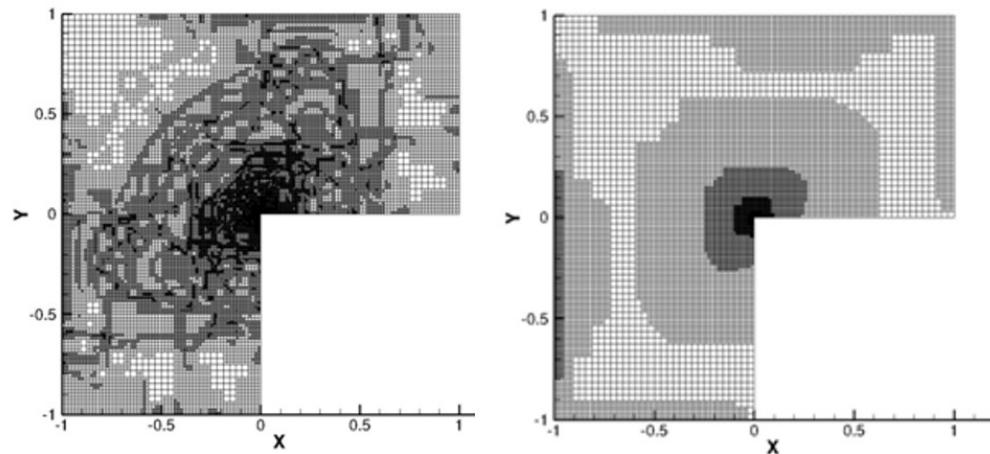


Figure 2.5: Grid generated without the grid correction steps on the left and a grid generated with the grid correction steps on the right. [14]

Chapter 3

Verification and Validation

The code used to perform the simulations of a flow past an immersed body is compared with other solutions from other work, and its verification and validity is tested using an array of cases. Validating a method consists in comparing the numerical results with experimental data and verifying a method consists in comparing the numerical results to an analytical solution.

3.1 Flow Around a 2D Cylinder

Flow past a circular cylinder is one of the most studied problems in fluid mechanics due to its geometric characteristics and practical importance in engineering [20]. These reasons made this a benchmark problem with a vast amount of data from both experimental and numerical data in the literature. The flow characteristics and drag coefficient on the cylinder are dependent on the Reynolds number. The Reynolds number is the ratio between inertial and viscous forces and can be calculated for this case study with:

$$Re = \frac{U_\infty D}{\nu} \quad (3.1)$$

where U_∞ is the velocity at the inlet boundary, D the diameter of the cylinder and ν is the kinematic viscosity of the fluid. The drag coefficient can be calculated with,

$$C_D = \frac{2F_D}{\rho U_\infty^2 A_{proj}} \quad (3.2)$$

where F_D is the drag force, ρ is the fluid density and A_{proj} is the projected area of the cylinder perpendicular to the flow ($A_{proj} = D$).

At low Reynolds numbers ($Re < 46$) the flow is stationary and symmetrical in respect to the cylinder centreline in the wake, where two vortices are generated. The size of these eddies increase with the Reynolds number. For $Re > 46$ there is an amplification of small instabilities upstream the cylinder and unsteadiness arises spontaneously even though the imposed conditions remain steady. Vortex shedding occurs behind the cylinder in a phenomenon known as the Kármán vortex street. If the Reynolds number increases even further, the wake of the flow becomes 3D and can no longer be studied as a 2D problem

[21].

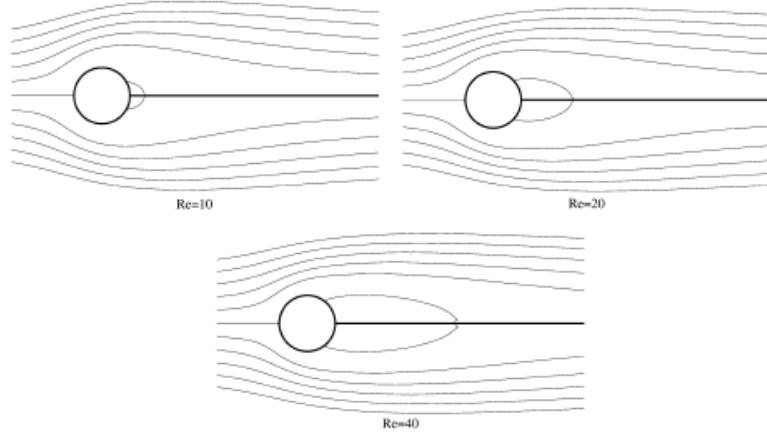


Figure 3.1: Expected streamlines for $Re = 10, 20$ and 40 [22].

For this section, three different Reynolds numbers were chosen to compare with the literature available ($Re = 10$, $Re = 20$ and $Re = 40$). The expected results are symmetrical, as previously described, and the separation zone is represented in Figure 3.1.

3.1.1 Computational Domain Confinement Issue - Blockage Effect

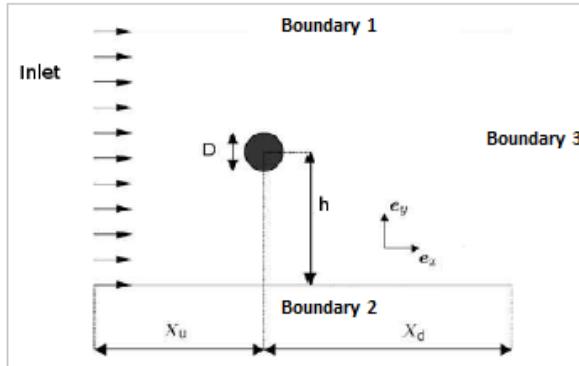


Figure 3.2: Domain used to study the flow past a 2D cylinder [16].

After analysing the range of results in the literature for the case of an immersed cylinder, it was obvious they were inconsistent. One of the reasons why results differed so much from one another may be because of different domain sizes that caused different values of the blockage effect, impacting the final results.

The blockage effect is created by immersing a cylinder in a flow can alter significantly the results obtained due to the flow confinement. The blockage factor is the ratio between the diameter (D) and half the height (h) of the domain ($B = D/h$) and its effect is considered to be insignificant when $B \leq 0.01$ [23].

In order to define a size to the domain that did not impact the results, a need to study this factor arose. It was decided to use a body-fitted method already available and study the results for different

domain sizes. A Reynolds number equal to 40 was chosen and a cylinder of diameter (D) 0.02 was placed inside a steady flow represented in Figure 3.2.

Since this problem was already studied by Sen, Mittal and Biswas in [23], a comparison can be made with the cited work. From Figure 3.3 it can be confirmed that for a blockage effect value around 0.01, there is little to no changes in the final value of the drag coefficient.

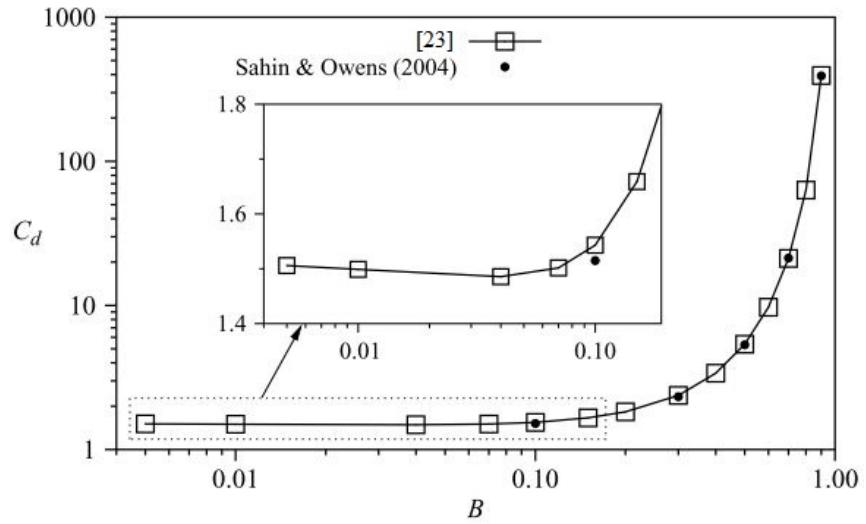


Figure 3.3: Variation of the drag coefficient with the blockage effect B for $Re = 40$ [23].

To confirm the effects of the blockage effect, the same study was performed using the in-house code. It was decided to use a body-fit grid because this code was already tested and this effect should not vary between different methods. After several simulations for four different domain sizes and respective blockage values, the results can be presented in Figure 3.4 and Table 3.1. The percentage error in Table 3.1 is calculated comparing the results from each grid size to the grid with the biggest domain.

To perform this study, the most refined grid from section 3.1.4 was used for $B=0.01$. Cutting the grid or adding new cells was required in order to generate grids with the same characteristics but different values for B .

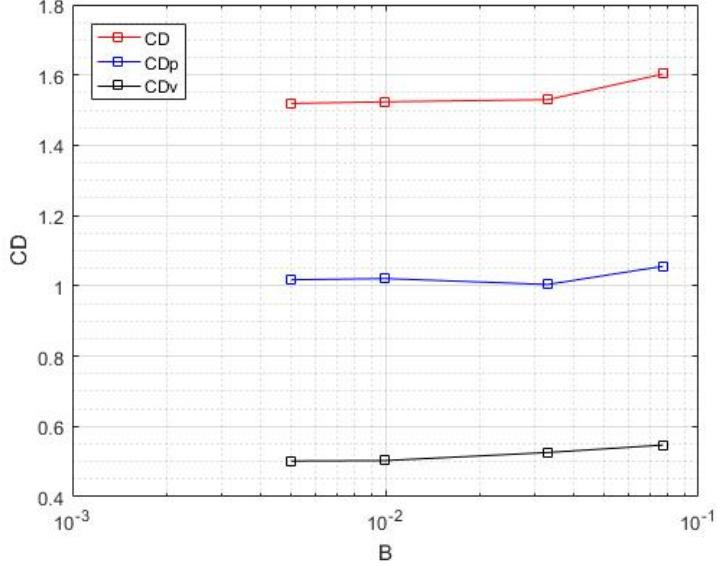


Figure 3.4: Variation of the drag coefficient with the blockage effect for $Re = 40$ in a body-fit grid with the SOL code.

Table 3.1: Drag coefficient with the respective viscous and pressure components variation with the blockage effect

	Blockage			
	0.077	0.033	0.01	0.005
C_D	1.6028	1.5297	1.5237	1.5191
C_{Dp}	1.0558	1.0041	1.0208	1.0174
C_{Dv}	0.546971	0.525806	0.5029	0.5017
Error (%)	5.512488	0.7098	0.3058	-

It is now obvious from the previous results that the conclusion from [23] is also valid for this work and because of that, the domain size to be studied from this point forward, shall respect the condition $B = 0.01$ since lower values do not significantly improve the results.

3.1.2 Geometry and Domain Characteristics

Quantifying the effect of the blockage and having defined the domain size, the whole geometry of the problem can be defined. This problem dimensions are shown in Table 3.2.

Table 3.2: 2D Cylinder Domain Dimensions.

D [m]	0.02	
X_d [m]	1.00	$X_d(D)$ 50D
X_u [m]	1.00	$X_u(D)$ 50D
h [m]	1.00	$h(D)$ 50D

It is now time to define the flow velocity and kinematic viscosity since these values will alter with the Reynolds numbers to be used in this work. Since the inlet velocity value was prescribed with a unitary value, the only value from Equation 3.1 that was altered from case to case was the kinematic viscosity.

The Reynolds number can now be easily calculated according to the following equation:

$$Re = \frac{0.02}{\nu} \quad (3.3)$$

and the values of kinematic viscosity used in each case are presented in Table 3.3.

Table 3.3: Reynolds and kinematic viscosity values

Re	ν
10	0.002
20	0.001
40	0.0005

The drag coefficient expression can also be easily simplified because $\rho = 1$, $U_\infty^2 = 1$, and $A_{proj} = DL$ where D is the diameter and L the height of the cylinder. Substituting values in Equation 3.2, a new equation is obtained:

$$C_D = \frac{F_D}{0.01} \quad (3.4)$$

3.1.3 Immersed Boundary Method Grids and Convergence Study

The different grids presented in Figure 3.5 were studied and several boundary conditions were tested in order to calculate the drag coefficient of an immersed cylinder in a flow since there is no consensus in the literature about what lateral boundaries should be used in this problem.

It is important to first clarify the designation of each boundary condition used: Symmetry plane, Imposed lateral velocity and Pressure Outlet. The symmetry plane imposes the Neumann boundary condition with zero gradient for all the variables except vectorial variables which will be assigned a zero value Dirichlet boundary condition ($\vec{u} \cdot \vec{n} = 0$). Pressure outlet imposes a Dirichlet boundary condition for the pressure and a Neumann null gradient condition for the velocity field. The imposed lateral velocity is simply imposing the wall with an unitary velocity value [16]. A pressure outlet boundary condition was chosen as Boundary 3 from Figure 3.2 for all three combinations of boundary conditions studied. The difference between each combination is the lateral boundary conditions. In one case an imposed lateral velocity was used, for the other a symmetry boundary and for the last one, a pressure outlet condition was chosen to study their influence in the numerical results.

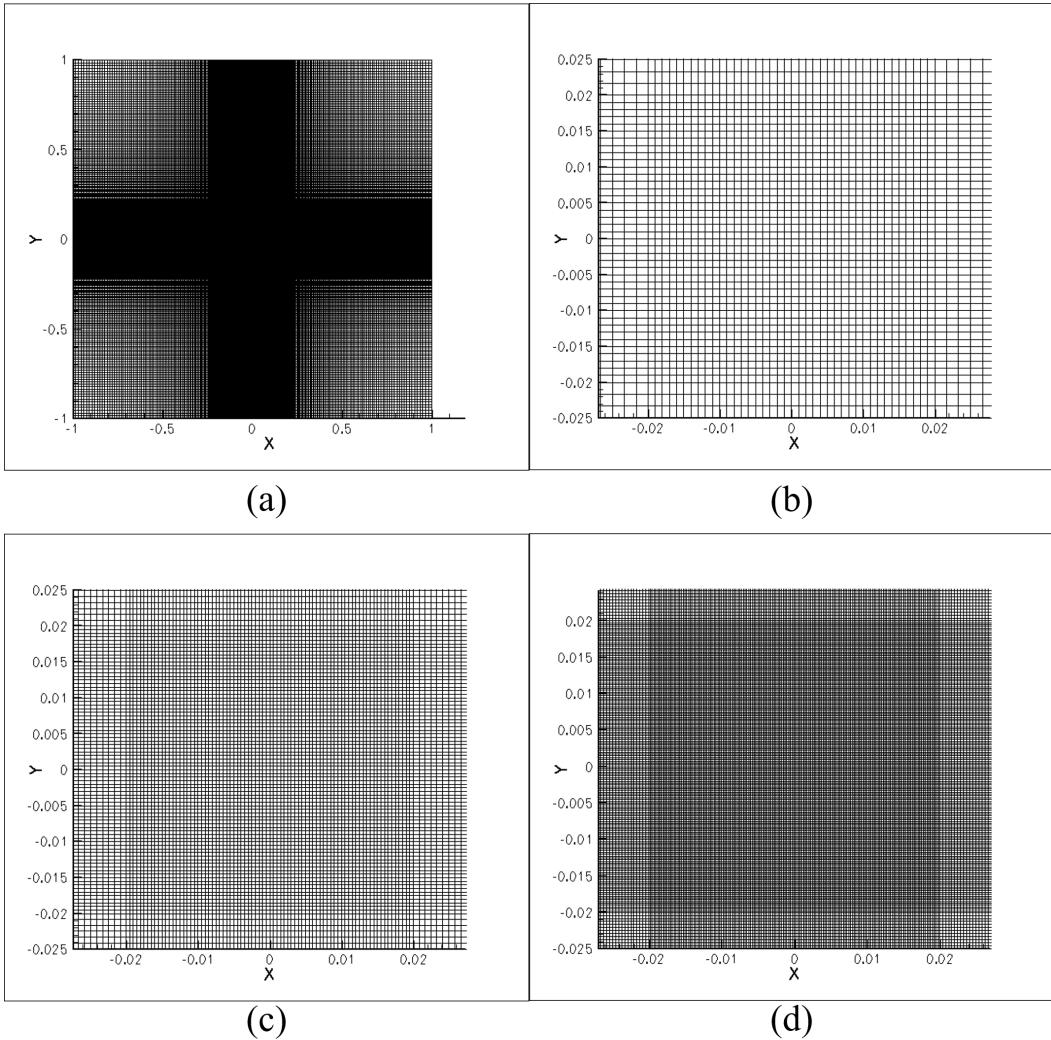


Figure 3.5: IBM grids. (a) Coarser discretization of the domain. (b) to (d) Close up of the three grids at study near the cylinder location.

It is now time to perform the simulations having in mind all the conclusions gathered above. In order to conclude the grid size needed for the IBM to converge, three different grids where studied.

Table 3.4: IBM Mesh refinement

	1	2	3
Number of cells	121104	484416	1937664
Diameter Volume Divisions ($D/\Delta x$)	20	40	80
Δx	0.001	0.0005	0.00025

With the grids of Table 3.4, it is now possible to present the results for each one in Table 3.5. The effect of the lateral boundary conditions was also studied. The C_D Error was calculated comparing the two most refined grids of each case.

Table 3.5: IBM convergence study

		$Re = 10$			$Re = 20$			$Re = 40$		
		LS	ILV	PO	LS	ILV	PO	LS	ILV	PO
1	C_D	2.652	2.653	2.636	1.912	1.912	1.903	1.420	1.420	1.415
	C_{Dv}	1.146	1.147	1.140	0.743	0.743	0.740	0.473	0.473	0.471
	C_{Dp}	1.505	1.506	1.496	1.168	1.169	1.163	0.947	0.947	0.944
2	C_D	2.731	2.732	2.715	1.973	1.974	1.964	1.471	1.471	1.465
	C_{Dv}	1.191	1.192	1.185	0.777	0.777	0.774	0.499	0.499	0.497
	C_{Dp}	1.540	1.540	1.530	1.196	1.197	1.191	0.972	0.972	0.968
3	C_D	2.731	2.732	2.749	2.000	2.000	1.991	1.493	1.493	1.487
	C_{Dv}	1.191	1.192	1.203	0.791	0.791	0.788	0.510	0.510	0.509
	C_{Dp}	1.540	1.540	1.545	1.208	1.208	1.202	0.983	0.983	0.979
C_D Error		0%	0%	1.25%	1.37%	1.32%	1.37%	1.5%	1.5%	1.5%

From table 3.5 it is possible to see that the drag coefficient values did not converge. This study would require a grid with even more cells. This more refined grid would require a lot more computational power and computational time. Due to the great increase of resources needed it was decided not to further refine the IBM grid. It is also possible to see that the drag coefficient values are slightly lower for the Pressure Outlet Boundary conditions. This tends to happen because this type of conditions decrease even more the confinement effect since it allows for fluid to escape through the boundary.

The failure of the studied grids to converge, despite the fact that they are non-structured grids with a high number of cells, seems to indicate a need to use adaptive refinement in this case.

3.1.4 Body-fit Grids and Convergence Study

To have a comparison term, the same study was made for body-fit grids. From the previous study, it was decided to only use Pressure Outlet boundary conditions since these would further decrease the blockage effect and because they would require less computational time. In Figure 3.6 the whole grid and solid boundary close-ups are displayed.

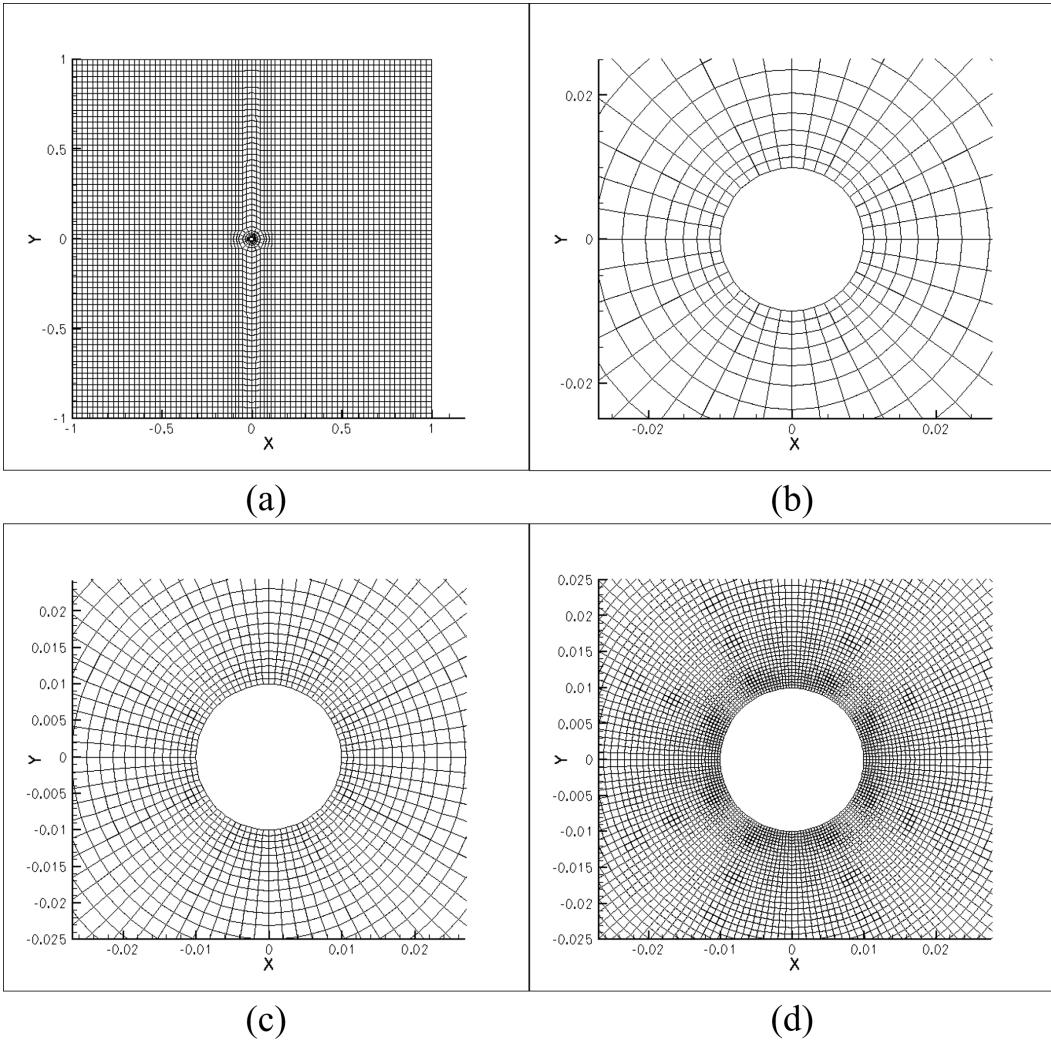


Figure 3.6: Body-fit grids. (a) Coarser discretization of the domain. (b) to (d) Close up of the three gridss at study near the cylinder location.

Table 3.6 provides more detailed information about each grid and an approximated number of Diameter Volume divisions to make the comparison to the IBM grids. The values from Table 3.6 are not exactly equivalent to the ones from Table 3.4 because the body-fit method follows the exact shape of the immersed body.

Table 3.6: Body-fit Mesh refinement

	1	2	3
Number of cells	28580	113600	454400
Diameter Volume Divisions ($D/\Delta x$)	13	26	51
Δx	0.00157	0.00079	0.00052

The results for each simulation are presented in Table 3.7 and it is possible to conclude that the Body-fit method requires less cells for the convergence of the drag coefficient. This advantage is a consequence of the better definition of the boundary.

Table 3.7: C_D results for the different body-fit grids.

		$Re = 10$	$Re = 20$	$Re = 40$
1	C_D	2.756	2.009	1.524
	C_{Dv}	1.197	0.783	0.503
	C_{Dp}	1.559	1.226	1.021
2	C_D	2.770	2.009	1.507
	C_{Dv}	1.211	0.794	0.513
	C_{Dp}	1.559	1.215	0.993
3	C_D	2.775	2.011	1.505
	C_{Dv}	1.216	0.798	0.517
	C_{Dp}	1.559	1.212	0.988
C_D Error		0.18%	0.1%	0.1%

3.1.5 Reynolds 10

Having performed several numerical analysis, it is now possible to focus on one specific Reynolds number at a time. In this section, a comparison between the achieved results and literature results will be made.

Firstly, Figure 3.7 presents the flow past the immersed body. In this Figure it is easily stated that the size of the eddies behind the solid body are very small.

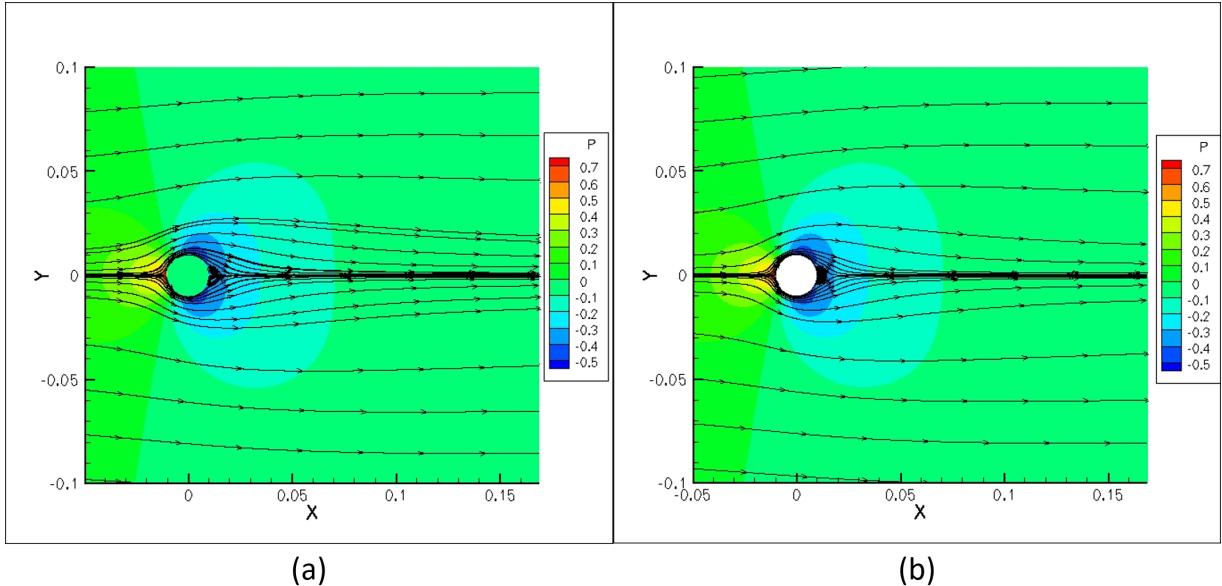


Figure 3.7: Pressure distribution with velocity streamlines in the IBM(a) and in the Body-fit Method (b) for $Re = 10$.

Table 3.8 summarizes the results from different authors. L_w/D is the quotient between the wake's length and the diameter of the immersed body. None of these works studied the effect of the blockage effect, and because of that, it is important to understand what were the domain characteristics for each work in order to have a better sense of why some values differ from one to another. A more detailed table is available in the appendix where more details can be found.

Table 3.8: Results from literature survey and from the present work for $Re = 10$.

Reference	year	L_w/D	C_D	B
Lima E Silva et al. [24]	2003	0.26	2.81	-
He and Doolen [25]	1997	0.24	3.17	0.018
Sen et al. [23]	2009	-	2.795	-
Ding et al. [22]	2004	0.252	3.07	0.03125
Park et al. [20]	1998	-	2.78	0.02
Present - Bodyfit	2016	-	2.78	0.01
Present work - STAR-CCM+	2016	0.23	2.82	0.025
Present work - IBM	2016	0.25	2.75	0.01

As expected, values from the present work fall in the range of results calculated by other authors.

3.1.6 Reynolds 20

Having performed a comparative analysis for $Re = 10$, it is now time to do the same for $Re = 20$. Similarly to the previous section, the first step is to do a visual analysis of the results from both IB and Body-fit methods.

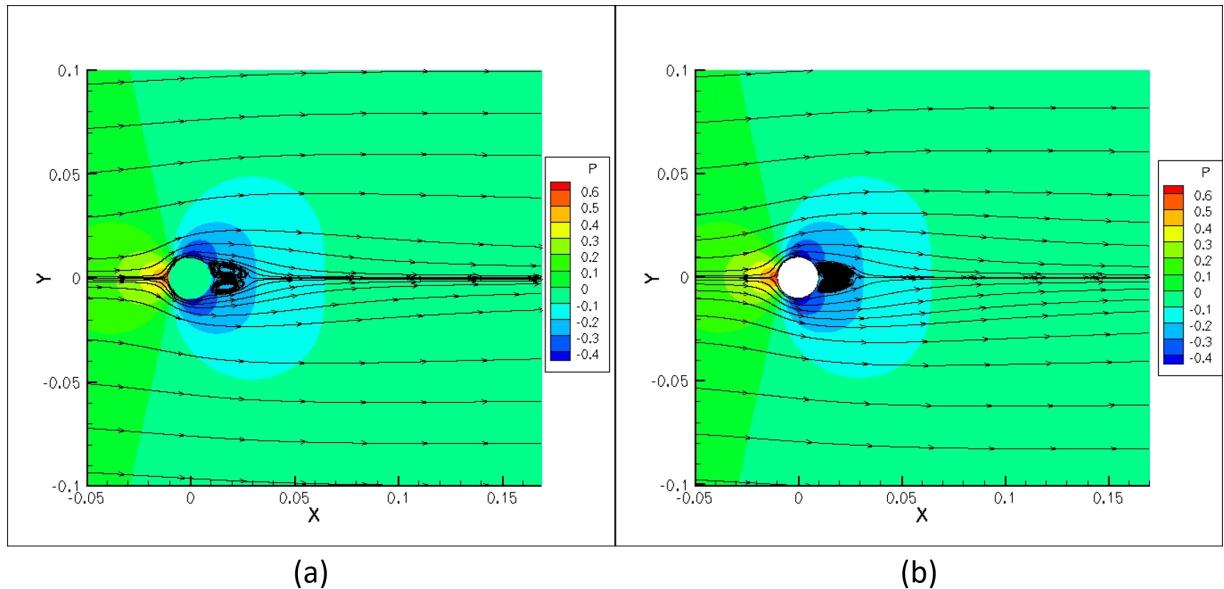


Figure 3.8: Pressure distribution with velocity streamlines in the IBM(a) and in the Body-fit Method (b) for $Re = 20$.

As expected, in Figure 3.8 it is obvious the growth in the eddies size. This behaviour falls in line with what was previously stated about the relation between Reynolds number and the size of the eddies.

Table 3.9 summarizes all the results obtained from the literature survey as well as with the information about the blockage effect and L_w/D .

Table 3.9: Results from literature survey and from the present work for $Re = 20$

Reference	year	L_w/D	C_D	B
Lima E Silva et al. [24]	2003	1.04	2.04	-
Ye et al. [2]	1999	0.92	2.03	-
Linnick and Fasel [26]	2005	0.93	2.16	0.056
Linnick and Fasel [26]	2005	0.93	2.06	0.023
Fornberg [27]	1980	0.91	2	-
Calhoun [28]	2002	0.91	2.19	-
Niu et al. [29]	2006	0.95	2.144	0.025
He and Doolen [25]	1997	0.92	2.152	0.018
Wang et al. [5]	2009	0.98	2.25	0.067
Xu and Wang [30]	2006	0.92	2.23	0.0625
Choi et al [31]	2007	0.9	2.02	0.0125
Pacheco et al. [32]	2005	0.91	2.08	0.033
Frisani and Hassan [21]	2015	0.9	2.22	0.05
Frisani and Hassan [21]	2015	0.93	2.229	0.05
Frisani and Hassan [21]	2015	0.85	2.168	0.05
Frisani and Hassan [21]	2015	0.92	2.124	0.05
Frisani and Hassan [21]	2015	0.92	2.167	0.05
Sen et al. [23]	2009	-	2.019	-
Ding et al.[22]	2004	0.93	2.18	0.03125
Park et al. [20]	1998	-	2.01	0.02
Present work- Bodyfit	2016	0.95	2.01	0.01
Present work - STAR-CCM+	2016	-	2.04	0.025
Present WORK - IBM	2016	0.98	1.99	0.01

Once again, the results obtained in the course of this work are similar to the values obtained by other work when a similar B is being used.

3.1.7 Reynolds 40

Finally, $Re = 40$ also requires the type of analyses already performed for the other Reynolds numbers.

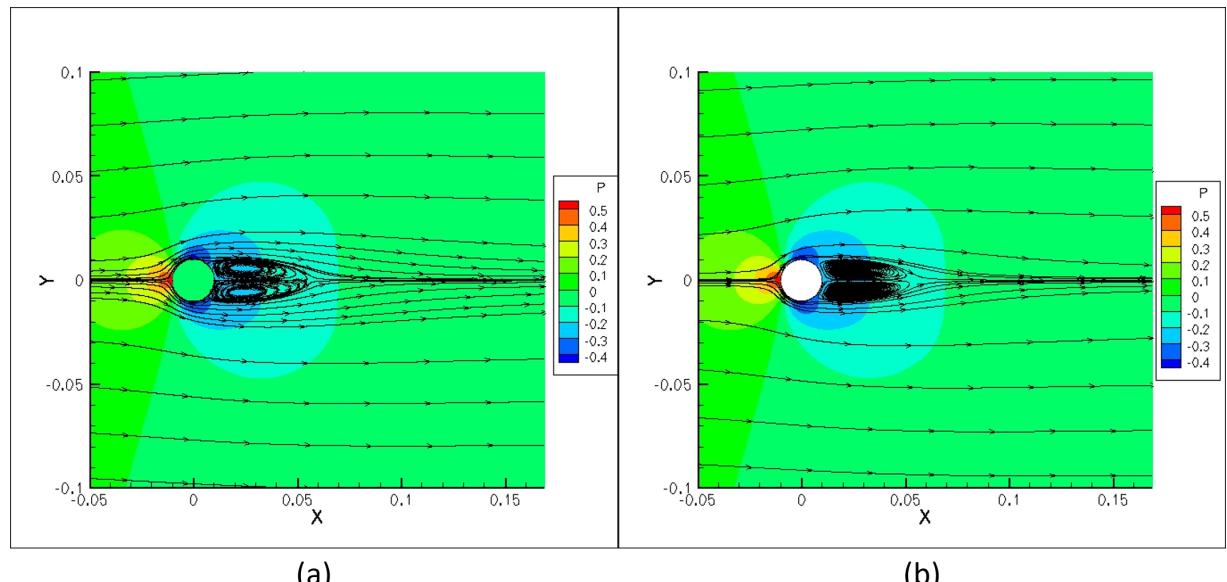


Figure 3.9: Pressure distribution with velocity streamlines in the IBM(a) and in the Body-fit Method (b) for $Re = 40$.

In Figure 3.9 the size of the eddies becomes even larger comparing it with the $Re = 10$ and $Re = 20$. Once again, Table 3.10 summarizes all the results from the literature.

Table 3.10: Results from literature survey and from the present work for $Re = 40$

Reference	year	L_w/D	C_D	B
Lima E Silva et al. [24]	2003	2.55	1.54	-
Ye et al. [2]	1999	2.27	1.52	-
Linnick and Fasel [26]	2005	2.23	1.61	0.056
Linnick and Fasel [26]	2005	2.23	1.54	0.023
Fornberg [27]	1980	2.24	1.5	-
Calhoun [28]	2002	2.18	1.62	-
Niu et al. [29]	2006	2.26	1.589	0.025
He and Doolen [25]	1997	2.25	1.499	0.018
Wang et al. [5]	2009	2.35	1.66	0.067
Xu and Wang [30]	2006	2.21	1.66	0.0625
Choi et al [31]	2007	2.25	1.52	0.0125
Pacheco et al. [32]	2005	2.28	1.53	0.033
Tseng and Ferziger [33]	2003	2.21	1.53	0.0625
Mittal et al. [34]	2008	-	1.53	0.025
Kim et al. [1]	2001	-	1.51	0.01
Frisani and Hassan [21]	2015	2.4	1.661	0.05
Frisani and Hassan [21]	2015	2.35	1.656	0.05
Frisani and Hassan [21]	2015	2.09	1.615	0.05
Frisani and Hassan [21]	2015	2.29	1.587	0.05
Frisani and Hassan [21]	2015	2.4	1.611	0.05
Sen et al. [23]	2009	-	1.514	-
Ding et al.[22]	2004	2.2	1.713	0.03125
Park et al. [20]	1998	-	1.51	0.02
Present work - Bodyfit	2016	2.15	1.50	0.01
Present work - STAR-CCM+	2016	-	1.53	0.025
Present work - IBM	2016	2.2	1.49	0.01

In this last case, the results for the drag coefficient can also be verified since they fall under the values from other authors.

3.1.8 2D Cylinder - Discussion

There are several conclusions that can be made from this case. Firstly, the domain size to study is very important, and by observing the results from the comparative table with the literature results for each Reynolds it becomes evident that the disparity of values can be caused by the different blockage values. It is simply not enough to perform analysis with a domain size chosen by comparison with other work. The error this effect can have in the results is large, so it is always necessary to try and make a compromise between domain size and computational power available for simulations.

It is also important to note that the IBM has more difficulties converging when a uniform refinement is performed when compared with a body-fit method. The computational power and time required in this case did not justify continuing the uniform refinement of the grid.

3.2 Analytical solution with Immersed Boundary Method and adaptive refinement

3.2.1 Analytical Singularity

In order to further test the Immersed Boundary Method, an analytical case would have to be selected to compare the IBM results to the analytical ones. This comparison is important due to the insight of the error quantification that an analytical solution can provide.

The analytical case chosen to study was the placement of a singularity inside a cylinder with a radius of 0.1 in the middle of a squared domain (1×1). This problem has an exact solution of the Navier-Stokes equations, which makes it ideal to test the IBM. The Reynolds number chosen to perform this analysis is $Re = 1000$. The decision of the problem to study and the Reynolds number was made following the work done in [35] and [36]. The equations that characterized the problem were altered in order to place the singularity inside the domain. The new equations 3.5, 3.6 and 3.7 give the exact solution for the horizontal, vertical velocities and pressure, respectively.

$$u = -\frac{2 \times (y - 0.5)}{(x - 0.5)^2 + (y - 0.5)^2} \quad (3.5)$$

$$v = \frac{2 \times (x - 0.5)}{(x - 0.5)^2 + (y - 0.5)^2} \quad (3.6)$$

$$p = -\frac{2}{(x - 0.5)^2 + (y - 0.5)^2} \quad (3.7)$$

where x and y are the Cartesian coordinates.

Figure 3.10 shows the analytical solution of the velocity magnitude.

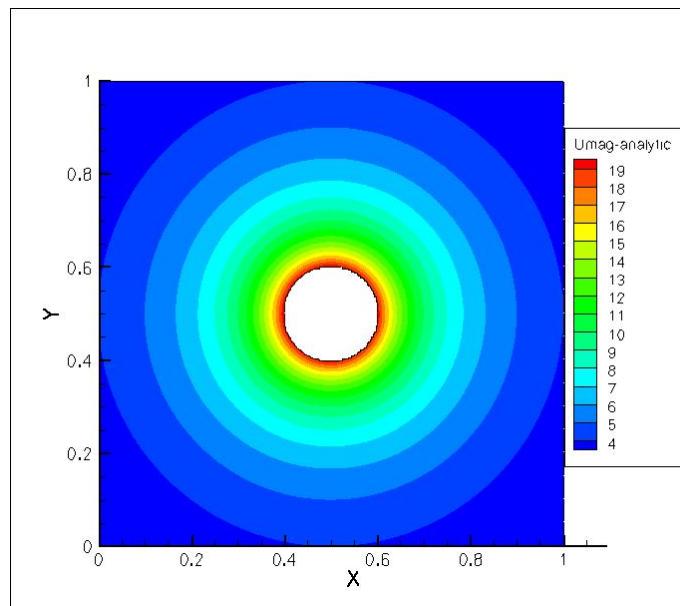


Figure 3.10: Analytical Singularity - analytical solution of the velocity magnitude.

In Figure 3.10 the domain used to analyse this case is clearly shown as is the cylinder inside where the singularity is contained. Having defined the case to study, it is now time to compare the different types of refinement and how does the error decay with each one. The most obvious criterion to begin this study is the uniform one where all the cells are refined, the results for this criterion are summarized in Table 3.11 and the adaptive algorithm start from a 20×20 initial grid. Table 3.11 groups the results for each refinement step's grid generated by each step of the refinement algorithm where $\bar{\varepsilon}$ is the average error and ε is the maximum error.

Table 3.11: Analytical Singularity - Error decay with Uniform Refinement

Refinement step	Cells	$\bar{\varepsilon}_u$	ε_u	$\bar{\varepsilon}_v$	ε_v	$\bar{\varepsilon}_{Umag}$	ε_{Umag}
0	400	8.44E-02	7.94E-01	8.44E-02	7.94E-01	1.00E-01	6.36E-01
1	1600	2.11E-02	2.71E-01	2.11E-02	2.71E-01	2.55E-02	2.20E-01
2	6400	5.01E-03	7.75E-02	5.01E-03	7.75E-02	5.76E-03	5.75E-02
3	25600	1.24E-03	2.66E-02	1.24E-03	2.66E-02	1.46E-03	2.09E-02
4	102400	3.63E-04	1.03E-02	3.63E-04	1.03E-02	4.76E-04	8.17E-03

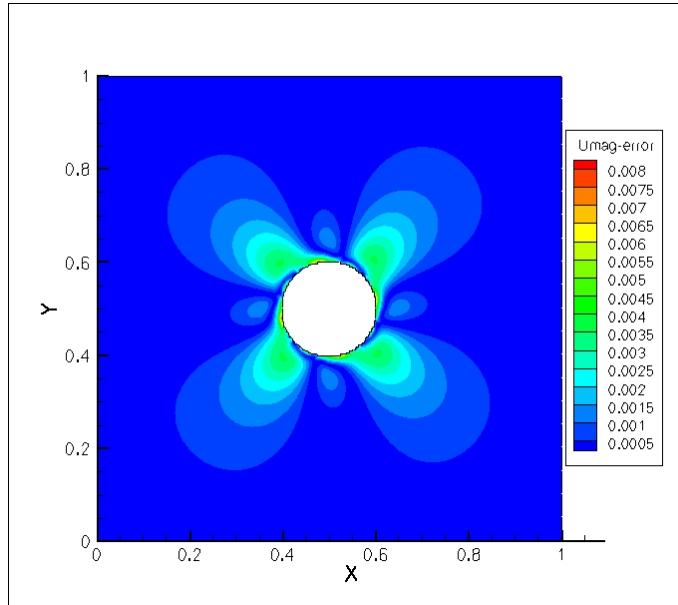


Figure 3.11: Analytical Singularity - Velocity magnitude error field after four refinements.

Figure 3.11 shows the error distribution of the domain. As error distribution is higher near the boundary of the cylinder, it is expected the other criteria will focus on refining the grid in this zone. This disparity between the error values between regions is what makes this a good case to test how efficient the error estimation and adaptive algorithm are.

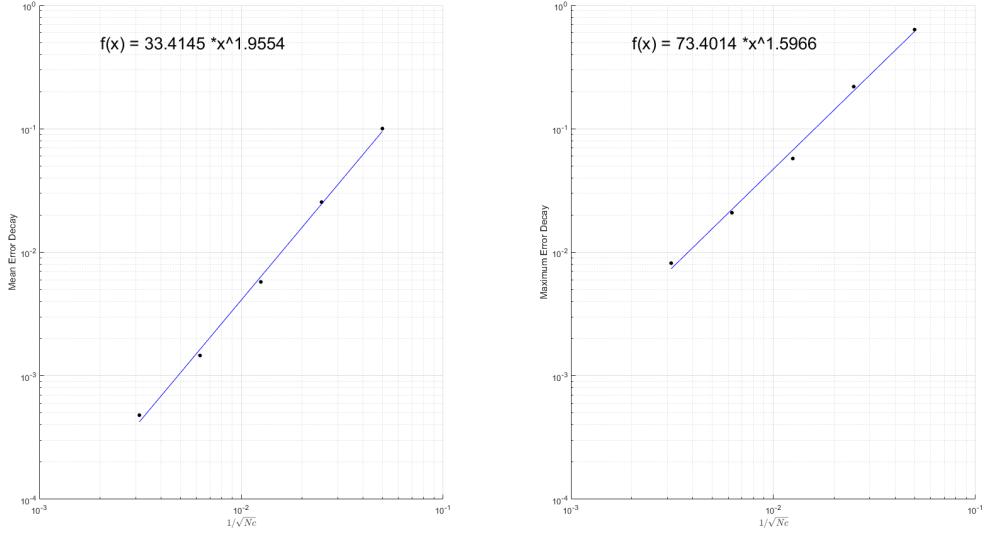


Figure 3.12: Analytical Singularity - Decay of the velocity magnitude error with $1/\sqrt{N_c}$ for the uniform criterion.

Since $E \propto h^2$ and $h \propto 1/\sqrt{N_c}$, the error decay is shown in function of $h \propto 1/\sqrt{N_c}$ because with the adaptive refinement, the value of h is not uniform. This is the benchmark case, every other criterion used to refine the initial grid should try to improve upon the previous results since this is a second order method but the maximum error does not decay with second order. The lower rate of decay makes this a good case to introduce the adaptive refinement.

With the benchmark set, the first criterion to test should be the analytical one because this is the most straightforward of the group since it is the analytical error that drives the adaptive refinement instead of using a error estimator. The results for this criterion are summarized in Table 3.12.

Table 3.12: Analytical Singularity - Error decay with Analytical criterion.

Refinement step	Cells	$\bar{\varepsilon}_u$	ε_u	$\bar{\varepsilon}_v$	ε_v	$\bar{\varepsilon}_{U\text{mag}}$	$\varepsilon_{U\text{mag}}$
0	400	7.76E-02	8.38E-01	7.76E-02	8.38E-01	9.11E-02	6.90E-01
1	772	2.23E-02	2.96E-01	2.23E-02	2.96E-01	2.76E-02	2.03E-01
2	1468	1.17E-02	7.45E-02	1.17E-02	7.45E-02	1.50E-02	5.31E-02
3	2620	9.93E-03	2.72E-02	9.93E-03	2.72E-02	1.28E-02	3.63E-02
4	4504	6.58E-03	2.20E-02	6.58E-03	2.20E-02	9.08E-03	2.35E-02
5	4852	4.34E-03	1.93E-02	4.34E-03	1.93E-02	6.00E-03	1.97E-02
6	6412	3.10E-03	9.93E-03	3.10E-03	9.93E-03	4.24E-03	1.12E-02
7	8032	2.47E-03	1.01E-02	2.47E-03	1.01E-02	3.15E-03	1.06E-02
8	10792	2.03E-03	6.66E-03	2.03E-03	6.65E-03	2.55E-03	5.89E-03
9	12412	1.79E-03	6.67E-03	1.79E-03	6.67E-03	2.31E-03	5.81E-03
10	17056	1.61E-03	5.28E-03	1.61E-03	5.28E-03	2.06E-03	5.37E-03

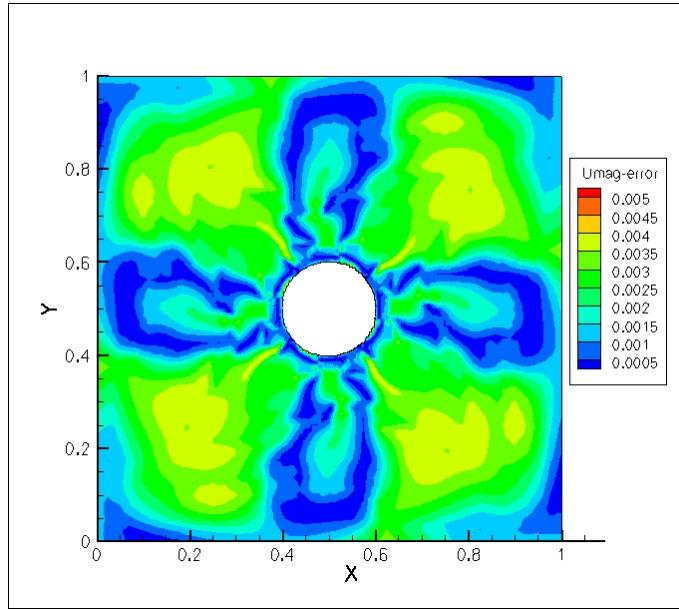


Figure 3.13: Analytical Singularity - Velocity magnitude error field for the level 10 of the analytical criterion.

Figure 3.13 shows a more even distributed error. This occurs because the criteria resulted in a more refined area near the cylinder leaving the diagonals with less cells.

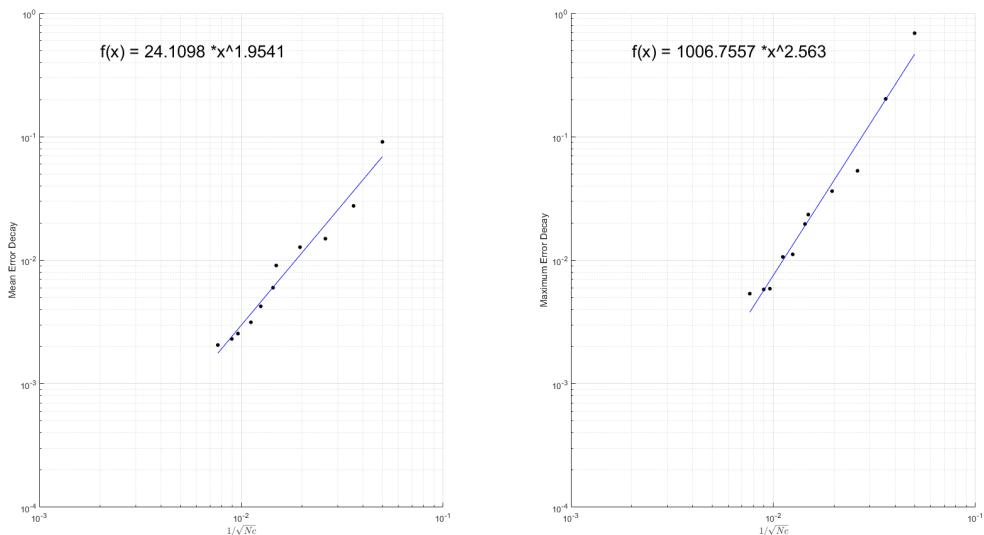


Figure 3.14: Analytical Singularity - Decay of the error with $1/\sqrt{N_c}$ for the analytical criterion.

Using this criterion the error decay increased almost one order of magnitude. The error decay is not as constant as in Figure 3.12 because of the grid interface correction, which prevents the accumulation of the grid interfaces between different levels of refinement. This phenomenon is more noticeable using the next criteria [14].

After the previous results, it is important to study how efficient the remaining criteria are at estimating the error and its location. The first error estimation criterion used is the Taylor criterion. Once again, the

results are summarized in Table 3.13.

Table 3.13: Analytical Singularity - Error decay with Taylor criterion.

Refinement step	Cells	$\bar{\varepsilon}_u$	ε_u	$\bar{\varepsilon}_v$	ε_v	$\bar{\varepsilon}_{Umag}$	ε_{Umag}
0	400	7.76E-02	8.38E-01	7.76E-02	8.38E-01	9.11E-02	6.90E-01
1	754	2.24E-02	2.96E-01	2.24E-02	2.96E-01	2.78E-02	2.03E-01
2	1435	1.17E-02	7.46E-02	1.19E-02	7.44E-02	1.51E-02	5.33E-02
3	3544	4.56E-03	2.39E-02	4.69E-03	2.40E-02	6.34E-03	2.06E-02
4	5284	2.70E-03	2.40E-02	2.81E-03	2.41E-02	3.56E-03	2.05E-02
5	8164	1.85E-03	2.41E-02	1.88E-03	2.41E-02	2.39E-03	2.07E-02
6	11368	1.40E-03	9.41E-03	1.49E-03	9.44E-03	1.94E-03	7.87E-03
7	12580	1.10E-03	9.41E-03	1.18E-03	9.44E-03	1.59E-03	7.87E-03
8	19288	7.93E-04	9.43E-03	8.54E-04	9.45E-03	1.11E-03	7.89E-03

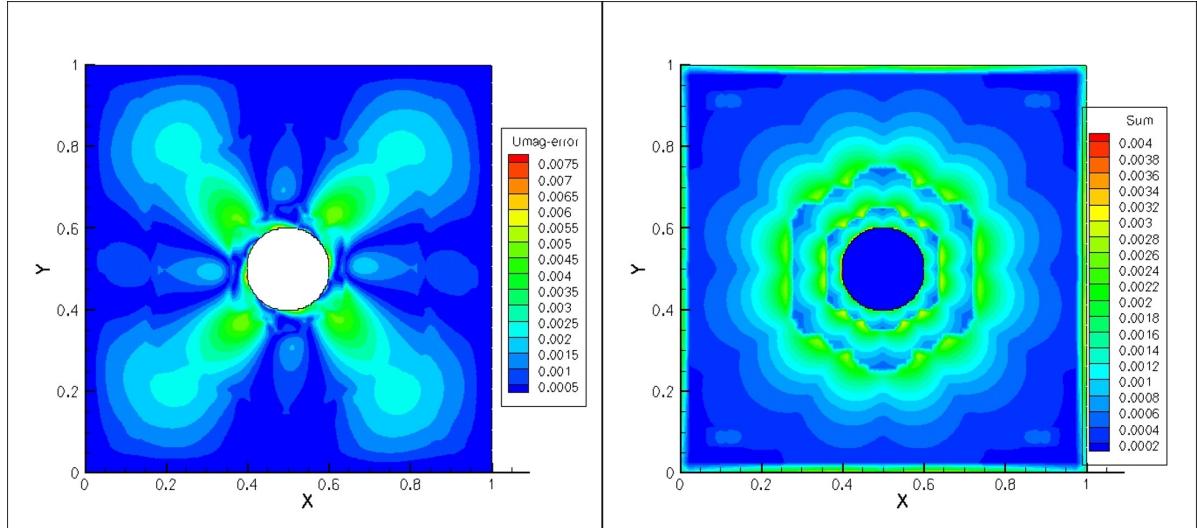


Figure 3.15: Analytical Singularity - Velocity magnitude error field on the left and error estimation criterion values on the right for the level 8 of the Taylor criterion.

The error distribution in Figure 3.15 is similar to the analytical criteria but with a higher error concentration in the diagonal in the regions near the cylinder. One important thing to note from error estimation distribution from Figure 3.15 is that the Taylor criterion over-estimated the error in the computational boundary. This will lead to an increase in the cells number, decreasing the computational efficiency.

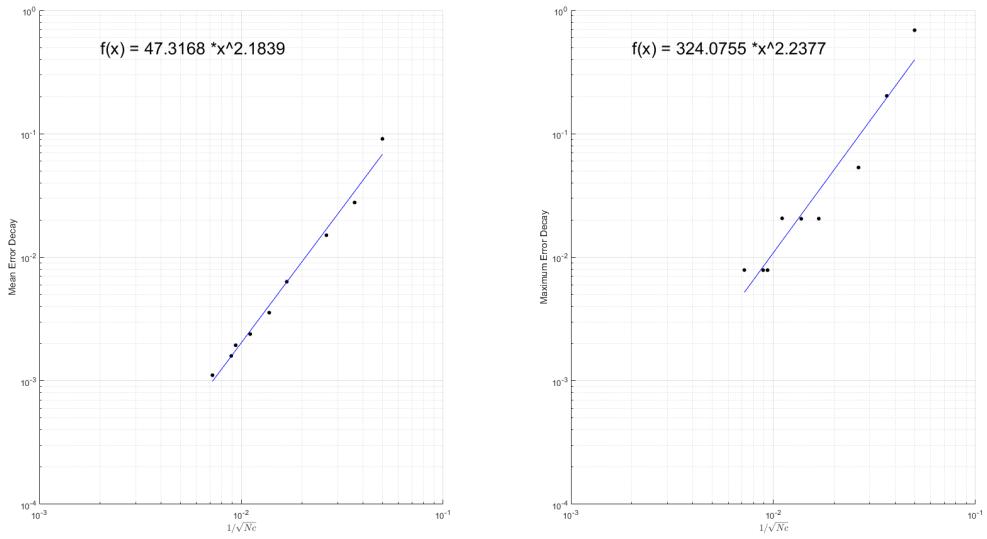


Figure 3.16: Analytical Singularity - Decay of the error with $1/\sqrt{N_c}$ for the taylor criterion.

Both mean and maximum error have a second order accuracy. The results achieved with this criterion are very similar to the ones obtained with the analytical one. This is a good indication of how well the criterion is working.

Finally, the last error estimation criterion can be tested. The residual criteria can now be studied and see how the results compare with the previous criteria. The results for this criterion are summarized in Table 3.14.

Table 3.14: Analytical Singularity - Error decay with Residual criterion.

Refinement step	Cells	$\bar{\varepsilon}_u$	ε_u	$\bar{\varepsilon}_v$	ε_v	$\bar{\varepsilon}_{Umag}$	ε_{Umag}
0	400	7.76E-02	8.38E-01	7.76E-02	8.38E-01	9.11E-02	6.90E-01
1	892	2.14E-02	2.96E-01	2.14E-02	2.96E-01	2.56E-02	2.02E-01
2	1840	9.78E-03	7.44E-02	9.78E-03	7.44E-02	1.17E-02	5.28E-02
3	2428	6.00E-03	7.44E-02	6.00E-03	7.44E-02	7.50E-03	5.29E-02
4	3196	5.43E-03	7.44E-02	5.43E-03	7.44E-02	6.30E-03	5.28E-02
5	5344	3.01E-03	2.41E-02	3.01E-03	2.41E-02	3.82E-03	2.06E-02
6	14140	1.47E-03	9.47E-03	1.47E-03	9.47E-03	1.90E-03	7.91E-03
7	40432	5.30E-04	2.69E-03	5.30E-04	2.70E-03	6.77E-04	2.69E-03

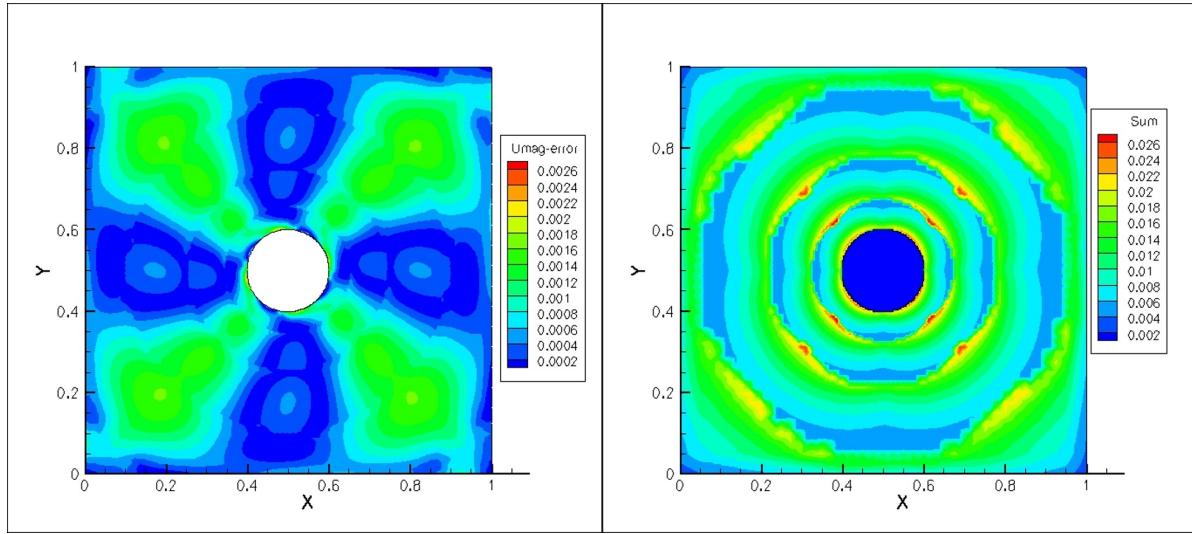


Figure 3.17: Analytical Singularity - Velocity magnitude error field on the left and error estimation criterion values on the right for the level 7 of the residual criterion.

The magnitude's velocity error distribution shape in Figure 3.17 is similar to the one in Figure 3.13. This similarity suggests the Residual criterion follows a close strategy to the Analytical one, refining the local area in the vicinity of the cylinder. From the error estimation distribution for the residual criterion, it is possible to see that this criterion is better at detecting the error in the diagonals of the domain and that it does not over-estimates the error in the computational boundaries.

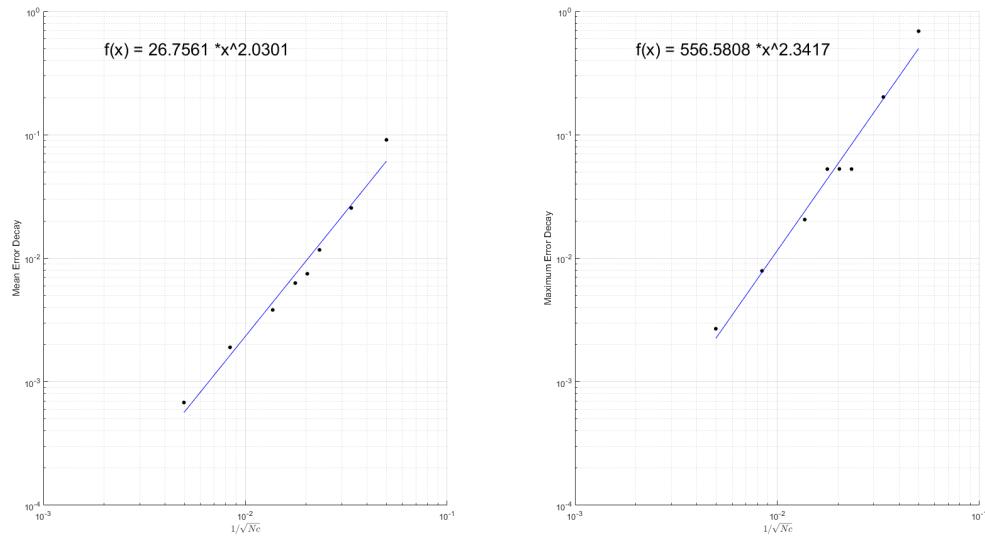


Figure 3.18: Analytical Singularity - Decay of the error with $1/\sqrt{N_c}$ for the Residual criterion.

3.2.2 Analytical singularity - Discussion

The adaptive grids obtained at the final levels of refinement for each criterion are shown in Figure 3.19.

The choice to place the singularity inside the cylinder was made to try and create a case-study with

local error which would put the error criteria at test. With this case, the benefits of such criteria would be apparent as in fact they did. The increase on the convergence order using local refinement criteria means that the use of this type of criteria allow for a better efficiency since the error can be reduced with less number of cells, reducing the required memory and time to solve the problem for a certain error level.

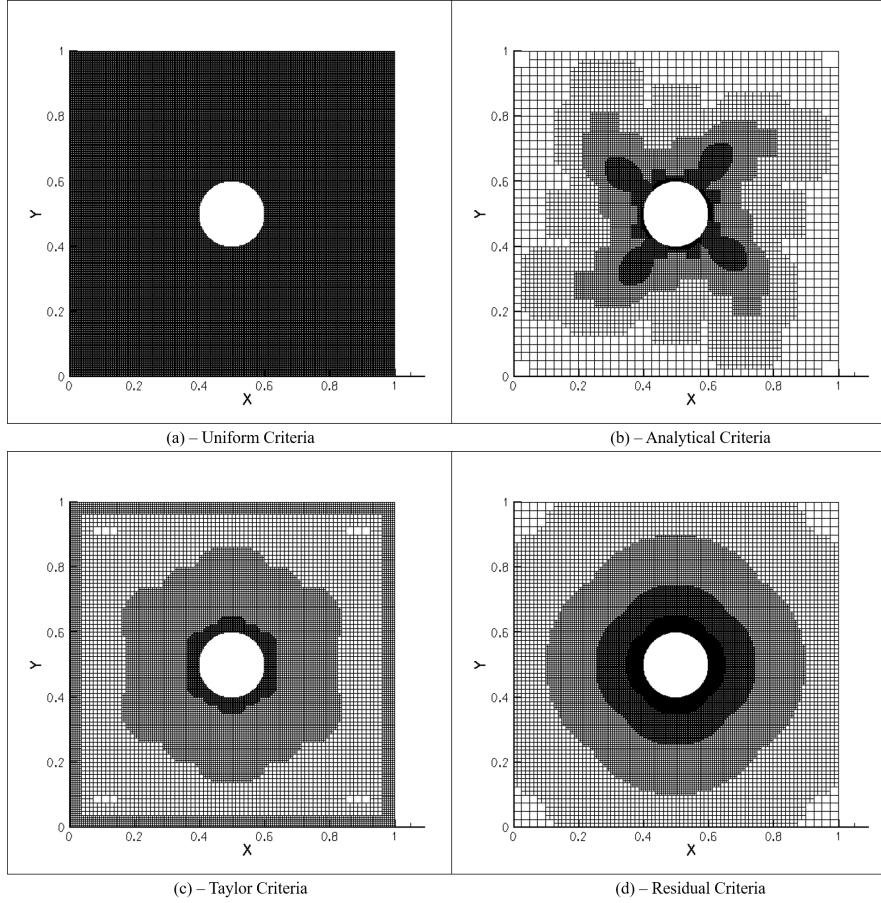


Figure 3.19: Analytical Singularity: Final grids for the different criteria.

For a better comparison between each criterion, a measure of efficiency should be used. The measure chosen is the ratio between mean and maximum errors. In Table 3.15 the efficiency values for the most refined grids of each error estimation criteria can be analysed. The uniform criterion is the one with the lowest efficiency and the analytical one has the highest efficiency as expected. The real comparison is between the Taylor and Residual error estimators and the Residual criterion seems to be far more efficient.

Table 3.15: Analytical Singularity: Efficiency Values for the most refined grid of each criteria.

	μ_u	μ_{mag}
Uniform criterion	0.0352	0.058
Analytical Criterion	0.305	0.384
Taylor Criterion	0.084	0.14
Residual Criterion	0.197	0.252

In Figure 3.20 a comparison between criteria is presented. The analytical adaptive criterion is the

best at lowering the maximum error but both error estimation criteria exhibit a similar behaviour, evidencing their good capabilities in reducing the computational error.

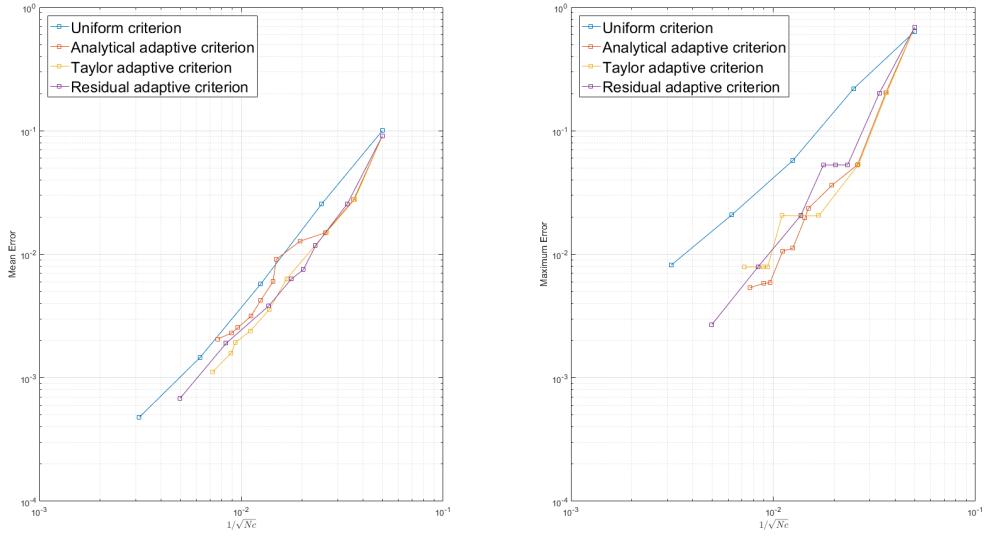


Figure 3.20: Analytical Singularity - Comparison between the different criteria of the error decay with $1/\sqrt{N_c}$.

This case-study is also a good opportunity to compare the diffusive and convective schemes used with others available in the SOL code. Figure 3.21 presents the results for the analytical adaptive criteria with different schemes. The GPS is the ghost points family scheme, the FLS uses the Face Least Squares scheme explained in section 2.2.5 and the pair GQS uses a Least Squares scheme for the diffusion and a triangular interpolation with skewness correction as the convection scheme.

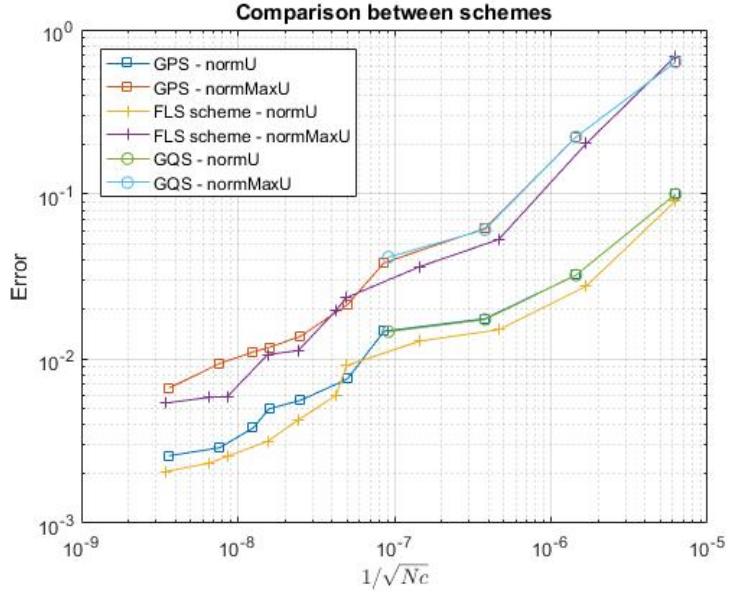


Figure 3.21: Analytical Singularity - Comparison between schemes for the analytical criterion.

After a simple analysis of Figure 3.21 the FLS scheme is clearly proven to be a better choice for

performing all the computational analyses with adaptive grids since the decrease in error is more evident with this scheme and it does not have the same instability problem of pair GQS that caused the simulation to diverge.

Chapter 4

IBM with automatic adaptive refinement

The main purpose of this work was to implement an automatic refinement criteria that would be able to reduce the computational power needed to achieve accurate solutions using the Immersed Boundary method. In the previous chapter, these criteria were compared with computational methods from other works and with analytical solutions, where it's possible to conclude that these error estimation criteria allow for solutions with a better grid convergence, improving the capabilities of the initial IBM.

In the present chapter, the results with the verified adaptive algorithm for different flows will be presented. Section 4.1 will once again study the case of a flow with an immersed cylinder. This case was extensively studied in 3.1 in order to compare the IBM to other computational methods. It is now time to see how the implementation of the previously developed error estimation criteria altered the results.

In Section 4.2, a NACA airfoil was chosen and was then embedded in the flow. The choice of this case was made due to the more complex geometry and because there was already several data in the literature (see [37], [38], [39] and [40]) that helped verifying the obtain results.

4.1 Flow Around a 2D Cylinder with adaptive criteria

Once again, the present section studied the flow around a 2D cylinder. This section was already extensively introduced in Section 3.1. At this time, only the effects of the implemented adaptive criteria remain to be seen. These effects can be studied performing a new convergence study of the drag coefficient C_D and comparing it with the results of the body-fit grid with more cells in Section 3.1.

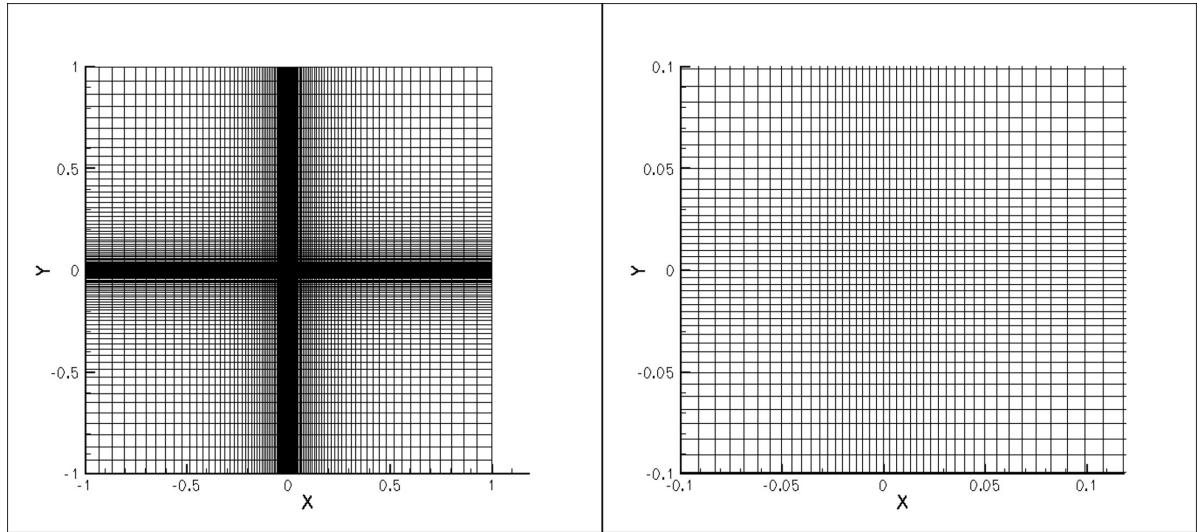


Figure 4.1: Initial grid with a close up on the cylinder location on the right for the adaptive refinement simulation of the 2D cylinder case.

The simulations in this chapter were conducted for $Re = 10$, $Re = 20$ and $Re = 40$ as before and the same domain size was used which is free of confinement effects. The only alteration when comparing this Section to Section 3.1 is the use of a more coarse grid. The decision to use a coarser grid was made since the criteria will only increase the number of cells where it is needed. Lowering the number of cells in the initial grid conserves computational power. The initial grid has only 10000 cells and it is shown in Figure 4.1.

4.1.1 Reynolds 10

Since this case was already introduced, the same steps will now be performed using the adaptive refinement. To start, the results for $Re = 10$ will be analysed.

Starting with the Taylor adaptive criterion, the results are summarized in Table 4.1. This Table shows the number of cells and the values for the total drag coefficient and for each one of its components. The iterative error is calculated with:

$$\epsilon = \frac{C_D^{k+1} - C_D^k}{C_D^{k+1}}, \quad (4.1)$$

where C_D^k is the drag coefficient result for the current step and C_D^{k+1} is the result for the next step with the more refined grid.

Table 4.1: $Re = 10$ - Convergence study for the Taylor adaptive criterion.

Level	Cells	C_D	C_{Dp}	C_{Dv}	Iterative convergence (%)
0	10000	2.260	1.335	0.926	10.641
1	11383	2.530	1.460	1.070	6.290
2	13888	2.699	1.520	1.180	0.113
3	17536	2.702	1.520	1.182	2.166
4	25399	2.762	1.546	1.217	0.036
5	40621	2.763	1.550	1.214	-0.003
6	69751	2.763	1.553	1.210	0.536
7	127915	2.778	1.558	1.220	0.020
8	244423	2.779	1.559	1.220	-

As expected, the results in the drag coefficient convergence show an evolution by steps as the results presented in Section 3. Since the difference between the drag coefficient of level 7 and 8 is about 0.02%, and that small improvement of the drag coefficient value required the number of cells to double, it is reasonable to consider that the grid of level 7 has converged. The respective pressure distribution and grid is shown in Figure 4.2.

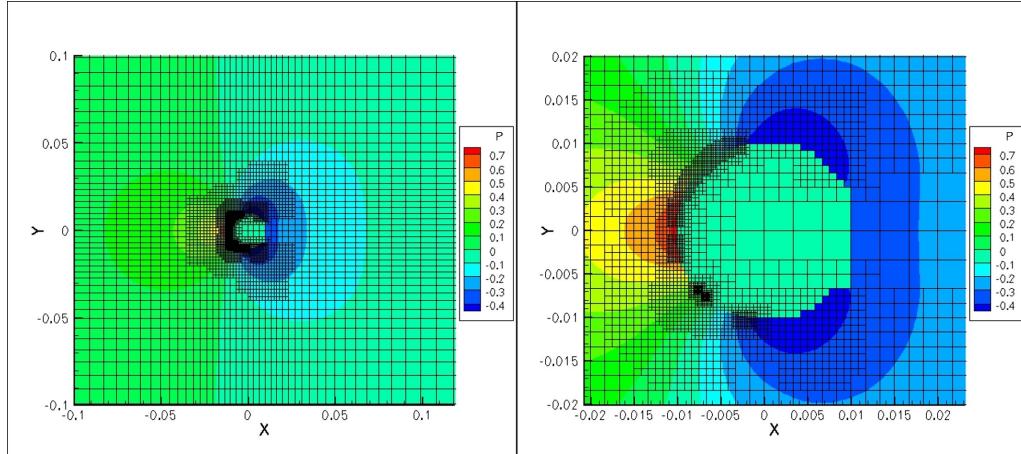


Figure 4.2: Pressure distribution for the level 7 of the Taylor criterion for $Re = 10$.

As expected the pressure distribution is very similar to the one presented in Section 3.1 for the respective Reynolds number. The error for this criteria when comparing the results to the body-fit case is 0.13%. It is important to note that this percentage of error was achieved with a 72% reduction in the number of cells of the body-fit grid.

The other criterion used can now be studied and compared with the previous results. Table 4.2 summarizes the results for the Residual criterion.

Table 4.2: $Re = 10$ - Convergence study for the Residual criterion.

Level	Cells	C_D	C_{Dp}	C_{Dv}	Iterative convergence (%)
0	10000	2.260	1.335	0.926	10.315
1	10894	2.520	1.455	1.066	6.447
2	13918	2.694	1.526	1.168	-0.002
3	17590	2.694	1.526	1.168	-0.092
4	23116	2.692	1.524	1.167	2.099
5	29416	2.749	1.549	1.201	-0.002
6	35938	2.749	1.549	1.201	-0.001
7	51880	2.749	1.549	1.201	-0.038
8	72106	2.748	1.548	1.200	0.978
9	90058	2.775	1.560	1.215	0.423
10	182320	2.787	1.565	1.222	-0.016
11	266860	2.787	1.565	1.222	0.112
12	560668	2.790	1.568	1.222	-

Once again, the results show an evolution by steps. Level 10 has already achieved a good convergence rate since the drag coefficient changes very little. Since the number of cells triples this is considered a good compromise between computational accuracy and the computational power required. Pressure distribution and grid for Level 10 of Residual criterion is shown in Figure 4.3 and is congruent with the expected results.

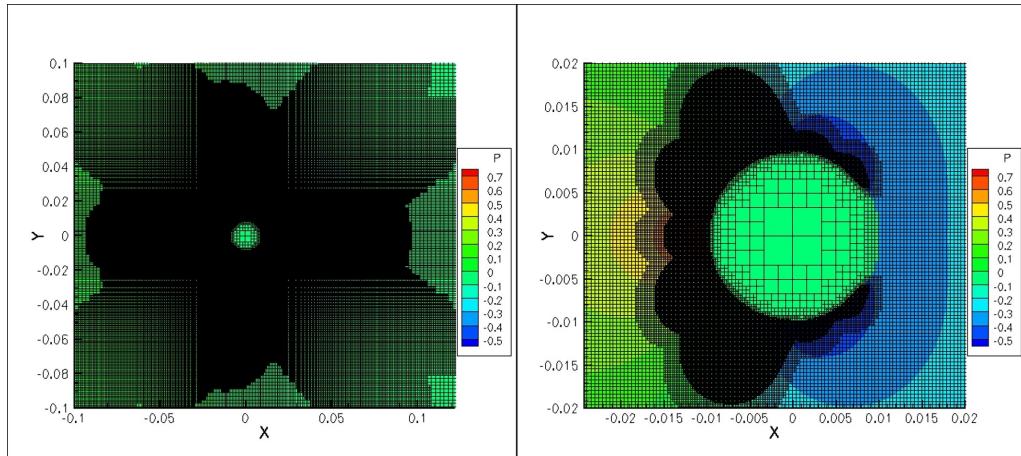


Figure 4.3: Pressure distribution for the level 10 of the Residual criterion for $Re = 10$.

The grid for this criterion is more uniformly refined comparing with the taylor criterion. This will result in a higher number of cells but with a smoother pressure distribution since the different levels of refinement are further apart. The Residual criterion has an error of 0.532% achieved with a 60% reduction of the number of cells.

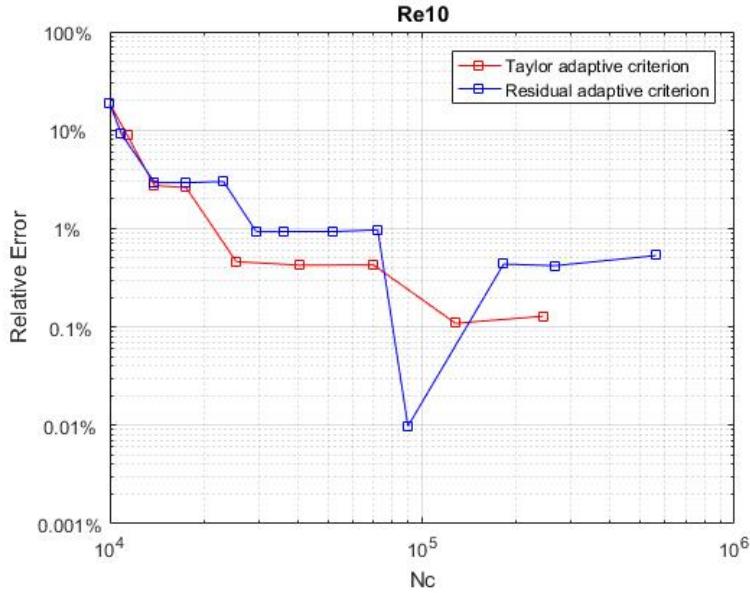


Figure 4.4: Relative error between both adaptive criteria and the body-fit method for $Re = 10$.

In Figure 4.4 it is possible to see how the drag coefficient error converges with the increase in cell's numbers. Both criteria are fast to converge but the Residual adaptive criteria does not over-estimate the error in the computational boundaries and it refines a more broader area of the cylinder. The Taylor adaptive criteria on the other hand, adds unnecessary cells to the computational boundary and the refinement in the cylinder zone is less smooth.

4.1.2 Reynolds 20

The second Reynolds number to study is $Re = 20$. The first criterion to study will be, once again, the Taylor criterion and its results are summarized in Table 4.3.

Table 4.3: $Re = 20$ - Convergence study for the Taylor criterion.

Level	Cells	C_D	C_{Dp}	C_{Dv}	Iterative convergence (%)
0	10000	1.621	1.043	0.578	10.422
1	11332	1.810	1.122	0.688	7.645
2	13912	1.960	1.182	0.778	0.000
3	17515	1.960	1.182	0.778	2.384
4	25408	2.008	1.201	0.807	-0.042
5	40255	2.007	1.201	0.806	-0.113
6	69364	2.005	1.206	0.799	-0.194
7	127021	2.001	1.206	0.795	0.000
8	242221	2.001	1.206	0.795	0.000
9	472621	2.001	1.206	0.795	-

For $Re = 20$ the drag coefficient values are fast to converge. Level 7 of the convergence study has already converged. The error for this level is of only 0.5% and the drag coefficient value do not change even when the number of cells triples. This error percentage was achieved with a 77% reduction in the number of cells in the grid. The results for this level are shown in Figure 4.5.

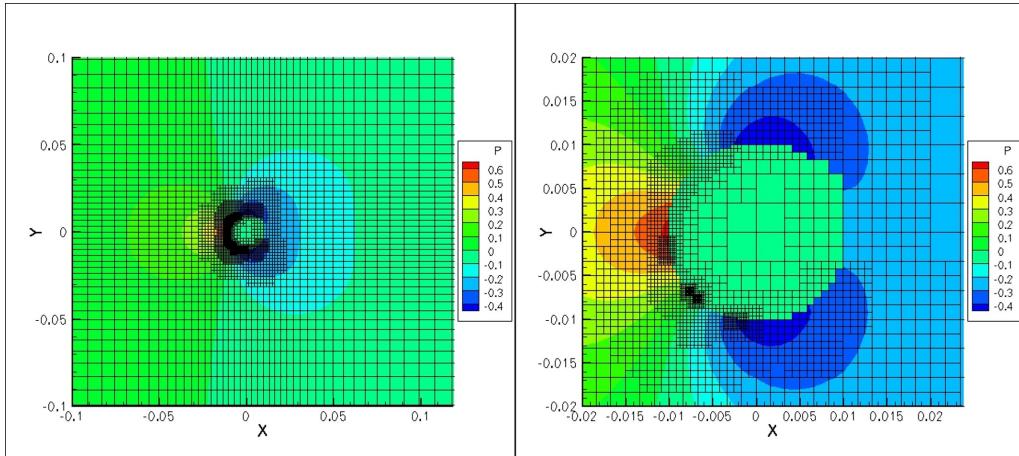


Figure 4.5: Pressure distribution for the level 7 of the Taylor criterion for $Re = 20$.

The results shown in Figure 4.5 were the ones expected, with a very small area being more refined resulting in different levels of refinement near each other.

Applying the Residual criterion, the results achieved are shown in Table 4.4.

Table 4.4: $Re = 20$ - Convergence study for the Residual criterion.

Level	Cells	C_D	C_{Dp}	C_{Dv}	Iterative convergence (%)
0	10000	1.621	1.043	0.578	10.387
1	10816	1.809	1.127	0.683	-0.047
2	11914	1.809	1.126	0.682	6.974
3	13960	1.944	1.185	0.759	-0.005
4	16222	1.944	1.185	0.759	-0.099
5	20950	1.942	1.183	0.759	2.271
6	26686	1.987	1.203	0.784	-0.004
7	31324	1.987	1.203	0.784	-0.002
8	43906	1.987	1.203	0.784	-0.063
9	62026	1.986	1.203	0.783	1.054
10	79408	2.007	1.212	0.795	0.448
11	159508	2.016	1.215	0.801	-0.029
12	217534	2.015	1.215	0.800	0.205
13	414712	2.020	1.217	0.803	-

The small change in the results from the final levels is the results of a relevant increase in the number of cells. For example, the results between level 12 and 13 differ 0.2% but this change required the number of cells to double. Such a small change in the results does not justify the increase of computational power needed. Level 11 is then considered to have converged and this is the grid with the best compromise between accuracy and computational power. Results are now shown in Figure 4.6.

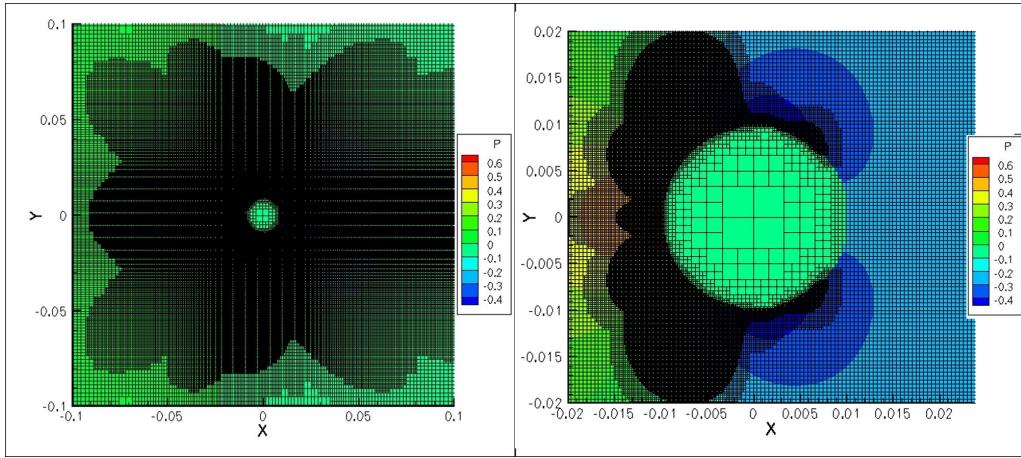


Figure 4.6: Pressure distribution for the level 11 of the Residual criterion for $Re = 20$.

As seen for $Re = 10$ the residual criterion tends to present a more smooth grid, and Figure 4.6 shows that exactly. The error for this case is of 0.25% and it is achieved with a 65% of the number of cells.

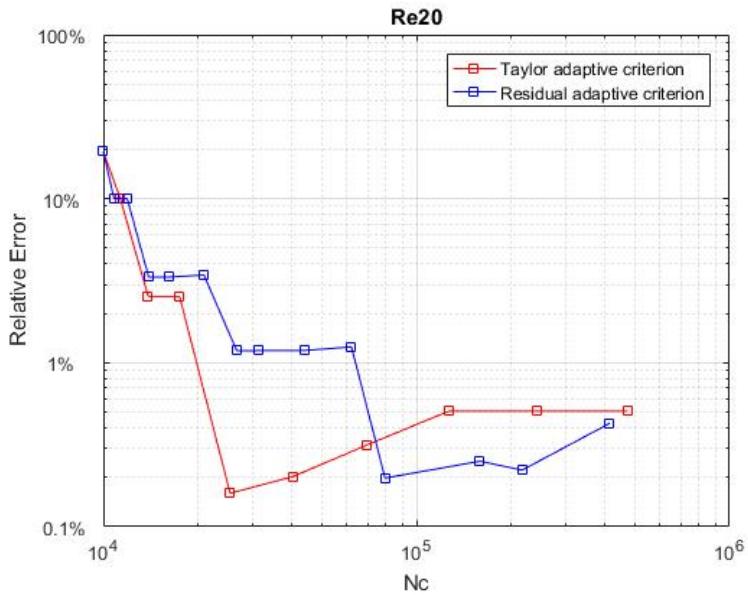


Figure 4.7: Relative error between both adaptive criteria and the body-fit method for $Re = 20$.

The same conclusions as for $Re = 10$ can be made for Figure 4.7 but now, the Taylor adaptive criterion is even faster to converge.

4.1.3 Reynolds 40

Finally, the third Reynolds number to study is $Re = 40$. The first criterion presented is once again the Taylor criteria and its results are summarized in Table 4.5.

Table 4.5: $Re = 40$ - Convergence study for the Taylor criterion.

Level	Cells	C_D	C_{Dp}	C_{Dv}	Iterative convergence (%)
0	10000	1.200	0.864	0.336	-9.950
1	11287	1.333	0.905	0.428	-0.661
2	13159	1.342	0.917	0.424	0.139
3	13207	1.340	0.916	0.424	-7.815
4	17404	1.453	0.955	0.498	0.000
5	24607	1.453	0.955	0.498	-2.694
6	39757	1.494	0.972	0.522	-0.059
7	69025	1.494	0.974	0.521	0.075
8	126886	1.493	0.976	0.517	0.000
9	242086	1.493	0.976	0.517	-0.693
10	472486	1.504	0.999	0.505	-

Again, the final levels has the lowest error percentage of 0.08%. But this decrease in the error percentage was achieved with a considerable increase in the number of cells. Since the error of level 6 is about 0.76% and this small error percentage was achieved with only 40 thousand cells this was the grid chosen to present the results in Figure 4.8.

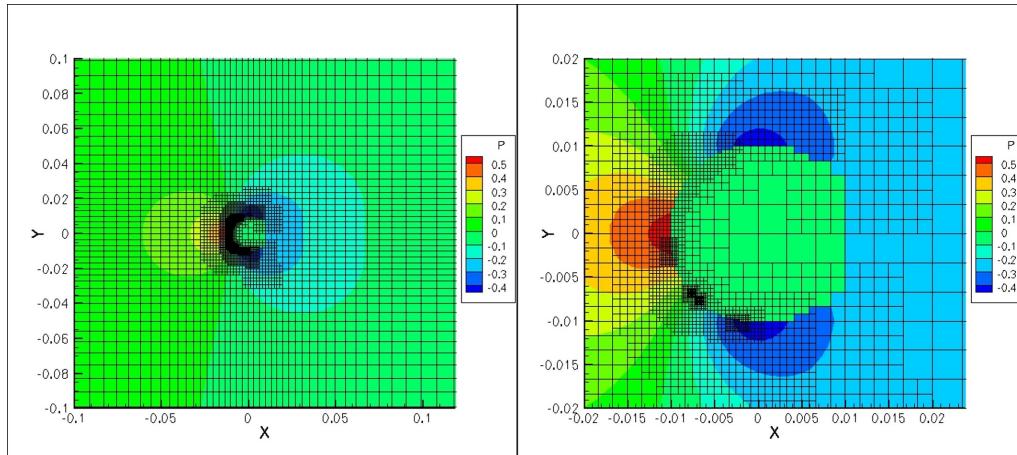


Figure 4.8: Pressure distribution for the level 10 of the Taylor criterion for $Re = 40$.

The final grid shows, once more, the different levels of refinement in a small area. This grid allowed a 91% reduction in the number of cells for an error of only 0.76%.

The last results to be present are the ones correspondent to the Residual adaptive criterion. Its results are summarized in Table 4.6.

Table 4.6: $Re = 40$ - Convergence study for the Residual criterion.

Level	Cells	C_D	C_{Dp}	C_{Dv}	Iterative convergence (%)
0	10000	1.200	0.864	0.336	-10.248
1	10676	1.337	0.915	0.422	0.108
2	11770	1.336	0.914	0.422	-7.812
3	13534	1.449	0.962	0.486	0.114
4	16288	1.447	0.961	0.486	0.189
5	19552	1.445	0.959	0.485	-2.542
6	24610	1.482	0.978	0.505	0.006
7	29356	1.482	0.977	0.505	0.003
8	40360	1.482	0.977	0.505	0.051
9	53536	1.481	0.977	0.504	-1.197
10	68344	1.499	0.985	0.514	-0.499
11	140044	1.507	0.988	0.519	0.044
12	185668	1.506	0.988	0.518	-0.230
13	368554	1.510	0.990	0.520	-

For this criterion, Level 9 of the convergence study can be considered to have converged since the drag coefficient barely increases from this value onward. Such a small change in values does not justify doubling the number of cells. The results for the pressure distribution are shown in Figure 4.9.

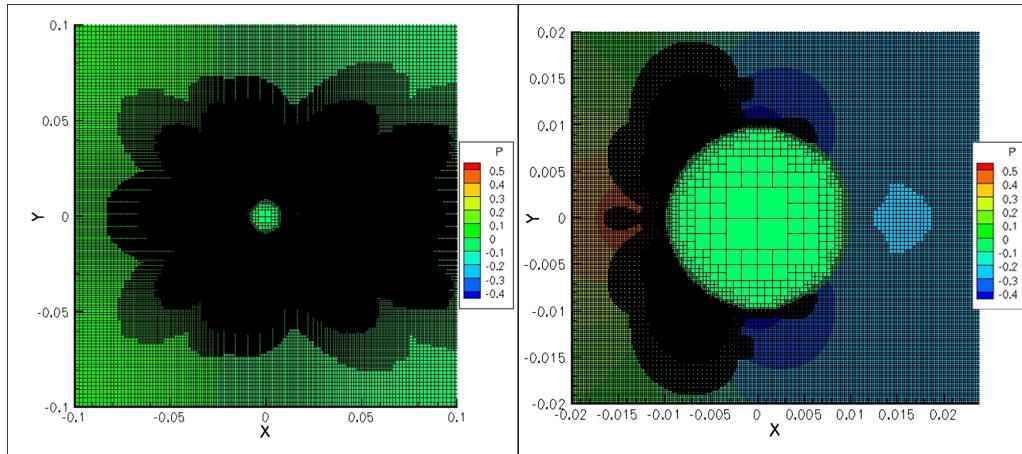


Figure 4.9: Pressure distribution for the level 13 of the Residual criterion for $Re = 40$.

The final grid for the Residual criterion is even more smoothly refined with some space between levels of refinement. The error for this case is 0.12% and it was achieved with a 88% reduction of the number of cells.

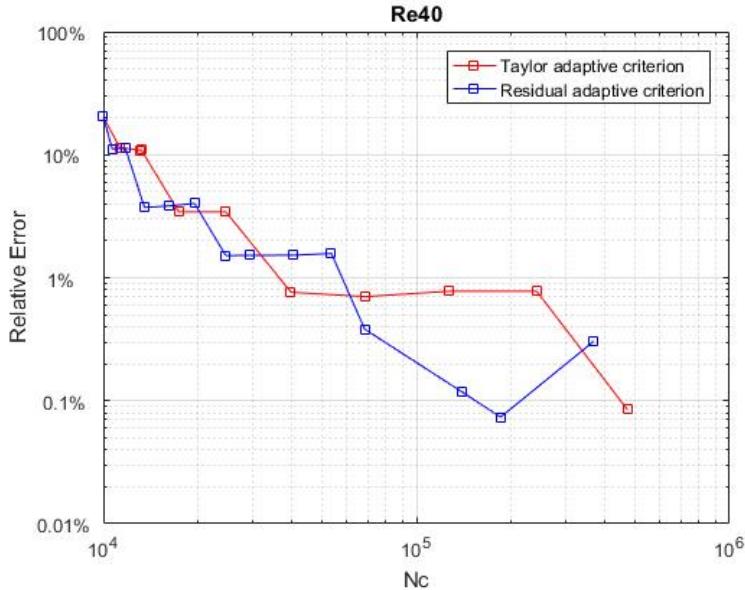


Figure 4.10: Relative error between both adaptive criteria and the body-fit method for $Re = 40$.

In Figure 4.10 it becomes more apparent the values obtained through the Taylor adaptive criterion tend to approximate those of the Residual adaptive criterion with the increase in number of cells. One of the reasons why the Taylor adaptive criterion needs a larger number of cells is the unnecessary refinement in the computational boundaries.

4.1.4 2D Cylinder with adaptive criteria

The results from the present section clearly show the advantages of coupling the adaptive refinement algorithm with the IBM when compared with the results without the algorithm and even with the results from the body-fit method. The error lower than 1% when compared with the body-fit method with less than half the number of cells, clearly shows this method produces more efficient grids. Furthermore, the jump in the relative error that occurred for each Reynolds number, may even imply that the accuracy of the results obtained in this section is even better than the body-fit ones.

4.2 Flow Around the NACA 0012 airfoil

In the present section, another case where the use of refinement criteria can be tested is studied. The body chosen to immersed in a flow was an airfoil. The NACA 0012 was the chosen airfoil and it is symmetric with a 12% thickness of the chord length. Using a thin airfoil simplifies this specific problem because there is no need to develop turbulence models since the flow remains laminar for the chosen Reynolds number. The airfoil was designed according to Equation 4.2 where the x is the adimensional horizontal coordinate, c is the chord value and y_t is the half thickness value to the correspondent x .

$$y_t = 5 \times 0.12 \times \left[0.2969 \sqrt{\frac{x}{c}} - 0.1260 \left(\frac{x}{c} \right) - 0.3516 \left(\frac{x}{c} \right)^2 + 0.2843 \left(\frac{x}{c} \right)^3 - 0.1015 \left(\frac{x}{c} \right)^4 \right] \quad (4.2)$$

Figure 4.11 shows the shape of the selected airfoil after its coordinates that were calculated using Equation 4.2.

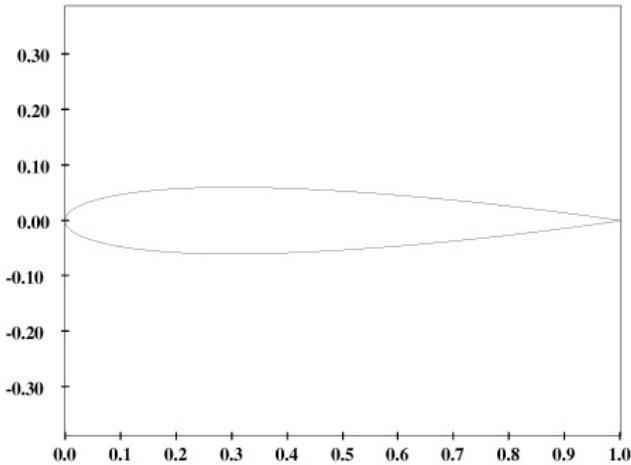


Figure 4.11: NACA 0012 airfoil shape.

This case was already studied in [37], [38], [39] and [40] for laminar flow and different Reynolds numbers. The Reynolds number can be calculated using Equation 4.3 and it is a function of the inlet velocity, chord length and of the kinematic viscosity.

$$Re = \frac{U_\infty c}{\nu} \quad (4.3)$$

The Reynolds number chosen to perform the simulations of this case was $Re = 500$ since according to [[37] - [40]] this guarantees a steady flow around the airfoil. The domain size is set to guarantee a Blockage effect factor of 0.01. This study is performed at $\alpha = 0^\circ$ and $\alpha = 10^\circ$. These two different attack angles should produce flows with different characteristics.

The initial grid is shown in Figure 4.12 and it has 32320 cells.

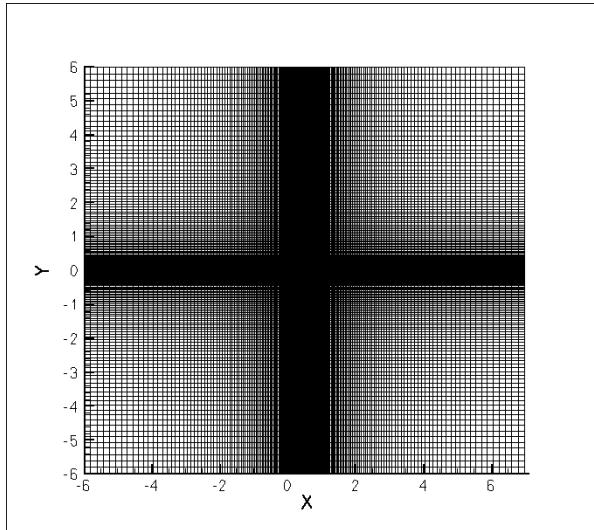


Figure 4.12: Initial grid for NACA0012 airfoil.

To compare the results using the Immersed Boundary method it was decided to also study this flow using a Body-fit grid. This grid was generated using a polyhedral grid algorithm in Matlab. The grid that resulted from the mesh generator is presented in Figure 4.13 and it has 19867 cells. The close-up of the Body-fit shows that the shape of the airfoil is well defined and that the grid should produce accurate results.

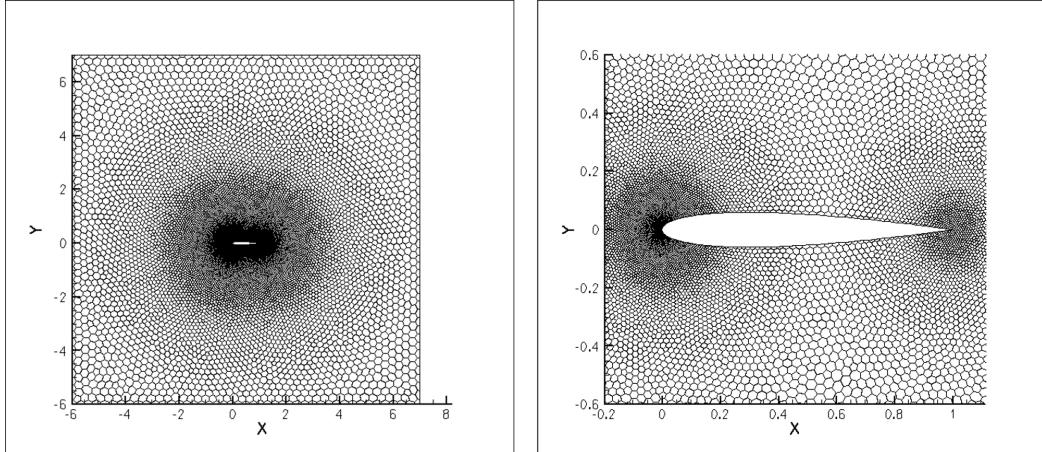


Figure 4.13: Body-fit grid for NACA0012 airfoil.

Firstly, the results for $\alpha = 0^\circ$ will be presented and analysed since this is a more straightforward case where the flow should remain laminar and not separate from the airfoil. Figure 4.14 summarizes some results found in the literature for $\alpha = 0^\circ$. These values will be compared to the results obtained with the IBM.

Resolution (on wing)	Δx_{\min}	C_d	C_l
<i>GILBM</i>			
257×65 (173)	$4.5e-3$	0.1682	$1.0e-13$
373×141 (251)	$4.5e-3$	0.1672	$1.0e-13$
257×65 (173)	$4.5e-4$	0.1736	$1.0e-13$
373×141 (251)	$4.5e-4$	0.1725	$1.0e-13$
<i>PowerFLOW</i>			
159,060 (828)	$1.2e-3$	0.1717	$0.227e-3$
418,800 (1275)	$7.8e-4$	0.1807	$-0.211e-3$
<i>CFL3D</i>			
257×65	—	0.1762	$0.115e-6$
373×141	$1.2e-4$	0.1741	$0.538e-5$

Figure 4.14: Naca0012 grid resolution dependence for $\alpha = 0^\circ$ from [40].

From Table 4.7 it is finally possible to analyse the results of the drag and lift coefficients. The results for the Taylor and Residual criteria are very close to the body-fit method, which confirms that the IBM is improved when it is paired with an adaptive refinement.

Table 4.7: NACA 0012 results for $\alpha = 0^\circ$.

Grid	Cells	C_d	C_l
Body-fit	93330	0.1728	0.0003
Uniform refinement criteria	2068480	0.1687	0.0034
Taylor Adaptive Criteria	402328	0.1752	0.0003
Residual Adaptive Criteria	110908	0.1773	0.0051

Figures 4.15, 4.16 and 4.17 show the Pressure distribution, horizontal and vertical velocities distributions respectively for the several grid and methods.

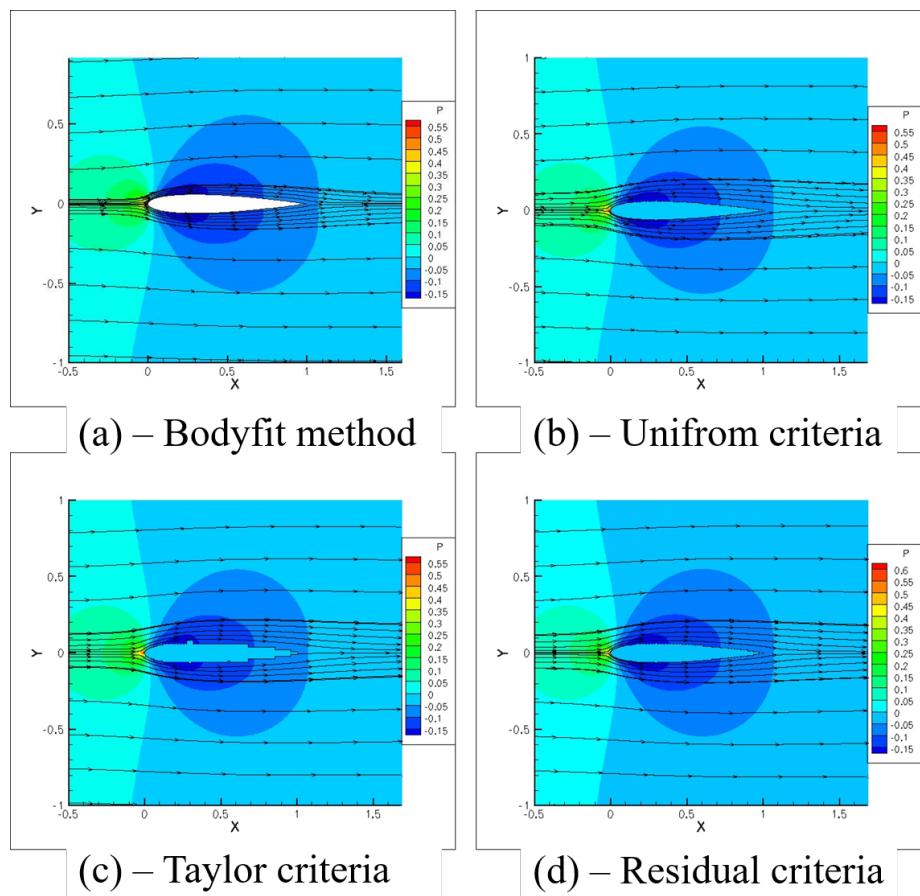


Figure 4.15: Pressure distribution of the four grids at study for $\alpha = 0^\circ$.

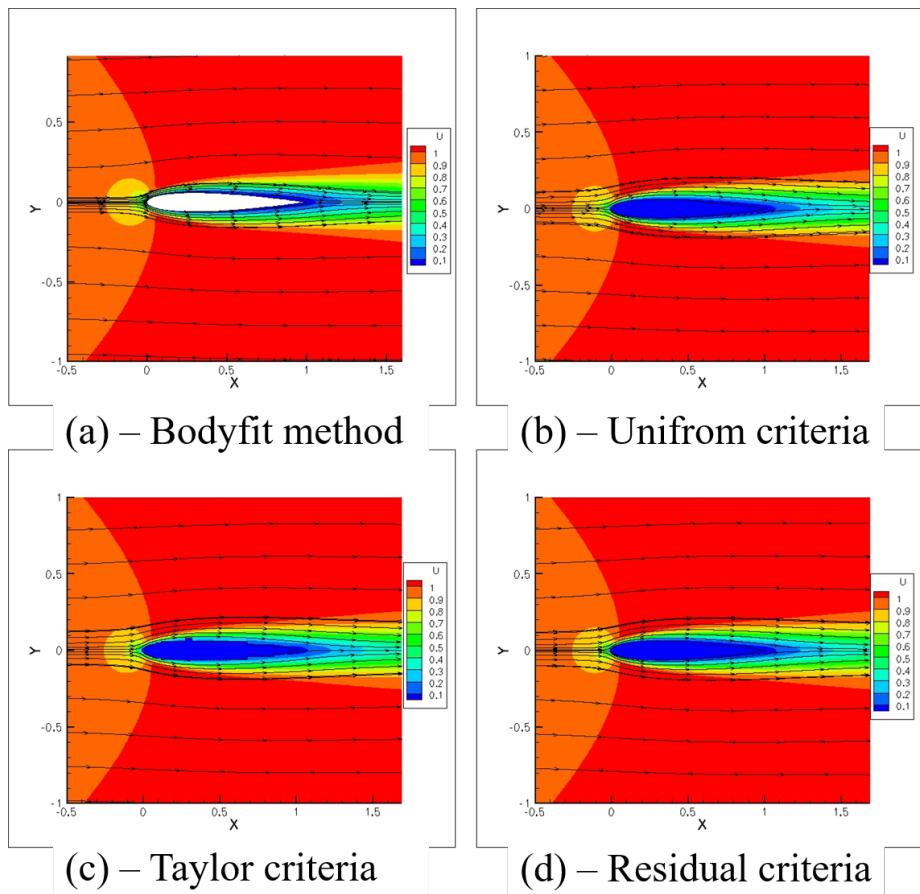


Figure 4.16: Distribution of the horizontal component of velocity for $\alpha = 0^\circ$

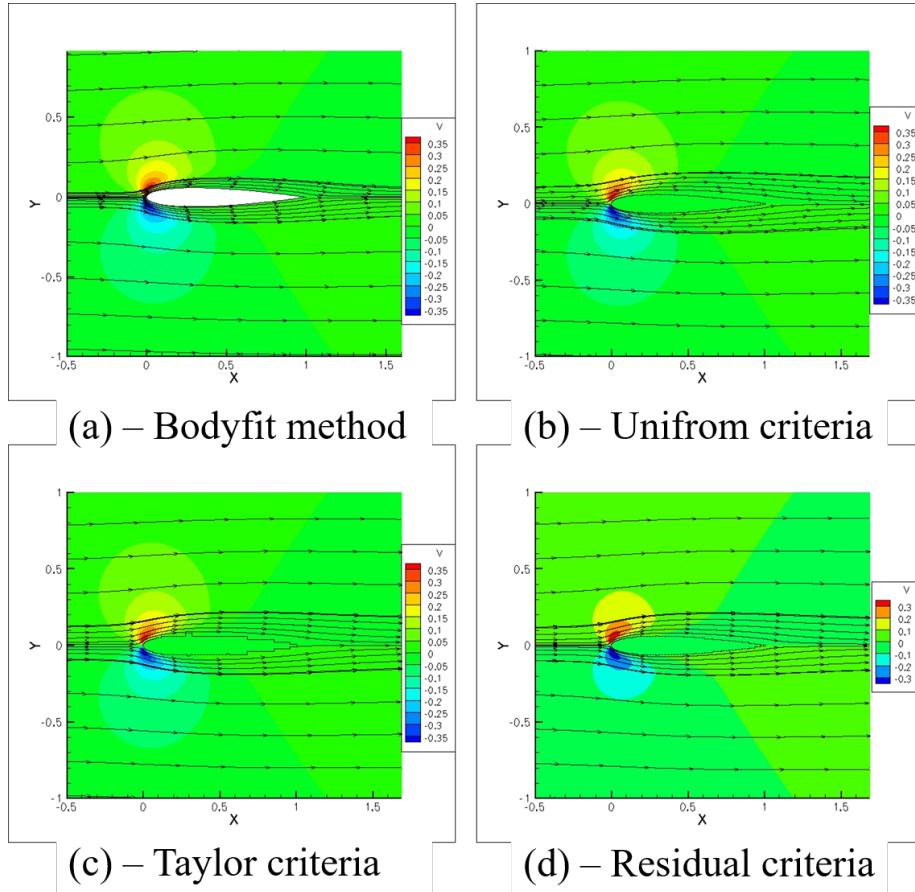


Figure 4.17: Distribution of the vertical component of velocity for $\alpha = 0^\circ$

The results are very similar between the Immersed Boundary method and the Body-fit method. The same happens when comparing the different error estimation criteria. As expected the stagnation point in $x = 0$ is observable and the pressure is higher at this point as it can be seen in Figure 4.15. The horizontal component of the velocity also behaves as expected since it is uniform away from the airfoil, but it decreases where the airfoil curvature is higher since the flow contours the airfoil. The same can be observed in Figure 4.17 since the vertical component of the velocity is higher in module in the zones where the airfoil curvature is more accentuated. All the previous distributions are very similar, including pressure and velocity higher and lower values. Finally, the streamlines also behave as expected since they are symmetric and they do not form any separation zone.

Changing the attack angle to $\alpha = 10^\circ$ increases the complexity of the solution since it creates a separation bubble in the upper surface of the airfoil. For the body-fit case, instead of performing a rotation of the body, the initial boundary conditions were changed to include the attack angle in the inlet boundary conditions. Maintaining the initial grid was simpler than altering the initial grid. These alterations should not impact the flow behaviour but it will impact the values of the velocity components. The most effective path to compare the results between the body-fit method and the IBM is to once again compare the drag and lift coefficients between the different methods. Table 4.8 summarizes the results for $\alpha = 10^\circ$ and once again the results obtained with the different criteria using the IBM are very similar to the results obtained with the body-fit method.

Table 4.8: NACA 0012 results for $\alpha = 10^\circ$.

Grid	Cells	C_d	C_l
Body-fit	93330	0.2079	0.4317
Uniform Refinement Criteria	2068480	0.2041	0.4383
Taylor Adaptive Criteria	77749	0.2178	0.4318
Residual Adaptive Criteria	45202	0.2079	0.4472

The results for $\alpha = 10^\circ$ are now shown in Figures 4.18, 4.19 and 4.20.

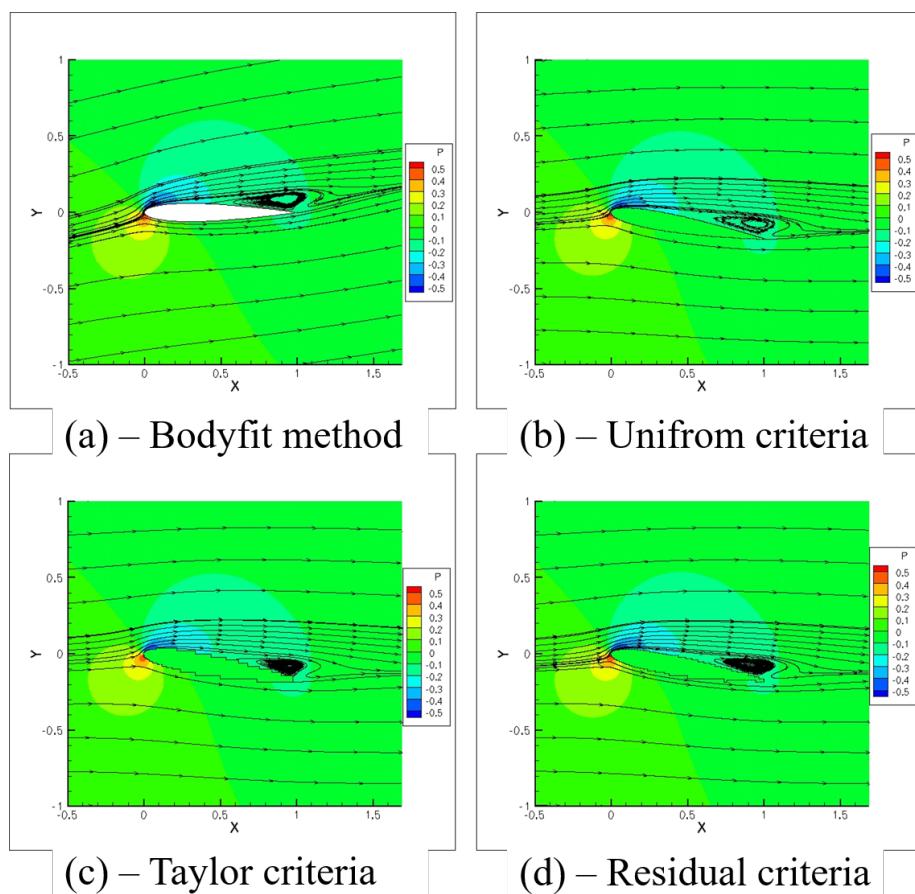


Figure 4.18: Pressure distribution for $\alpha = 10^\circ$.

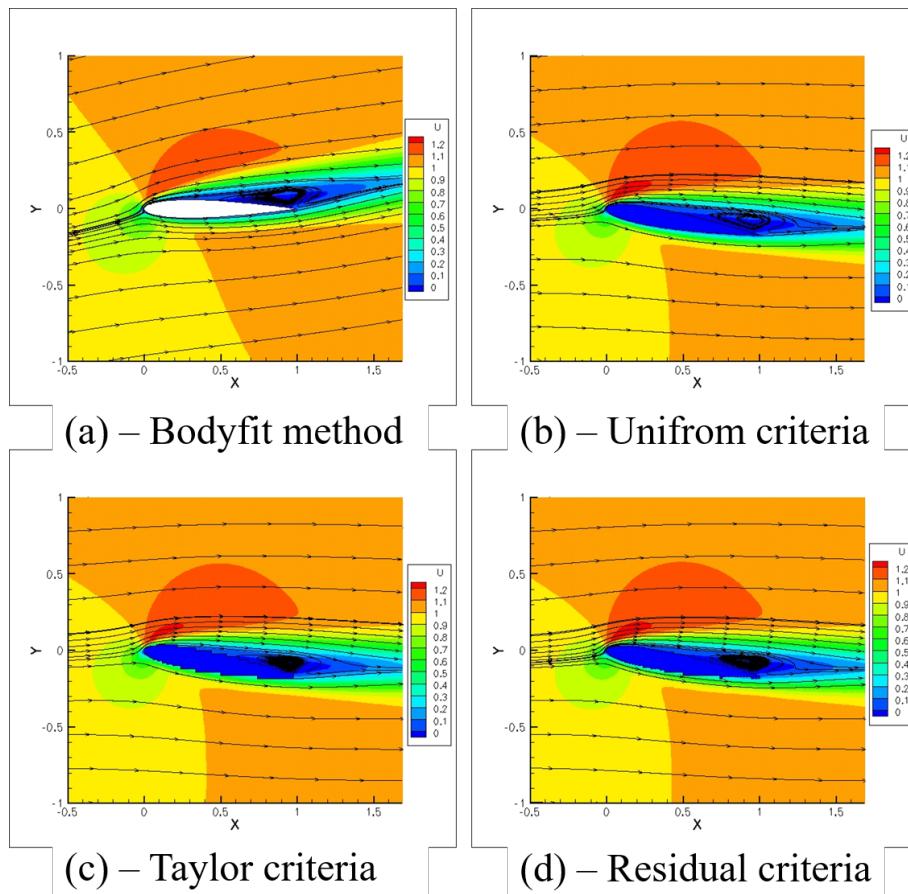


Figure 4.19: Distribution of the horizontal component of velocity for $\alpha = 10^\circ$.

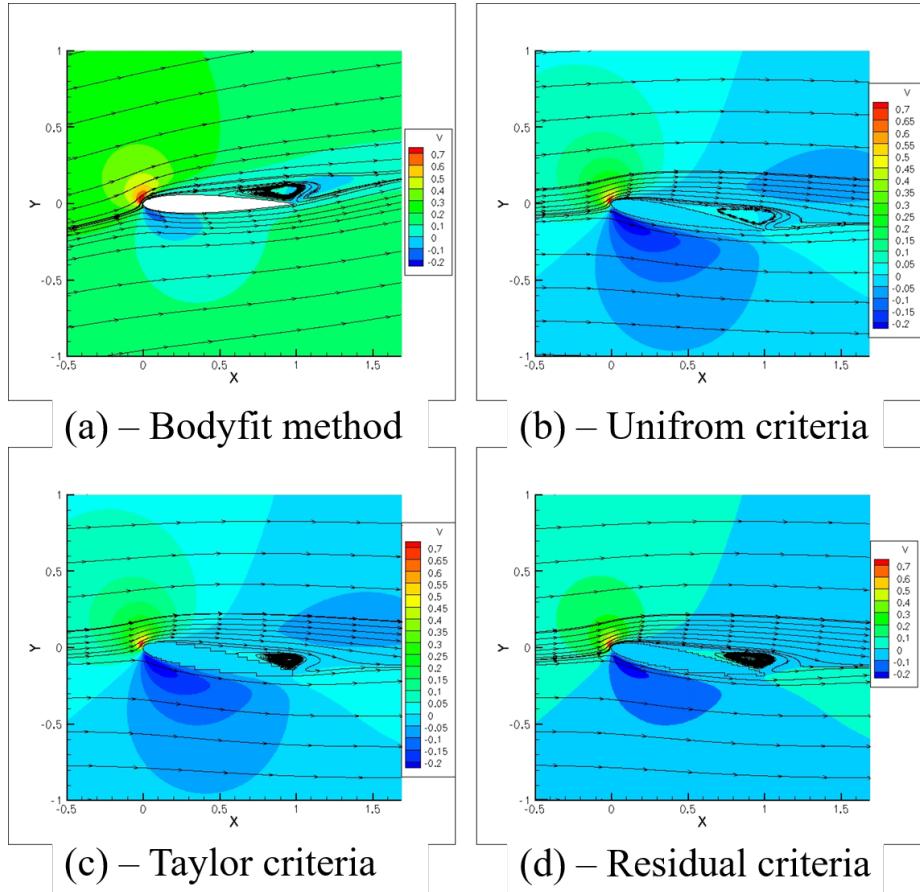


Figure 4.20: Distribution of the vertical component of velocity for $\alpha = 10^\circ$.

Once again, the results are very similar to one another. The location of the stagnation point can be seen in the previous figures and the flow is no longer symmetrical since a separation bubble appeared in the upper surface. The pressure distribution contours and its values are again very similar. The different components of the velocity are now different between the Body-fit method and the IBM but the contours are clearly similar and more important, the separation bubble occurs in the same zone for the different methods. Figure 4.20 shows the vertical component of the velocity is higher in upper side of the airfoil where the curvature is more accentuated.

Despite the increase in complexity and the appearance of a separation bubble, the different criteria maintain a good concordance between the results.

It can also be observed from the previous figures that the Taylor criteria grid refines a smaller area of the initial grid, focusing in the leading edge of the airfoil leaving the remaining of the airfoil with a more coarse shape. On the other hand, the Residual criteria refines a larger area of the domain maintaining a certain smoothness of the grid as seen in Section 4.1.

4.2.1 NACA 0012 - Discussion

The refinement history of the Residual criterion can be observed in Figure 4.21 where five different refinement steps are shown.

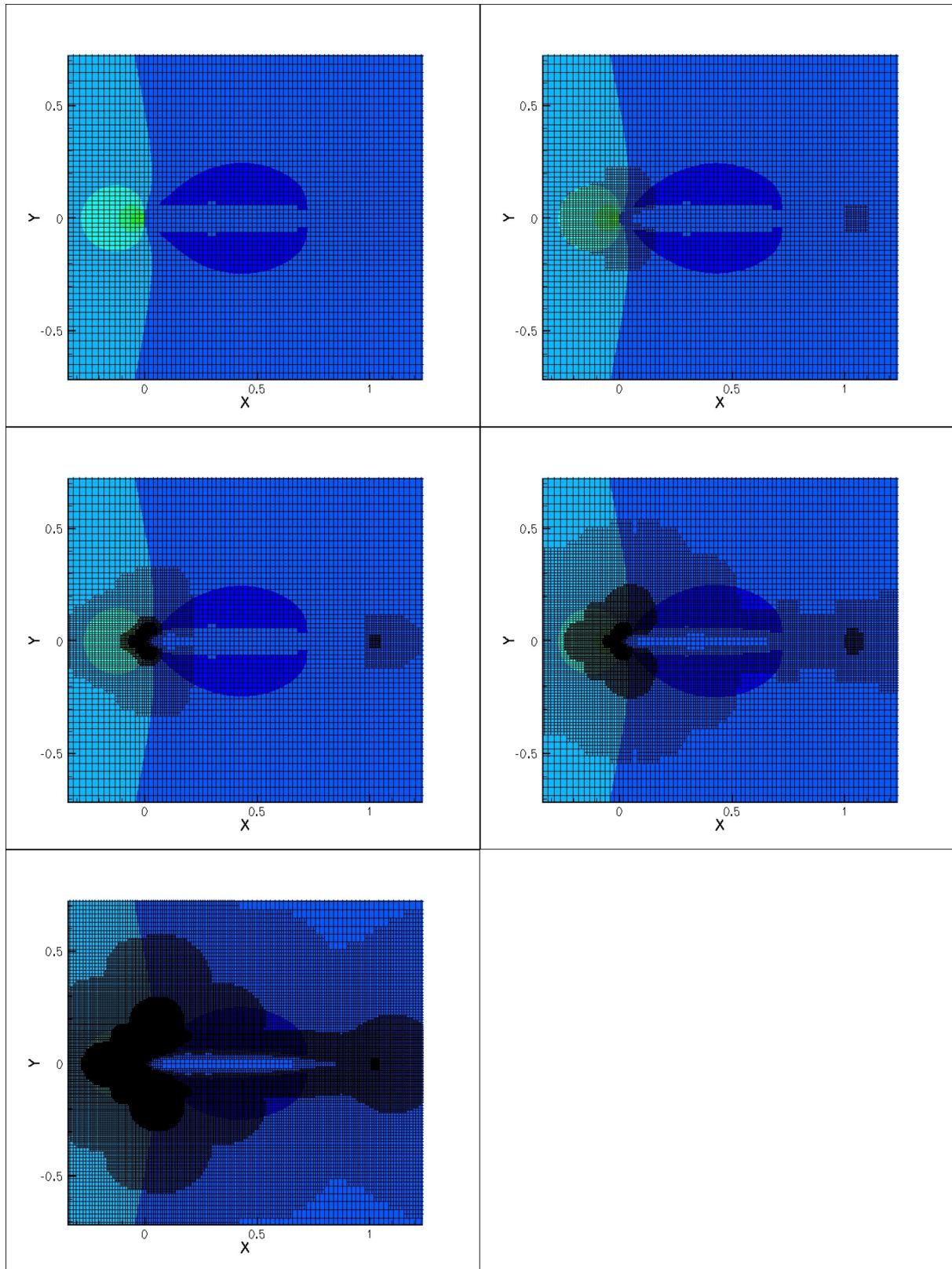


Figure 4.21: Fiva refinement steps of the Residual criterion.

In Figure 4.22 it is possible to conclude that the body-fit method has converged. The results obtained by this method will once again be used for comparison in order to analyse the results obtained by the IBM.

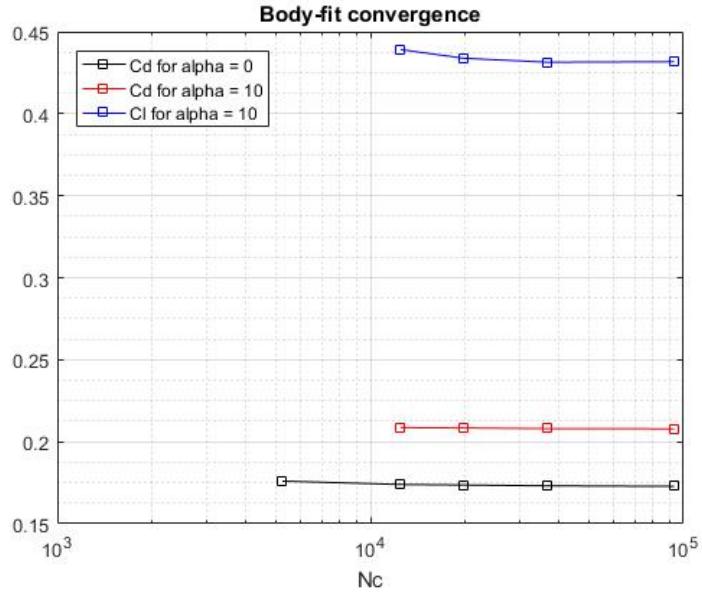


Figure 4.22: Body-fit method convergence for the NACA 0012 airfoil.

Figures 4.23 - 4.25 summarize the results for the IBM with the different criteria used by comparing them to the results from the most refined body-fit grid. The lift coefficient for $\alpha = 0^\circ$ was not compared since it is approximately zero as expected.

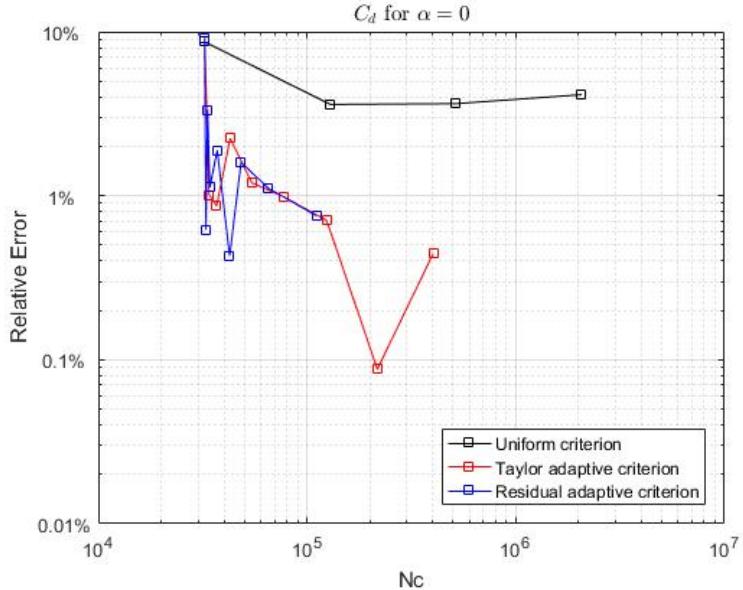


Figure 4.23: C_d error relative to the body-fit method for $\alpha = 0^\circ$.

The uniform criterion has a 2.37% error when compared to the most refined grid of the body-fit method. Both criteria have also failed to converge to a clear result before instabilities in the grid caused the simulation to diverge. Despite this, the results have a relative error lower than 1% when compared to the body-fit method.

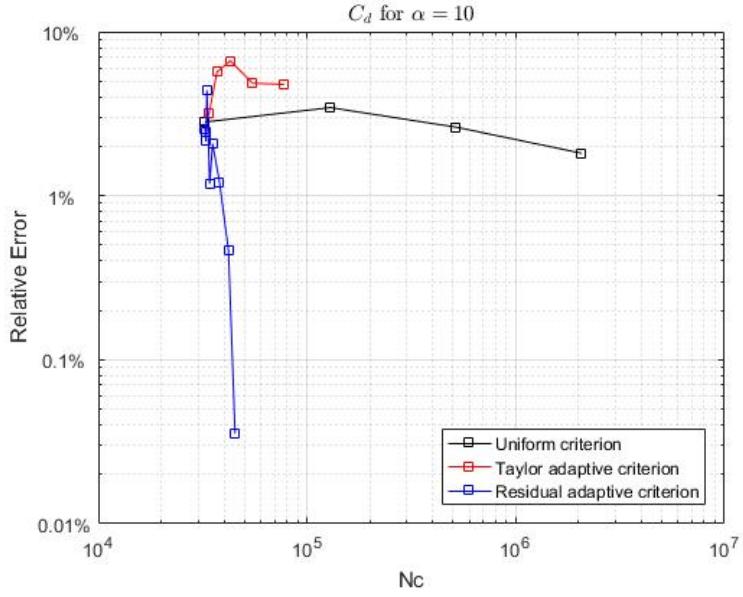


Figure 4.24: C_d error relative to the body-fit method for $\alpha = 10^\circ$.

The uniform criterion of the IBM has failed to converge despite the high number of cells. For this attack angle ($\alpha = 10^\circ$), the Residual adaptive criterion simulation resulted in a 0.04% error when compared to the body-fit method. More refinement levels should be used for a more clear convergence of this method, but the instabilities in the simulation caused it to diverge. Despite the lack of more refinement levels, the error between the two final levels for this criterion is only 0.43%. The Taylor criterion has converged for a 4.77% error.

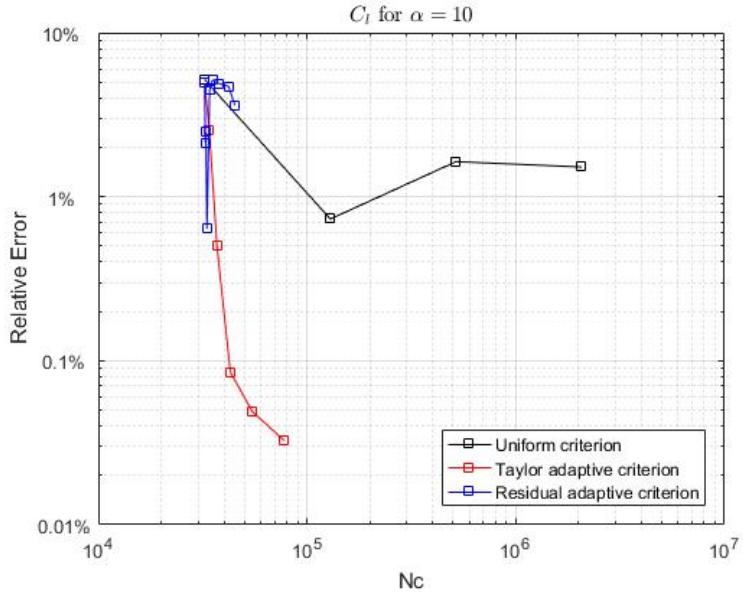


Figure 4.25: C_l error relative to the body-fit method for $\alpha = 10^\circ$.

The Taylor criterion has a small error value of 0.03%. The Residual criterion on the other hand has the largest error of 3.6% but it is still to converge. Once again, more levels of refinement for the Residual criterion would be useful, however the simulation failed to converge after the present results.

A possible justification for the difference in results for $\alpha = 10^\circ$ is that the flow for this angle of attack is unsteady. The literature is clear that for $\alpha = 0^\circ$ the flow is steady, however for $\alpha = 10^\circ$ there is not a consensus and the flow may have become unsteady causing the differences in values between the error criteria and the body-fit method.

Chapter 5

Conclusions

5.1 Achievements

In this Thesis, an extensive study of the IBM results was made with the purpose of verifying and validating the method by comparison with other works. After validating the method, the error estimation criteria could then be applied and the results compared to see the benefits these criteria bring. Implementing these criteria to generate adaptive grids allowed a great reduction in the computational power needed to successfully simulate flows.

The Taylor error estimation was altered in order to improve its results and the stability of the grids it would generate. In this new version of the error estimator, the Hessian matrix values would now be interpolated by the least squares polynomial from the IBM, improving the accuracy of the Hessian matrix and therefore, the accuracy of the error estimator itself.

Several numerical schemes were compared in order to understand how the scheme's choice would impact the results. From this analysis we could conclude that the Face Least Squares numerical schemes were the ones that achieved the best results, generating more stable adaptive grids that would have lower computational error. After concluding the best schemes by comparison in an analytical case, the decision was made to extend the use of these schemes to every other case in this Thesis.

An extensive study of an immersed 2D cylinder case was made for $Re = 10$, $Re = 20$ and $Re = 40$ both with and without the adaptive criteria. This served as an ideal example to show the capabilities of an automated adaptive grid algorithm coupled with the error estimation criteria since it demonstrated how much computational power could be saved.

Finally, the IBM with adaptive refinement was also applied to a NACA 0012 airfoil. This case proved to be more of a challenge to both body-fit and IBM generated grids.

With the studies performed in this Thesis, it is clear the benefits in terms of accuracy and computational power both criteria brought to the IBM. The Residual error estimator proved to be a better criteria, with more efficient adaptive grids and smooth refined areas. The Taylor error estimator on the other hand had some instability problems caused by non-smooth grids and it also over-estimated error in the computational boundary, adding unnecessary cells to the grids, and therefore increasing the computational

power need to solve each simulation.

5.2 Future Work

A further study on the numerical schemes and their improvement would be beneficial to further develop the work done by this Thesis. Improving the grids generated by implementing new steps to the adaptive algorithm that assured a symmetric and smooth grid would also help their stability allowing for more levels of refinement in more complex cases as the ones performed in this Thesis. One step that could be included is the coarsening of the adaptive grid. This extra step would be helpful to correct unnecessary refinements.

A total automated process was not achieved in this work. In order to do that, a new refinement criterion should be implemented to refine the computational domain after a simulation has failed to converged. This should be an adaptive criteria with geometric capabilities since a uniform refinement would waste computational power.

The IBM with the capabilities implemented in this work should be extended to the unsteady form of the incompressible Navier-Stokes Equations to detect vortex shedding in the NACA airfoil at high angles of attack and to implement higher order discretization schemes. Employing these schemes in the IBM would, furthermore, improve the results.

A final suggestion also corresponds to the final objective to extended this work to 3D. Adding turbulent models like RANS (Reynolds Average Navier-Stokes) or LES (Large Eddy Simulation) to this method would also be convenient to study the behaviour of bodies immersed in turbulent flows.

Bibliography

- [1] J. Kim, D. Kim, and H. Choi. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *Journal of Computational Physics*, 171:132–150, 2001.
- [2] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *Journal of Computational Physics*, 156:209–240, 1999.
- [3] R. vander Meulen. *The Immersed Boundary Method for the (2D) Incompressible Navier-Stokes Equations*. PhD thesis, 2006.
- [4] C. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.
- [5] Z. Wang, J. Fan, and K. Cen. Immersed boundary method for the simulation of 2D viscous flow based on vorticity – velocity formulations. *Journal of Computational Physics*, 228(5):1504–1520, 2009.
- [6] L. Sun, S. R. Mathur, and J. Y. Murthy. An Unstructured Finite-Volume Method for Incompressible Flows with Complex Immersed Boundaries. *Numerical Heat Transfer, Part B: Fundamentals*, 58(4):217–241, 2010.
- [7] R. Mittal and G. Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005.
- [8] E. M. Saiki and S. Biringen. Numerical Simulation of a Cylinder in Uniform Flow : Application of a Virtual Boundary Method. *Journal of Computational Physics*, 465(123):450–465, 1994.
- [9] M.-c. Lai and C. S. Peskin. An Immersed Boundary Method with Formal Second-Order Accuracy and Reduced Numerical Viscosity 1. *Journal of Computational Physics*, 160:705–719, 2000.
- [10] R. P. Beyer and R. J. Leveque. Analysis of a one-dimensional model for the Immersed Boundary Method. *SIAM Journal on Numerical Analysis*, 29(2):332–364, 1992.
- [11] D. Goldstein, R. Handler, and L. Sirovich. Modeling a No-slip Flow Boundary with an External Force Field. *Journal of Computational Physics*, 105:354–366, 1993.

- [12] P. G. Tucker and Z. Pan. A Cartesian cut cell method for incompressible viscous flow. *Applied Mathematical Modelling*, 24:591–606, 2000.
- [13] M. J. Berger and M. J. Aftosmis. Aspects and Aspect Ratios of Cartesian Mesh Methods. *Paper to appear in the Lecture notes of the International Conference on Numerical Methods in Fluid Dynamics*, 1998.
- [14] D. M. S. Albuquerque, J. M. C. Pereira, and J. C. F. Pereira. Residual Least-Squares Error Estimate for Unstructured h-Adaptive Meshes. *Numerical Heat Transfer, Part B: Fundamentals*, 67(3):187–210, 2014.
- [15] Albuquerque, Duarte. *Numerical Computation of Incompressible Flows on Adaptive and Unstructured Grids*. PhD thesis, 2013.
- [16] J. Alexandre. A 2-D Immersed Boundary Method on Low Reynolds Moving Body, 2014.
- [17] D. Matias and C. Martins. On the suppression of spurious pressure oscillations in immersed boundary methods with unstructured grids. (November), 2016.
- [18] L. Sun, S. R. Mathur, and J. Y. Murthy. An Unstructured Finite-Volume Method for Incompressible Flows with Complex Immersed Boundaries. *Numerical Heat Transfer, Part B: Fundamentals*, 58(4):217–241, 2010.
- [19] J. Magalhães, S. Pereira, S. José, and C. Fernandes. *An adaptive framework for the numerical simulation of environmental flows in complex geometries*. PhD thesis, 2011.
- [20] J. Park, K. Kwon, and H. Choi. Numerical solutions of flow past a circular cylinder at Reynolds numbers up to 160. *KSME International Journal*, 12(6):1200–1205, 1998.
- [21] A. Frisani and Y. A. Hassan. On the immersed boundary method: Finite element versus finite volume approach. *Computers & Fluids*, 121:51–67, 2015.
- [22] H. Ding, C. Shu, K. Yeo, and D. Xu. Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method. *Computer Methods in Applied Mechanics and Engineering*, 193(9-11):727–744, 2004.
- [23] S. Sen, S. Mittal, and G. Biswas. Steady separated flow past a circular cylinder at low Reynolds numbers. *Journal of Fluid Mechanics*, 620:89, 2009.
- [24] A. L. F. L. E. Silva. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics*, 189:351–370, 2003.
- [25] X. He and G. Doolen. Lattice Boltzmann Method on Curvilinear Coordinates System: Flow around a Circular Cylinder. *Journal of Computational Physics*, 134:306–315, 1997.
- [26] M. N. Linnick and H. F. Fasel. A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. *Journal of Computational Physics*, 204:157–192, 2005.

- [27] B. Fornberg. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics*, 98:819–855, 1980.
- [28] D. Calhoun. A Cartesian Grid Method for Solving the Equations in Irregular Regions. *Journal of Computational Physics*, 176:231–275, 2002.
- [29] X. D. Niu, C. Shu, Y. T. Chew, and Y. Peng. A momentum exchange-based immersed boundary-lattice Boltzmann method for simulating incompressible viscous flows. *Physics Letters A*, 354:173–182, 2006.
- [30] S. Xu and Z. J. Wang. An immersed interface method for simulating the interaction of a fluid with moving boundaries. *Journal of Computational Physics*, 2006.
- [31] J.-i. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati. An immersed boundary method for complex incompressible flows. *Journal of Computational Physics*, 224:757–784, 2007.
- [32] R. E. Peck, J. R. Pacheco, and T. Rodic. Numerical simulations of heat transfer and fluid flow problems using an Immersed-Boundary Finite-Volume method on nonstaggered grids. *Numerical Heat Transfer, Part B: Fundamentals*, 47:1–24, 2005.
- [33] Y. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192:593–623, 2003.
- [34] R. Mittal, H. Dong, M. Bozkurttas, and F. M. Najjar. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227:4825–4852, 2008.
- [35] P. H. Chiu, R. K. Lin, and T. W. H. Sheu. A differentially interpolated direct forcing immersed boundary method for predicting incompressible Navier – Stokes equations in time-varying complex geometries. *Journal of Computational Physics*, 229(12):4476–4500, 2010.
- [36] P. H. Chiu, T. W. H. Sheu, and R. K. Lin. An effective explicit pressure gradient scheme implemented in the two-level non-staggered grids for incompressible Navier – Stokes equations. *Journal of Computational Physics*, 227:4018–4037, 2008.
- [37] M. Hafez. Improved numerical simulations of incompressible flows based on viscous / inviscid interaction procedures. *Computers & Fluids*, 36:1588–1591, 2007.
- [38] K. Hejranfar and E. Ezzatneshan. Implementation of a high-order compact finite-difference lattice Boltzmann method in generalized curvilinear coordinates. *Journal of Computational Physics*, 267:28–49, 2014.
- [39] M. Hafez, A. Shatalov, and E. Wahba. Numerical simulations of incompressible aerodynamic flows using viscous / inviscid interaction procedures. *Computer Methods in Applied Mechanics and Engineering*, 195:3110–3127, 2006.
- [40] T. Imamura and K. Suzuki. Flow Simulation Around an Airfoil by Lattice Boltzmann Method on Generalized Coordinates. *AIAA Journal*, 43(9):1–6, 2005.

Appendix A

Literature Table

A more detailed version of the Tables that compared the C_D values in the literature and in the present work in Section 3.1.

Paper	Year	Characteristics	Boundaries	I/d 10	I/d 20	I/d 40	Cd Re10	Cd Re20	Cd Re40	notes
Lima E Silva et al.	2003	Impulsive diffusive direct forcing, NSEs	Neumann for the lateral	0.26	1.04	2.55	2.81	2.04	1.54	-
Ye et al.	1999	Cut cell method, NSE	Neumann for the exit	-	0.92	2.27	-	2.03	1.52	-
Linnick and Fasel	2005	IIM, streamfunction-vorticity	-	-	0.93	2.23	-	2.160	1.610	B=0.056
Linnick and Fasel	2005	IIM, streamfunction-vorticity	-	-	0.93	2.23	-	2.06	1.54	B=0.023
Fornberg	1980	Numerical solution	Free stream condition $\psi=0$ out	-	0.91	2.24	-	2.00	1.5	-
Calhoun	2002	Interior sharp direct-forcing, streamfunction-vorticity	Lateral: Far field velocity Out: $\psi=0$	-	0.91	2.18	-	2.19	1.62	-
Niu et al.	2006	Explicit sharp direct-forcing, LBE	-	-	0.95	2.26	-	2.144	1.589	B=0.025
He and Doolen	1997	Implicit diffusive direct-forcing, LBE	Far field velocity	0.24	0.92	2.25	3.170	2.152	1.499	B=0.018
Wang et al.	2009	Implicit diffusive direct-forcing, NSEs	Outflow: non-reflecting boundary Lateral: Neumann	-	0.98	2.35	-	2.250	1.660	B=0.067
Xu and Wang	2006	IIM, NSEs	Lateral: Neumann Out: $du/dx=dv/dx=0$	-	0.92	2.21	-	2.230	1.660	B=0.0625
Choi et al	2007	Interior sharp direct-forcing, NSEs	Dirichlet boundary condition: $u/u_{inf}=1$	-	0.90	2.25	-	2.020	1.520	B=0.0125

Pacheco et al.	2005	Exterior sharp direct-forcing, NSEs	Inflow: Dirichlet Out: convective Lateral: $du/dx=0$	-	0.91	2.28	-	2.080	1.530	B=0.033
Tseng and Ferziger	2003	Interior sharp direct-forcing, NSEs	-	-	-	2.21	-	-	1.530	B=0.0625
Mittal et al.	2008	Exterior sharp direct-forcing, NSEs	-	-	-	-	-	-	1.530	B=0.025
Kim et al.	2001	Exterior sharp direct-forcing, NSEs	Inflow+lateral: Dirichlet Outflow: Convective	-	-	-	-	-	1.510	B=0.01
Frisani and Hassan	2012	Implicit diffusive direct-forcing, NSEs	Inlet+lateral: Dirichlet Outlet: Convective(homogeneous Neumann)	-	0.90	2.40	-	2.220	1.661	B=0.05 IFEM, IB surface mesh, bi- linear
Frisani and Hassan	2012	Implicit diffusive direct-forcing, NSEs	Inlet+lateral: Dirichlet Outlet: convective(homogeneous Neumann)	-	0.93	2.35	-	2.229	1.656	B=0.05 IFEM, IB surface mesh, bi- quadratic
Frisani and Hassan	2012	Implicit diffusive direct-forcing, NSEs	Inlet+lateral: Dirichlet Outlet:convective(homogeneous Neumann)	-	0.85	2.09	-	2.168	1.615	B=0.05 IFEM, IB volume mesh, bi- linear
Frisani and Hassan	2012	Explicit sharp direct-forcing, NSEs	Inlet+lateral: Dirichlet Outlet:convective(homogeneous Neumann)	-	0.92	2.29	-	2.124	1.587	B=0.05 IFVM, bi- linear

