UNIVERSIDADE FEDERAL DO PAMPA

Sérgio Rossi Brito da Silva

Numerical Grid Generator in Generalized Coordinates

Alegrete
2022

Sérgio Rossi Brito da Silva

Numerical Grid Generator in Generalized Coordinates

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Mecânica da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia Mecânica.

Supervisor: Prof. Dr. Cesar Flaubiano da Cruz Cristaldo

Alegrete
2022

Ficha catalográfica elaborada automaticamente com os dados fornecidos
pelo(a) autor(a) através do Módulo de Biblioteca do
Sistema GURI (Gestão Unificada de Recursos Institucionais) .

d111n da Silva, Sérgio Rossi Brito
Numerical Grig Generator for Generalised Coordinates / Sérgio Rossi Brito da Silva.
47 p.

Trabalho de Conclusão de Curso(Graduação)-- Universidade Federal do Pampa, ENGENHARIA MECÂNICA, 2022.
"Orientação: César Faubiano da Cruz Cristaldo".

1. Malha numérica. 2. Fluidodinamica computacional. 3. Geometrias complexas. I. Título.

SERGIO ROSSI BRITO DA SILVA

NUMERICAL GRID GENERATOR IN GENERALIZED COORDINATES

Monografia apresentada ao Curso de Engenharia Mecânica da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia Mecânica.

Dissertação defendida e aprovada em: 11, Agosto de 2022.

Banca examinadora:

_____

Prof. Dr. Cesar Flaubiano da Cruz Cristaldo

Orientador

Unipampa

_____

Prof. Dr. Adriano Roberto Carotenuto

Unipampa

_____

Prof. Dr. Rafael Maroneze

Unipampa

Assinado eletronicamente por **RAFAEL MARONEZE**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 16/08/2022, às 15:04, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **ADRIANO ROBERTO DA SILVA CAROTENUTO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 16/08/2022, às 15:19, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Assinado eletronicamente por **CESAR FLAUBIANO DA CRUZ CRISTALDO**, **PROFESSOR DO MAGISTERIO SUPERIOR**, em 16/08/2022, às 15:42, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0898322** e o código CRC **AD6AF110**.

Acknowledgements

I would like to thank my family, for support me to keep going, and thank my teachers for the knowledge I have acquired and for guiding me. Mainly, I would like to thank all of my friends who helped me in this journey, and thank all the places I've lived in this period, where by that, I mean the people who lived there and lived with me. I hope to continue seeing each one of them in the future.

Thank you *Navio pirada* where I spent my first months, they guided me in my first steps of this journey and, even though I didn't live there, they made me feel at home.

Thank you *Rep 202*. I have spent the most part of my graduation there, where I met the majority of my friends and where I developed the most. A special thanks for Lauri, who I've spent most of my time and helped me tho loosen up. Chico's advices helped and continue to help me in my academical and personal plans. Piaui, whose presence impacts everywhere and everyone, didn't fail to impact my life too with great joy. Ana, Andressa, Bruna, Rafa, my dear friends who put up with me all the time, and helped me with all they could. The list of people whom I met there is too big to quote here, but I'd like to say that almost everyone made me the man I am today.

Thanks to *Condomínio do Chaves*, I spent most of my time there in 2019 and, even with all the troubles that happened that year, thanks to Aurélio and Bianca it weren't a total lost

Thanks to *Luz vermelha*, I didn't lived there, but the time I have spent with them were full of parties and fun. A special thanks to Keshi, great friend who studied with me during the majority of the course, helped and carried me in a lot of subjects. Filipin and DK, Great friends whom I've met in the beginning of the course, and continued with me until the end. Raunir, Jet and Goiano, freshman from 2019, thanks for all the fun that you have given me.

And thanks to *Castelo Ratimbum* (where I consider to have lived since 2020). Thanks to Lari who accompanied me until the end, after the end of *Rep 202*, and, together with Leo, made me company in the pandemic situation of COVID-19, put up with me, carrying me and being great friends. Vitor, who studied with me a lot in the ERES. Victor, Weila, Adrielly and Kawa, I didn't spend too much time with they, but they are great people who I'd like to know more. To Guilherme, whom I spent most of my final semester with, and even thought it wasn't long, he became a great friend of mine, he taught me a lot of things I had already forgotten.

All the people I didn't quoted here still made a big impact on me, on my life and on my academic course, it would be just to big to quote everyone. But all the property tenants, do not follow the same rule.

A friend that I haven't mentioned yet is Lucas. He gave me support in all of it, accompanied me from the start until the very end of the course. Lucas, without you, I wouldn't have arrived where I am today...

Resumo

A complexidade dos problemas fluido-dinâmicos fazem com que uma solução analítica não seja possível, além disso, uma solução experimental pode não ser plausível em diversos contextos. O avanço da tecnologia fez com que os métodos numéricos surgissem como uma alternativa para esses problemas. Um dos parâmetros que impactam na solução desse método é a malha numérica utilizada, sendo esta, ainda influenciada pela geometria do problema. Em sistemas de coordenadas comuns, para geometrias complexas as malhas numéricas podem gerar uma perda de acurácia na solução do problema, para isso, este trabalho examina a possibilidade da implementação de um gerador de malhas utilizando equações diferenciais parciais elípticas para geração de malhas que se adequadam às fronteiras, mesmo estas apresentando geometrias complexas. Após desenvolvermos e testarmos nosso gerador com diferentes fronteiras, avaliamos os resultados, comparando sua convergencia em função do refinamento da malha, realizado através do aumento de curvas em direção de η. Admitindo um resíduo de 10^{-6}, resultamos, para 5 camadas, um numero médio de 65 iterações para convergir, enquanto para 25 foram necessárias em média 3487 iterações.

Palavras-chave: malha numérica, fluidodinâmica computacional, geometrias complexas

Abstract

The complexity of the fluid-dynamic problems make an analytical solution not possible in addition, an experimental solution may not be plausible in many contexts. The advancement of technology made numerical methods emerge as an alternative to these situations. One of the parameters that impact the solution of this method is the numerical grid used, which is still influenced by the geometry of the problem. In common coordinates systems, for complex geometries, numerical grids can generate a loss of accuracy in the solution of the problem. This work examines the possibility of implementing a mesh generator using elliptic partial differential equations to generate grids that fit the boundaries, even though they present complex geometries. After we developed and tested our generator with different boundaries, we evaluated the results, comparing their convergence as a function of mesh refinement made at layers in η direction. Accepting an error of 10^{-6}, resulted, for 5 layers a mean of 65 iterations to converge, while for 25 layers were needed a mean of3487 iterations

Key-words: numerical grid, computational fluid dynamics, complex geometries

List of Figures

Lista de siglas

Eu Euler number

Fr Froude number

Ma Mach number

Re Reynolds number

CFD Computational Fluid Dynamics

CPU Central Process Unit

CS Control Surface

CV Control Volume

FDM Finite Difference Method

FEM Finite Element Method

FVM Finite Volume Method

PDE Partial Differential Equation

RTT Reynolds Transport Theorem

Contents

1 Introduction

The complexity of the conservative equations in fluid dynamics problems means that few solutions can be found by the analytical approach alone, requiring several simplifications that can not exist in the real world. The experimental approach comes as an alternative; however, it can be expensive and sometimes impractical (MALISKA, 2017).

With the advance of technology, a new opportunity becomes possible due the current computers. Even its formulations coming from centuries ago (ANDERSON; TANNEHILL; PLETCHER, 2016), the application of numerical techniques become a reality to solve complex problems (MALISKA, 2017). Giving a few examples, Zante et al. (2014) cite that designs based on these new techniques can produce efficient open rotors, and Johnson, Tinoco e Yu (2005) shows 30 years of those techniques applied and developed by Boeing Commercial Airplanes in Seattle.

The wide acceptance of numerical methods makes Computational Fluid Dynamics (CFD), an emerging field able to resolve complex fluid dynamic problems quickly and in a cheaper way. Still, that doesn't mean a solution in a few seconds for every situation, due to Central Process Unit (CPU) limitations (ANDERSON; TANNEHILL; PLETCHER, 2016), the performance of new computers show a trend in using this approach to study the phenomena related to fluid motion.

The success of this field comes from the high versatility and relative simplicity in its applications. Even without the same accuracy, the CFD doesn't face the same limitations as the analytical method, covering a wide range of cases, and is also a cheaper alternative to laboratory experiments (MALISKA, 2017).

According to Lomax, Pulliam e Zingg (2013), to obtain a satisfactory solution using CFD, we generally follow the following steps, independently of the application under study: Problem Specification and Geometry Preparation, Selection of Governing Equations and Boundary Conditions, Selection of Gridding Strategy and Numerical Method, and Assessment and Interpretation of Results. Our interest in this work falls in the third step when the generation of grids occurs.

A numerical grid, exemplified in Figure 1, is a representation of the physical space in the computational domain where we calculate the solution of the simulation, being able to improve or poorly the accuracy of the results, which explains the importance to use appropriate grids when modeling the problem (ANDERSON; TANNEHILL; PLETCHER, 2016).

We can improve the grid performance by increasing the number of points that compose it; however, it demands more CPU power to solve the problem, explaining the importance of working with optimum grids, made in a manner to improve the results and lower the processing used.

A common problem emerges when the physical domain has complex geometries (FLETCHER, 2012), as happens in many situations like heat exchangers, airplanes, engine

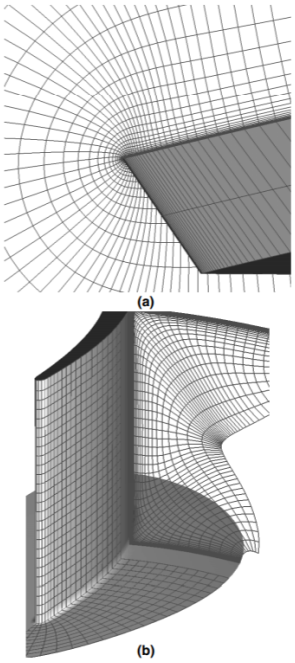


Figure 1 – Example of numerical grid (CHAN et al., 2002)

intakes, and others where the boundaries of those objects do not coincide with usual coordinates systems properly. Figure 2 gives an example of that problem showing a case where the use of a cartesian grid to a circular object does not follow the shape of the object, leading to the inaccuracy of the solution.

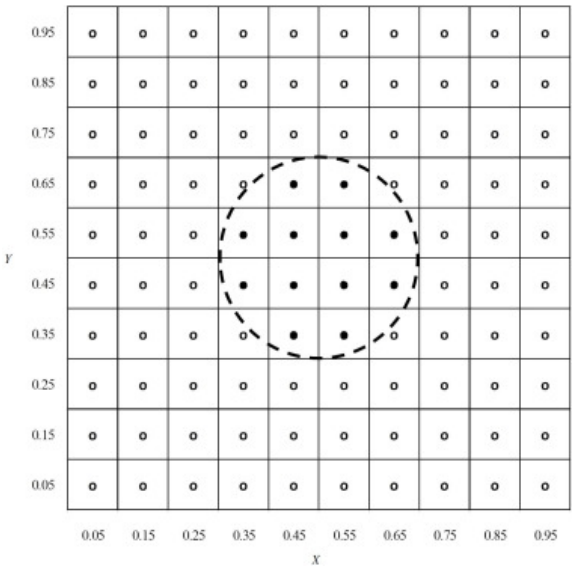


Figure 2 – Non-Boundary-Fitted grid (ZAKARIA et al., 2019)

A favorable alternative to deal with that problem is to use grids in generalized coordinates (MALISKA, 2017), which makes it able to follow complex shapes. The grid in Figure 3, based on Thompson's elliptic grid generation in a three-dimensional case

(THOMPSON, 1982), exemplifies an application of this coordinate system and was extensively tested and used in the Boeing transonic flow analysis system, the A488 (JOHNSON; TINOCO; YU, 2005).

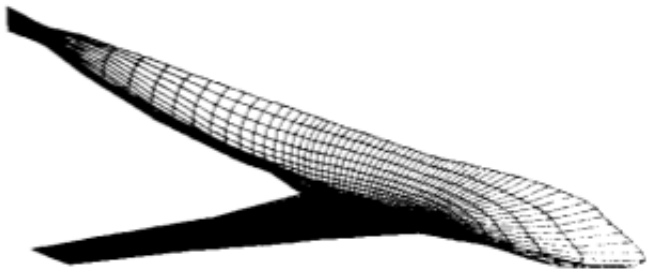


Figure 3 – Grid for a low wing transport configuration using generalised coordinates (YU, 1980)

1.1 Objectives and justification of this work

As the accuracy of the solutions in the CFD is of great importance, the main objective of this work is to suggest the implementation of a numerical grid generator to create a grid able to follow complex geometries. To achieve that, we trace the following specific objectives:

- Development of a program to create a numerical grid at generalized coordinates using elliptical Partial Differential Equation (PDE);
- Test of the program on different shapes of outer and inner boundaries;
- Evaluation of the convergence of the grid in function of the refinement;

As the use of computer processing is a trend in the current scenario, the justification of this work comes from the understanding numerical grid in the numerical approaches exploring the techniques using elliptic PDEs to create the grid, comparing its cost-benefit related with a high refinement on different boundaries shapes.

1.2 Outline

In the next chapter, we explain the background and the related works necessary to understand the rest of this work. Then we provide a mathematical formulation chapter, containing the equations to transform the physical domain in generalised coordinates, the conservative equations used in CFD and its transformation in generalized coordinates. The following chapter shows the methodology used in this work. Then we present the results of our generator and finish this work with our remarks.

2 Background and Related Works

This chapter covers the relevant background and related works necessary to understand the rest of this text. We start explaining the concept of CFD, bringing the difference between it and the traditional methods used when solving a problem. Then, we delve into the numerical grids used in the simulations, presenting the classification and the influence of some grid features. We pass to explain how generalized coordinates works, showing advantages in their use, and bringing examples to generate numerical grids. Finally, the chapter is finished by giving a brief explanation on the numerical method used to discretize the equations that governs the problems.

2.1 Computational Fluid Dynamics

The CFD is a science that makes predictions about the fluid-flow phenomena based on the conservation equations that govern fluid motion (KUNDU; COHEN, 2002). This approach applies some numerical method, such as the Finite Difference Method (FDM), Finite Volume Method (FVM), or the Finite Element Method (FEM), to discretize the equations and solve the PDEs that governs the problem through computing power, solving those equations in a computational grid. The CFD is a powerful tool in the work of engineers, being a new alternative to solve fluid dynamics problems rather than the theoretical and the experimental approaches.

In the comparisons made by Maliska (2017) and Anderson, Tannehill e Pletcher (2016), they came to the conclusions: while the theoretical approach has the advantage of its precision and provides general information of the situation, it is limited to simple problems and geometries; the experimental method gives the most realistic results but is expensive and sometimes impractical; the CFD, in contrast, can solve complex problems in complex geometries and it is cheaper compared to the cost of the experimental approach, yet, can be limited by the computational power and has the influence of numerical errors.

The advance of technology and improvement of numerical techniques have minimized the numerical approach problems and made CFD play the role of technology enablers (MOUKALLED; MANGANI; DARWISH, 2015). Either way, the CFD is not a replacement for either of the traditional methods but should be used to complement each other. While the analytical method still provides exact solutions and simplification, decreasing the computational effort, the experimental approach plays a fundamental role in validating and delimiting the limits of the approximations in the governing equations (FLETCHER, 1991).

The numerical errors present in CFD are, by definition, the difference between the actual value and the calculated result (SCHNEIDER; MARCHI, 2005). Kundu e Cohen (2002) cite four sources of errors: the truncation error, caused by the limitation of the computational domain, errors in the input data, errors in the initial and boundary conditions, created by approximations of properties, and modeling errors, caused by some

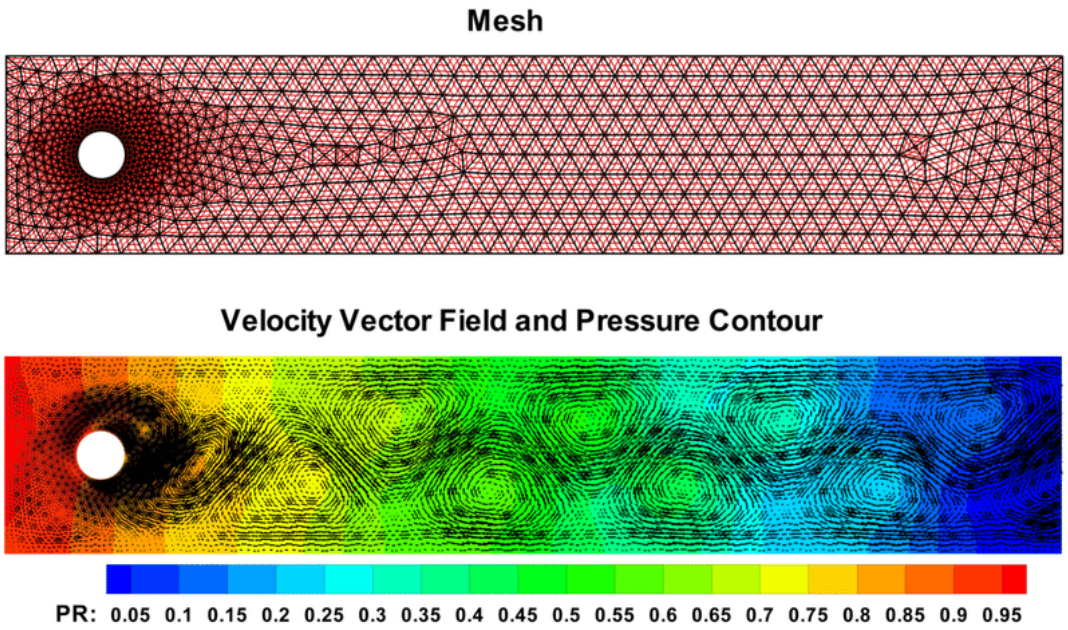


Figure 4 – Simulation of a flow above a circular region (WANG; QUAINI; CANIC, 2018)

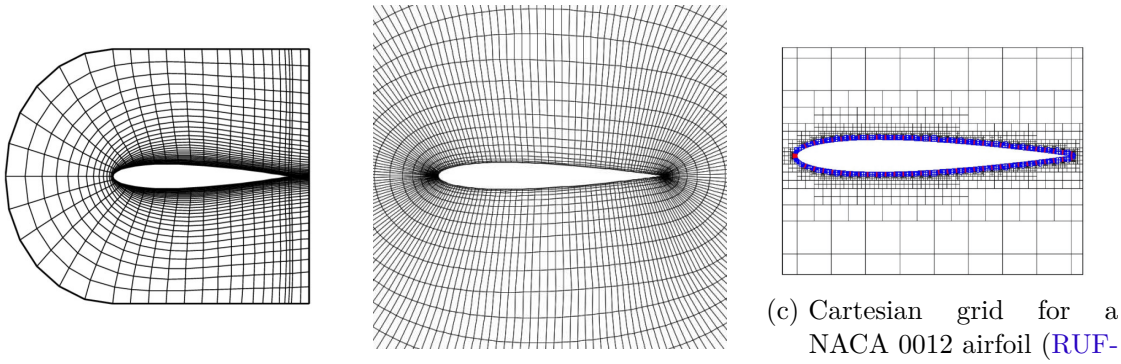
mistake when modeling the problem. Maliska (2017) divides those errors into two levels, the numerical, caused when solving the differential equations that must be compared with other known solutions to validate, and errors using equations that don't represent the problem.

Getting realistic results needs the application of a validated CFD model. We can validate the model by analyzing the numerical results with the analytical solution or another already validated numerical model, known as numerical validation, or comparing the numerical results with experimental ones called physics validation (MALISKA, 2017).

As the computational power and the numerical errors limit the CFD, the search for different ways to reduce these limitations is necessary (MARCHI et al., 2013). The solution of the problems usually involves a search for an optimum numerical grid aiming to get a smooth grid (NOACK; ANDERSON, 1990) with more concentration of points in the regions of interest. Figure 4 shows an example of a numerical grid at the top of the image, used in a 2D simulation of a flow passing by a circular object resulting in the velocity vector field and the pressure contour at the bottom of the image. In the next section, we explain the features of a grid, explaining the difference between types of grids and their effects.

2.2 Numerical grids

Numerical grids, also known as computational grids, are sets of points in the computational domain where the simulation runs. Since the physical space is a continuum medium, applying a CFD model requires its discretization to fit in the computational



(a) C-grid for a NACA 0012 airfoil (SILVA; BARATA, 2011)

(b) O-grid for a NACA 0012 airfoil (DJEDDI; EKICI, 2015)

(c) Cartesian grid for a NACA 0012 airfoil (RUFFIN; ZAKI; SEKHAR, 2012)

Figure 5 – Differents grids applied to the same region

domain for that reason, several approaches in grids generation have been explored. As examples of works that explore grid generators: Noack e Anderson (1990) present a solution-adaptive procedure to generate a parabolic grid-generator scheme; Hiester et al. (2014) evaluate the impact of adaptive meshes on two-dimensional lock-exchange flow; Silva e Barata (2011) present a parallel numerical method for generating an orthogonal grid generator with boundary point distribution. The choice and generation of the numerical grid are one of the first steps to do as explained by Anderson, Tannehill e Pletcher (2016) and Steinberg (1993) justifies that the importance of that step arose from the necessity to compute the solutions in complex geometries.

Since we calculate the results there, the space between the grid points influences the accuracy of the solution. It results in a decrease of the truncation error with the decrement of this spacing. In Figure 5, different grids were applied to the same airfoil and, in both cases, they have a higher concentration of points closer to the airfoil where the most relevant effects happen and where we desire a higher accuracy. That is a common practice since, as explained by Noack e Anderson (1990), the search for an optimum grid usually seeks to minimize the truncation errors. This increase in the concentration of points it's called refinement and usually leads to better solutions to the simulation in exchange to increase the use of computational power (MARCHI et al., 2013) so it generally occurs where the flow properties gradients happen.

Figure 5 also shows examples of different types of grids. Figures 5a and 5b bring grids that use generalized coordinates, which permit them to follow the domain boundaries. Figure 5c use a cartesian coordinate system what according to Ruffin, Zaki e Sekhar (2012), has the advantage that the entire process of grid generation and flow solution is easily automated, however, that also leads to uncertainty because the grid cannot accompany the shape of the airfoil making necessary to refine the grid a lot to obtains a satisfactory solution, to minimize this problem they apply a normal ray refined technique

to allow the use of the cartesian grid efficiently.

The classification of grid types can happen in different ways. Figures 5a and 5b, show boundary-fitted grids, presenting the ability to adapt themselves to the domain borders. Those grids have the advantage of accuracy in the results close to the boundaries and can represent the boundary conditions properly. The disadvantage is difficult to generate it, consuming a large amount of computing time (THOMPSON; WARSI; MASTIN, 1985).

Another classification refers to structured and unstructured grids, represented in Figure 6. The structured grids have their points, or cells in the FVM case, making a uniform pattern resulting in a well-ordered set which brings several advantages in implementing the computational problem due to this ordination. The unstructured grid is more versatile, being easier to adapt to the domain, however, it also leads to a more complex implementation (MALISKA, 2017).

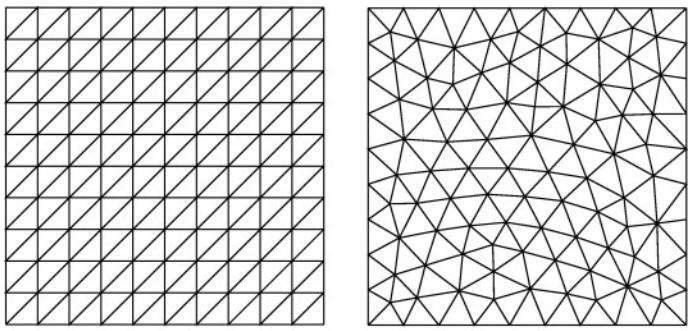


Figure 6 – Difference between structured and unstructured grids (HIESTER et al., 2014)

The creation of numerical grids can occur in different manners. Anderson, Tannehill e Pletcher (2016) describe some methods that generate structured grids: Complex Variable methods, Algebraic methods, and Differential Equations techniques. The Complex Variable methods have the advantage that the transformation is, at least partial, analytical while the others are entirely numerical, unfortunately, it's restricted only in 2D spaces. The Algebraic Method is extremely fast, but the quality of the grid is empirical (SILVA; BARATA, 2011). The Differential Equations techniques have the advantage of quality control and, even they are not so fast compared to analytical methods (RUFFIN; ZAKI; SEKHAR, 2012), much effort has been made to improve these techniques (NOACK; ANDERSON, 1990). The Differential Equation techniques are also subdivided according to the type of PDE used.

Like elliptic PDEs, elliptic schemes result in smooth grids being a great advantage of this approach. It commonly uses the Poisson equations guaranteeing that the mapping of the domain can be undone (ANDERSON; TANNEHILL; PLETCHER, 2016). Another advantage of Poisson's equations is the ability to control the refinement of the grid by

altering some of its parameters. The disadvantage of this approach is the computational effort that demands to create the grid.

The others divisions are the hyperbolic and parabolic schemes. hyperbolic schemes are faster than the elliptic schemes once the equations that create the grids are calculated once (ANDERSON; TANNEHILL; PLETCHER, 2016), they are usually applied in an open system since it initiated by the initial points distribution and the outer boundary must be accepted wherever it occurs. The parabolic schemes are made by advanced the solution away from the initial data while satisfying the limits of the outer boundary (ANDERSON; TANNEHILL; PLETCHER, 2016).

The smoothness of the grids required for better accuracy is easily encountered in boundary-fitted grids accomplished by generalized coordinate systems (THOMPSON, 1982), where the grid spacing varies smoothly, and the angle between the gridlines does not become so small. Noack e Anderson (1990) explains that the need for the smoothness of the grid comes because of the numerical approximations of the derivatives, bringing robustness against folding (STEINBERG, 1993). The following section explains more about that coordinate system, exemplifying how to apply it in the numerical grids.

2.3 Generalised coordinates

In engineering problems, we often encounter flows phenomena in and around several situations where it is difficult to trace, in conventional coordinates, the lines of the computational grid to match with the physical boundaries bringing problems with the accuracy of the solution. Such difficulties motivate studies of generalized coordinates to map the physical space in a form that makes the lines of the computational grid coincide with the physical ones, resulting in a boundary-fitted grid (FLETCHER, 2012).

According to Maliska (2017), the use of generalized coordinates starts with researchers of the FDM method and was one of the most important research of this community to deal with arbitrary geometries. He explains that the main reasons to adopt this coordinate system are: the need to solve increasingly complex problems in arbitrary domains, difficulty to solve those problems in conventional coordinates systems, the possibility to concentrate the cells in the regions of interest, possibility to develop methodologies that can be generalized.

A generalized coordinate system can be formed by applying a distortion of the physical domain and mapping it into a simpler one, as exemplified in Figure 7. When a common coordinate system is already used to express the physical domain the distortion can be made by parametrizing these coordinates in a manner that the new coordinate system can accompany the desired shapes. According to Maliska (2017), the created grid in which the discretization came from another coordinate system results in a structured grid. For example, Figure 7 presents a grid made by applying a cut at the tail of the airfoil until the boundary of the domain and then following the shapes in a clockwise until return

to the cut segment resulting in a grid without a hole inside, simple connected (STEINBERG, 1993). Mathematically, the parametrization is made by writing the boundaries as the bottom and the top of the new domain stating one of the new coordinates constant while varying the other (THOMPSON, 1982).

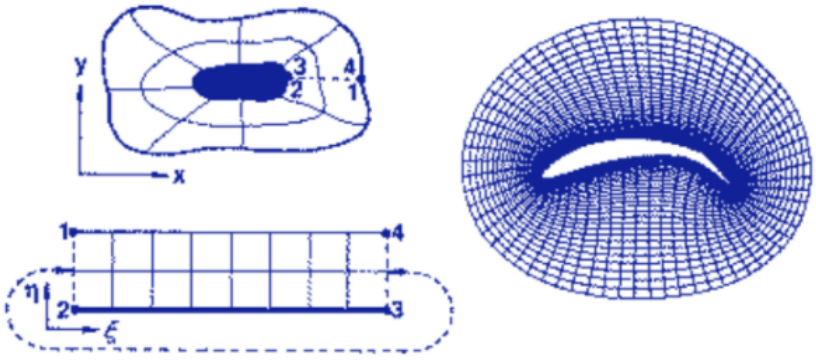
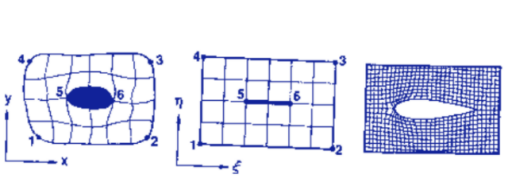


Figure 7 – Representation of an airfoil in generalised coordinates (THOMPSON; WARSI; MASTIN, 1985)

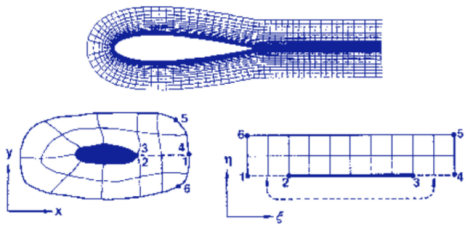
A complication appears when using generalized coordinates emerging as a consequence of this distortion, resulting in a parametrization of the governing equations and changes at the boundary conditions (FLETCHER, 2012). We present the parametrization of the equations in the next chapter, about the changes in the boundary conditions, it can be exemplified in Figure 7 where the effect of the surface of the airfoil will be present at the bottom of the new domain and no flux pass by it, the top has the boundary conditions at the borders of the physical one, and the horizontal limits of the logical domain are connected. However, that is just one case that can happen, the Figures 8 and 9 show others cases of grids created in generalized coordinates.

Figure 8 presents a situation similar to Figure 7, where grids are also applied around an airfoil, and the approach used to generate them could have been used in any case even the profile not being the same. In Figure 8a a slit configuration is shown (THOMPSON; WARSI; MASTIN, 1985), where the airfoil is transformed into a single segment at the new domain. Figure 8b shows a grid made almost by the same process of Figure 7, however, the cut segment is no longer at the right and left sides of the new domain but as part of the bottom together with the airfoil, and the horizontal limits are part of the outer edges leading to changes at the boundary conditions of the new domain. Comparing those 3 cases, we can see that different shapes of grids can be created in generalized coordinates, while Figure 7 and 8a resulted in an O-grid while Figure 8b, even being made in a similar way to the first, resulted in a C-grid.

Figure 9 presents how to generate grids even with more than one object is inside the boundary. Moreover, they also exemplify how the approach used to create the grids changes the refinement of the grid.



(a) Grid made by a slit scheme



(b) Grid made by a cut scheme

Figure 8 – Different distortion schemes to generate a grid (THOMPSON; WARSI; MASTIN, 1985)

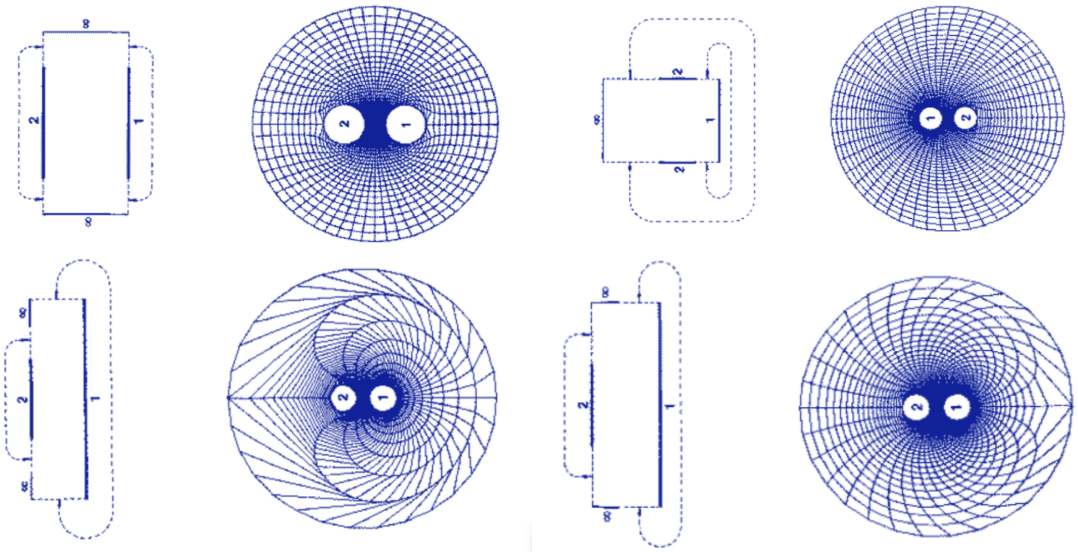


Figure 9 – Generalized coordinates cases with 2 circular bodies inside (THOMPSON; WARSI; MASTIN, 1985)

Those new configuration takes the domain initially expressed by x and y, considering a cartesian coordinate system, and applies a transformation that changes these coordinates to new coordinates, ξ and η as written below. Still we are considering 2 dimensions the same ideas proposed here are applied in any number of dimensions.

$$\xi = \xi(x, y) \quad \eta = \eta(x, y) \tag{2.1}$$

Steinberg (1993) explains that any space of n dimensions can be discretized in a logical space with k dimensions since $k \leq n$. The relation between the old and the new coordinates can be given by a matrix called the Jacobian matrix. To do that properly that matrix must be invertible so the change of coordinates can be unmade. The details of these transformations are shown in the chapter on the mathematical formulation.

3 Mathematical formulation

This section presents the mathematical formulation used in this work and starts presenting the transformation of the domain in generalised coordinates. Then, we present a brief explanation of the analytical formulation of the conservative equations that govern fluid dynamics, showing a dimensional analysis of those equations next, which permits us to apply the results of the equations to similar problems. We finish this chapter present transform the equations to the appropriate coordinate system.

3.1 Grid Generation

Elliptic Grid Generators are made initially considering that each coordinate in the computational space must satisfy an elliptic PDE in the physical domain being the Poisson's equation the most common to use due to its ability to control the distribution of the coordinate lines (THOMPSON; WARSI; MASTIN, 1985) witch permits to refine the grid in local regions changing some parameters. These equations can be rewritten in the computational domain as follows:

$$\begin{aligned} Ax_{\xi\xi} - 2Bx_{\xi\eta} + Cx_{\eta\eta} + D(Px_{\xi} + Qx_{\eta}) = 0 \\ Ay_{\xi\xi} - 2By_{\xi\eta} + Cy_{\eta\eta} + D(Py_{\xi} + Qy_{\eta}) = 0 \end{aligned} \tag{3.1}$$

The indices indicate a derivative at the corresponding coordinate and the constants are defined as:

$$\begin{aligned} A = x_{\eta}^2 + y_{\eta}^2 \quad B = x_{\eta}x_{\xi} + y_{\eta}y_{\xi} \\ C = x_{\xi}^2 + y_{\xi}^2 \quad D = (x_{\xi}y_{\eta} - x_{\eta}y_{\xi})^2 \end{aligned} \tag{3.2}$$

The P and Q parameters are the responsible to the refinement of the grid. As explained by Thompson, Warsi e Mastin (1985), Q will perform a refinement in the η coordinates leading to a more refine grid close to lower values of this coordinate, while P will make the same effect in relation to ξ however, since the values at the boundary are fixed, it will be more effective far from the boundary as illustrated in the Figure 10.

Figure 10 – Effect of P and Q parameters in the Poisson's Equations (THOMPSON, 1982)

In order to solve these equations in the computational domain, the same must be discretized by some numerical scheme. The derivatives present and the equations are calculated by a centered discretization. Resulting as the operations shown below where

the () must be replaced by the field variable, and i and j are positions related to ξ and η respectively.

$$\frac{\partial()}{\partial\xi} = \frac{()_{i+1,j} - ()_{i-1,j}}{2\Delta\xi} \tag{3.3}$$

$$\frac{\partial()}{\partial\eta} = \frac{()_{i,j+1} - ()_{i,j-j}}{2\Delta\eta} \tag{3.4}$$

$$\frac{\partial^2()}{\partial\xi^2} = \frac{()_{i,j+1} - 2()_{i,j} + ()_{i,j-j}}{\Delta\xi^2} \tag{3.5}$$

$$\frac{\partial^2()}{\partial\eta^2} = \frac{()_{i,j+1} - 2()_{i,j} + ()_{i,j-j}}{\Delta\eta^2} \tag{3.6}$$

$$\frac{\partial^2()}{\partial\xi\eta} = \frac{()_{i+1,j+1} + ()_{i-1,j+1} - ()_{i+1,j-j} - ()_{i-1,j-1}}{4\Delta\xi\Delta\eta} \tag{3.7}$$

Using the same notation above, the Poisson's equation can be rewrite as follows, where the derivatives are replaced and the equation is multiplied by 2.

$$\begin{aligned} 2A[()_{i,j+1} - 2()_{i,j} + ()_{i,j-j}] - B[()_{i+1,j+1} + ()_{i-1,j+1} - ()_{i+1,j-j} - ()_{i-1,j-1}] \\ +2C[()_{i,j+1} - 2()_{i,j} + ()_{i,j-j}] \\ +2\Delta\xi D\left(P[()_{i+1,j} - ()_{i-1,j}] + Q[()_{i,j+1} - ()_{i,j-j}]\right) = 0 \end{aligned} \tag{3.8}$$

Assuming the values of P and Q equal to 0, the system can be rearranged and the Poisson's equation turns into the Laplace's equation.

$$\begin{aligned} 2A(x_{i,j+1} + x_{i,j-j}) - 4(A+C)x_{i,j} &= B(x_{i+1,j+1} + x_{i-1,j+1} - x_{i+1,j-j} - x_{i-1,j-1}) \\ &\quad -2C(x_{i,j+1} + x_{i,j-j}) \\ 2A(y_{i,j+1} + y_{i,j-j}) - 4(A+C)y_{i,j} &= B(y_{i+1,j+1} + y_{i-1,j+1} - y_{i+1,j-j} - y_{i-1,j-1}) \\ &\quad -2C(y_{i,j+1} + y_{i,j-j}) \end{aligned} \tag{3.9}$$

Solve this equations are an iterative solution and requires a pre-made grid which can be made by any scheme. Noack e Anderson (1990) made a mixed hyperbolic/algebraic scheme to generate a parabolic scheme, however, the same can also be applied here. The development of their scheme starts by specifying the following relations where J^{-1} is the determinant of the inverse Jacobian and β is an orthogonality measure.

$$\begin{aligned} x_\xi y_\eta + x_\eta y_\xi &= J^{-1} \\ x_\xi x_\eta - y_\eta y_\xi &= \beta \end{aligned} \tag{3.10}$$

Considering an orthogonal grid, that is, considering $\beta = 0$, and solving these equations results in the next equations.

$$x_\eta = \frac{y_\xi S_\eta}{\sqrt{x_\xi^2 + y_\xi^2}} \qquad y_\eta = \frac{x_\xi S_\eta}{\sqrt{x_\xi^2 + y_\xi^2}} \tag{3.11}$$

$S\eta$ is the arc length with constant ξ given by the square root of the sum of the squares of the derivatives of the coordinates related to η. However, another way to express this arc can be given with relation to the distance R between the boundaries and a parameter used to control the grid spacing Δs.

$$S_\eta = \sqrt{x_\eta^2 + y_\eta^2} = R * \Delta s \tag{3.12}$$

Δs is given by the following equation where s is a function analytically specified, and the maximum index is the last index in the η direction.

Starting from the inner boundary, where $\eta = 0$, the increment of η is calculated based on the previous line. In order to obtain a orthogonal, or nearly orthogonal grid, the next points are found by summing the previous one with the derivative related to the η direction resulting in the equation below. Besides that, is considering that $\Delta\eta = \Delta\xi = 1$, which simplifies the calculation of the derivative.

$$\begin{aligned} x_{i,j}^o &= x_{i,j-1} + x_\eta \\ y_{i,j}^o &= y_{i,j-1} + y_\eta \end{aligned} \tag{3.13}$$

However, as it is not possible to get an completely orthogonal grid with it when both the inner and outer boundaries are specified, the result is relaxed by interpolated points calculated by:

$$\begin{aligned} x_{i,j}^{int} &= x_{i,j-1} + \Delta s(x_{i+1,j-1} - x_{i-1,j-1}) \\ y_{i,j}^{int} &= y_{i,j-1} + \Delta s(y_{i+1,j-1} - y_{i-1,j-1}) \end{aligned} \tag{3.14}$$

Finally, the grid is calculated by the next equations where ϵ is the ratio between the actual value of j minus 1 per the max index of j minus 1.

$$\begin{aligned} x_{i,j} &= \epsilon x_{i,j}^{int} + x_{i,j}^o(1 - \epsilon) \\ y_{i,j} &= \epsilon y_{i,j}^{int} + y_{i,j}^o(1 - \epsilon) \end{aligned} \tag{3.15}$$

3.2 Conservative Equations

To solve fluid dynamic problems using the CFD is necessary to apply the conservative equations that govern fluid dynamics. Although the conservative equations can be derived in two different ways, the statistical and the continuum approaches (CURRIE, 2016), we are interested in the continuum, considering the matter as a continuous medium where each point of the fluid has a unique value of pressure, velocity, density, and others field variables.

In this approach, there are two different reference frames to consider, the lagrangian, which follows an amount of matter in time, making the properties dependent on the initial position and the time, and the eulerian that pays attention in a fixed Con-

trol Volume (CV), a specified region in the observed domain, where the properties are dependent by the time and position.

The derivative of a property in the lagrangian perspective, called total derivative, has the time as the only independent variable while in the eulerian perspective, besides the time, there are the spatial derivatives as well. The relation of the derivatives of those perspectives can be seen at the equation below where ϕ is the analyzed property, D represents the derivative in the lagrangian approach, x and u represent respectively the position and the velocity where i is an index notation denoting all the dimensions of the space.

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x_i} u_i \tag{3.16}$$

Considering a flow in a CV, the variation of an extensive property, a property related to the amount of mass carried by this flow, is given by its variation inside the CV and the difference that passes the surface of this CV, the Control Surface (CS). This relation can be expressed by the equation below, where n is the intensive property, that is independent of the mass, the ρ is the specific density, $\forall$ and A are, respectively, the volume and the area that are encounter in the infinitesimal part in this equation and $\vec{V}$ is the velocity where the arrow indicates a vector.

$$\frac{dN}{dt} = \frac{\partial}{\partial t} \int_{CV} n\rho d\forall + \int_{CS} n\rho \vec{V} \cdot d\vec{A} \tag{3.17}$$

In the right side, the first term in related to the variation in time inside the CV while the last is the flow that pass by it. If the volume of the CV is fixed, applying the divergence theorem and derivative rules we reorganize the terms resulting in:

$$\frac{1}{\forall} \frac{dN}{dt} = \frac{D(n\rho)}{Dt} + n\rho(\nabla \cdot \vec{V}) \tag{3.18}$$

Those equations are called as Reynolds Transport Theorem (RTT) in its integral and derivative form, respectively. By that relation we derive all the conservative equations needed.

3.2.1 Continuity equation

One of the most important equations in fluid dynamics is the continuity equation. As by definition the amount of mass in a closed system is constant, there is no rate of mass in time and the RTT applied to the conservation of mass becomes:

$$0 = \frac{D\rho}{Dt} + \rho(\nabla \cdot \vec{V}) \tag{3.19}$$

Applying this result in the formulation of the RTT results in a simplification of the theorem that helps to derive the other equations.

$$\frac{1}{\forall}\frac{\partial N}{\partial t} = \rho\frac{Dn}{Dt} = \rho\left(\frac{\partial n}{\partial t} + \frac{\partial n}{\partial x_i}u_i\right) \tag{3.20}$$

Considering a specific mass constant in time and space, what is considered by a Mach number (Ma) lower than 0.3 the continuity equation is simplified.

$$0 = \nabla \cdot \vec{V}_r \tag{3.21}$$

3.2.2 Navier Stokes

The application of the RTT for momentum conservation results in other of the most important equations in fluid dynamics. The rate of the momentum in time results in the summation of forces applied to the system, those forces can be caused by both field or body forces. considering an infinitesimal cube the body forces applied can be expressed in terms of the differential of area times the tension, as shown in Figure 11 where τ_{ij} are the tensions at the face turned to the i axis and applied at j direction, i and j being the index notation where each one represents the dimensions of the space

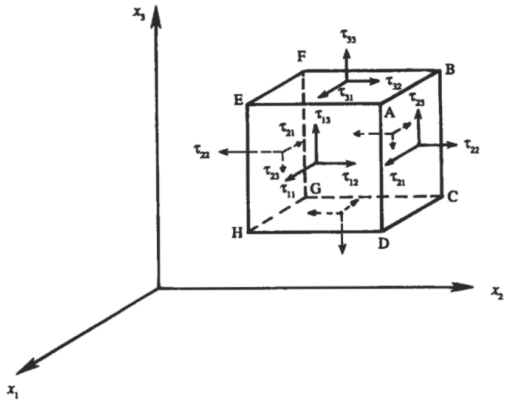


Figure 11 – Tensions in a infinitesimal cube (KUNDU; COHEN, 2002)

For a Newtonian fluid, the module of each tension can be expressed as follows, where δ_{ij} is the operator kronecker delta, being 1 when the index values are the same, and 0 when they are different.

$$\tau_{ij} = \delta_{ij}(-P + \lambda(\nabla \cdot \vec{V})) + \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{3.22}$$

As force and velocity are vectors, the RTT results in a vector equation. The summation of the body forces divided by the differential of volume results in the effect of the gradient of pressure and viscous forces, considering the fluid viscosity, μ, and specific

mass as constants results in the Navier-Stokes equations shown next, where P is the pressure and $\vec{B}$ are the field forces like gravity and magnetic forces.

$$-\nabla P + \mu\nabla^2\vec{V} + \frac{\vec{B}}{\forall} = \rho\frac{D\vec{V}}{Dt} \tag{3.23}$$

Unfortunately, these equations are too complicated to solve in most cases, so are usually made some considerations that simplify those equations.

3.3 Dimensional analysis

The solution of some problems can also be the solution for similar problems depending on what the phenomenon is looking for. This is very useful in the development of prototypes making it possible to create smaller or bigger versions of the model saving money for the project.

To make this possible, considering the difference in velocities, density, and other properties, the equations used must be dimensionless. Before starting, let's differentiate the dimensional properties with a line over, for example, the continuity equation is written now as:

$$0 = \frac{\partial\overline{\rho}}{\partial\overline{t}} + \sum_{i=1}^{3}\frac{\partial(\overline{u_{ri}}\cdot\overline{\rho})}{\partial\overline{x_i}}$$

Now considering that each variable has a related dimensionless variable that, multiplied by some characteristic value of the situation, gives us the dimensional variable. For the variables in the equation above let's consider.

$$\begin{aligned}\overline{\rho} &= \rho\cdot\rho_c\\ \overline{t} &= t\cdot t_c\\ \overline{u_{ri}} &= u_{ri}\cdot V_c\\ \overline{x_i} &= x_i\cdot L\end{aligned} \tag{3.24}$$

So the continuity equation becomes:

$$0 = \frac{\rho_c}{t_c}\left(\frac{\partial\rho}{\partial t}\right) + \frac{V_c\rho_c}{L}\left(\sum_{i=1}^{3}\frac{\partial(u_{ri}\cdot\rho)}{\partial x_i}\right)$$

To turn this equation back to the continuity equation form, we found that the characteristic time is given by:

$$t_c = L/V_c \tag{3.25}$$

Applying the same to the Navier-Stokes equations, with field forces and the viscosity as constants, and expressing the pressure as below to joint the forces of pressure together with the effect of the field forces, so the characteristic pressure is given as follows:

$$\overline{P} = P\cdot P_c \tag{3.26}$$

$$-\sum_{j=1}^{3}\frac{\partial\overline{P}}{\partial\overline{x_i}}+\mu\sum_{j=1}^{3}\frac{\partial^2\overline{u_i}}{\partial\overline{x_j}^2}+\frac{\vec{B_i}}{\forall}=\overline{\rho}(\frac{\partial\overline{u_i}}{\partial\overline{t}}+\sum_{j=1}^{3}\frac{\partial\overline{u_i}}{\partial\overline{x_j}}\overline{u_{rj}})$$

$$-\frac{P_c}{L}\sum_{j=1}^{3}\frac{\partial P}{\partial x_i}+\frac{V_c}{L^2}\mu\sum_{j=1}^{3}\frac{\partial^2 u_i}{\partial {x_j}^2}+\frac{\vec{B_i}}{\forall}=\rho_c\frac{V_c^2}{L}\rho(\frac{\partial u_i}{\partial t}+\sum_{j=1}^{3}\frac{\partial u_i}{\partial x_j}u_{rj}) \tag{3.27}$$

$$P_c=\rho V_c^2 \tag{3.28}$$

However, for other terms is useful to, instead solving it, replace by an unique term called dimensionless number. The dimensionless number of the equation above with the viscosity is called Reynolds number (Re) $=\frac{\rho V_c L_c}{\mu}$, that number can be seen as the ratio between the viscosity forces, μ and the inertial forces, $\rho V_c L_c$ so, the bigger the Reynolds number, more the viscous forces domain the flow.

The case of the field forces depends of what forces are applied, to the gravitational forces the relation becomes, where Froude number (Fr) represents the ratio between inertial forces and the gravitational force.

$$\frac{L}{\rho_c V_c^2}\cdot\frac{mg_i}{\forall}=\frac{Lg_i}{V_c^2}=\frac{1}{Fr^2} \tag{3.29}$$

Using those dimensionless numbers the Navier-Stokes equations becomes:

$$-\nabla P+\frac{1}{Re}\nabla^2\vec{V}+\frac{1}{Fr^2}=\rho\frac{D\vec{V}}{Dt} \tag{3.30}$$

This is one way to dimensionless the Navier-Stokes equations, other forms can be, considering the pressure a constant as the viscosity and field forces, it will result in the Euler number (Eu) $=\frac{P}{\rho V_c}$ witch represents the ratio between the pressure and inertial forces. Another way is to joint the effect of field forces together with the pressure, considering the gravitational force as the only field force, and considering $\overline{P}=P\cdot P_c-\rho_c g L_c$:

3.4 Change of coordinate of the conservative equations

The equations presented so far are applied to the cartesian system. As the grid will be made in generalized coordinates, we must make the necessary changes to fit it in the new coordinate system. This transformation is made by applying a transformation matrix, also called Jacobian matrix, that is expressed by:

$$\mathbf{J}=\begin{bmatrix}\xi_x & \xi_y\\ \eta_x & \eta_y\end{bmatrix} \tag{3.31}$$

Where each element of this matrix is the derivative of one new coordinate by an old one, indicated by the index. A requirement here is that this matrix must be invertible

which makes it possible to return the new coordinate system to the before by applying the inverse of this matrix which is the same to write the transformation matrix from the generalized coordinates to the cartesian and can be expressed as follows:

$$\mathbf{J}^{-1} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \tag{3.32}$$

By solving the inverse of that matrix we can express the following equations that will be used to derive the equations in the appropriated coordinate system. Here J is the determinant of the transformation matrix, the jacobian.

$$\xi_y = Jx_\eta \quad \xi_x = -Jy_\eta \quad \eta_y = -Jx_\xi \quad \eta_x = Jy_\xi \tag{3.33}$$

The following equation shows how the jacobian matrix is used to make the change of coordinates, then the case of the velocity is shown, which results in the contravariant velocities which commonly appear in the conservative equations in generalized coordinates.

$$\begin{bmatrix} \phi_\xi \\ \phi_\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} \tag{3.34}$$

$$\vec{V} = \begin{bmatrix} V_\xi \\ V_\eta \end{bmatrix} = \begin{bmatrix} \xi_x u_x + \xi_y u_y \\ \eta_x u_x + \eta_y u_y \end{bmatrix} = \begin{bmatrix} J(-y_\eta u_x + x_\eta u_y) \\ J(y_\xi u_x - x_\xi u_y) \end{bmatrix} \tag{3.35}$$

Besides that, the derivatives will also change, the following equations present how the derivative becomes, then how it results in the divergence operator, and with that in mind, how the total derivative becomes:

$$\frac{\partial \phi}{\partial x_i} = \frac{\partial \phi}{\partial \xi}\frac{\partial \xi}{\partial x_i} + \frac{\partial \phi}{\partial \eta}\frac{\partial \eta}{\partial x_i} \tag{3.36}$$

$$\nabla \cdot \vec{\phi} = \frac{\partial}{\partial \xi}(\xi_x \phi_x + \xi_y \phi_y) + \frac{\partial}{\partial \eta}(\eta_x \phi_x + \eta_y \phi_y) = \frac{\partial}{\partial \xi}\phi_\xi + \frac{\partial}{\partial \xi}\phi_\eta \tag{3.37}$$

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial \xi}V_\xi + \frac{\partial \phi}{\partial \eta}V_\eta \tag{3.38}$$

By applying those transformations in the RTT equation, it becomes:

$$\left(\frac{1}{\forall}\frac{dN}{dt}\right)_{\xi\eta} = \left(\frac{\partial \phi\rho}{\partial t} + \frac{\partial \phi\rho}{\partial \xi}V_\xi + \frac{\partial \phi\rho}{\partial \eta}V_\eta\right) + \phi\rho\left(\frac{\partial V_\xi}{\partial \xi} + \frac{\partial V_\eta}{\partial \eta}\right) \tag{3.39}$$

It results in the following continuity equation, being the others conservative equations derived analogously.

$$0 = \frac{\partial \rho}{\partial t} + \rho\left(\frac{\partial V_\xi}{\partial \xi} + \frac{\partial V_\eta}{\partial \eta}\right) \tag{3.40}$$

4 Methodology

This chapter presents the procedure made in the development of this work to create, test and evaluate the grids.

4.1 Characterization of this work

The approach of this work is classified as quantitative, having an applied nature since the results can be applied in future simulation (CESARIO; FLAUZINO; MEJIA, 2020).

According to Gil et al. (2002), this work can be classified as exploratory since it objectives to get familiar with the numerical simulations, in specific with the numerical grids. Another classification that fits in this work is to be experimental once we plan to test our grids to provide a good accuracy to the simulations.

4.2 Work Procedures

To make this work we use the Elliptical PDEs technique, by using the Poisson's equations discretized, to create a grid in the 2D domain in generalized coordinates, solving the equations by applying the Thomas' algorithm to calculate the solutions of system with tridiagonal equations. The results were calculated using the FORTRAN language while all the plots were created in python where the codes are presented in the attachments.

The transformation of the domain is made by applying a cut between the boundaries of the domain and considering those boundaries as lines with constant η while ξ is increased in clockwise, this is exemplified in the Figure 12 where a general domain is represented at the left and at the right is the result of the transformation applied.

The first step of creation of the grid is to delimit the inner and outer boundaries of the domain in cartesian coordinates. Then, with the outer and inner boundaries defined, as the elliptical technique requires a pre-made grid to be created, we create a reference grid by following the work of Noack e Anderson (1990) as presented before. And to finish the grid we apply the elliptical PDEs by solving the Poisson's equations through the

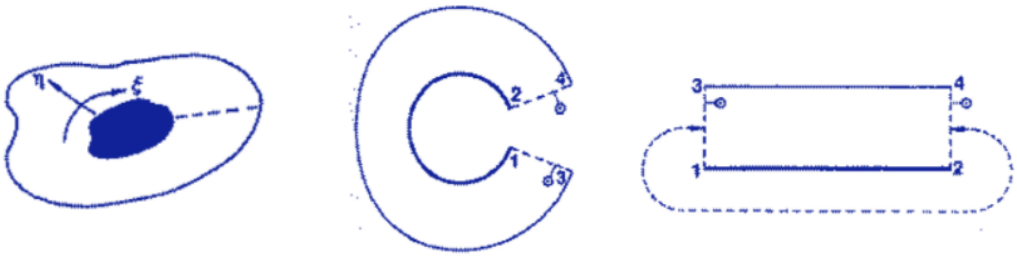


Figure 12 – Exemple of distorsion applied in this work (THOMPSON; WARSI; MASTIN, 1985)

Thomas' algorithm where the minimum error, the difference between the current result with the result of the previous iteration, was fixed by 10^{-6}.

5 Results

In this section, we present and discuss the results obtained by our generator. We first defined the boundaries to be used in this work where we defined 80 as the fixed number of points that delimit them.

For the outer boundaries we use the most common shapes used in the works: a circle, a square and a C-shaped boundary, as represented in Figure 13. For the inner boundaries we also use the square and the circle, but we also added an variation for each one, representing a deformed square and a deformed circle. Furthermore we also added 2 airfoils, an symmetrical airfoil named NACA 0012, and an asymmetrical airfoil named NACA 8412. The new shapes are present in Figure 14.

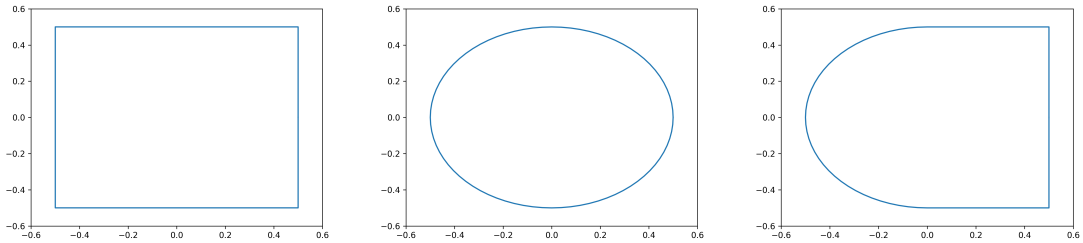


Figure 13 – Outer boundaries

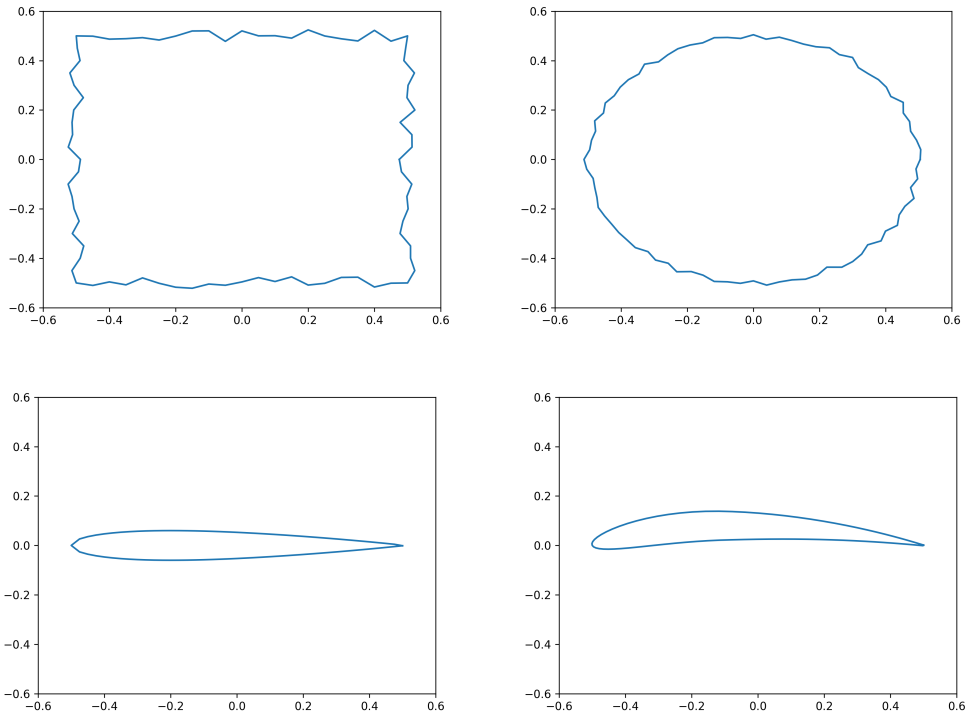


Figure 14 – Inner boundaries

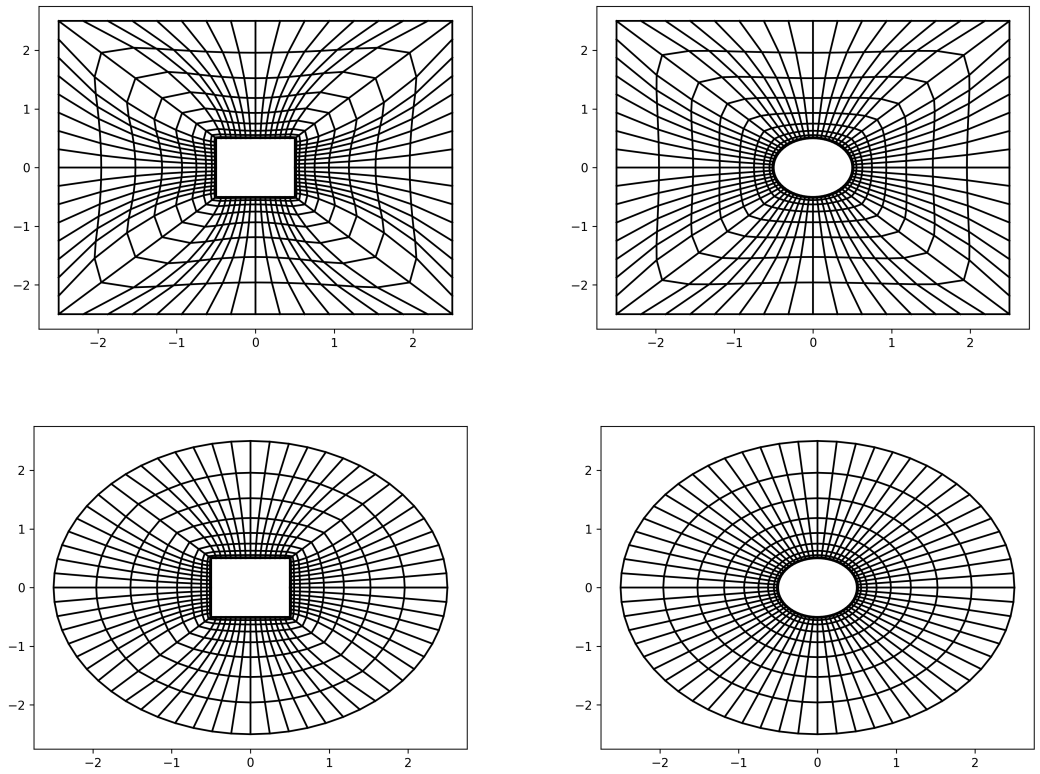


Figure 15 – Reference grid to simple geometries

With the boundaries delimited, we try replicate the Noack e Anderson (1990) reference grid and test it with the inner and outer boundaries as simple geometric shapes, which results in the Figure 15. Through those tests we assure that the reference grid is working and we can pass to the next step where we apply of our generator.

It can be seen that in some cases, mainly when the boundaries has corners, the reference grid is not smooth which can cause accuracy errors when applying it in simulations. This is not seen in the case which the inner and outer boundaries are a circle, where the geometry makes the reference grid smoothly already.

Applying our generator, we aim to improve the previous grid to result in a more smooth and orthogonal grid. The complexity of our generator can be calculated by the complexity of the Thomas' algorithm is $O(n)$, that is, the computational effort of each iteration of the Thomas' algorithm is directly proportional to number of η lines in the grid. The algorithm is repeated until converge, where the convergence criterion was to get an error, the difference between the current and last iteration, less than 10^{-6}.

Apply our generator in the reference grid resulted in a more smoothly grid, Figure 16, furthermore, we also have am impact in the distribution of the η lines, showing the effect or our generator in both, η and ξ directions.

After test with simple geometries and knowing that our program is working, we

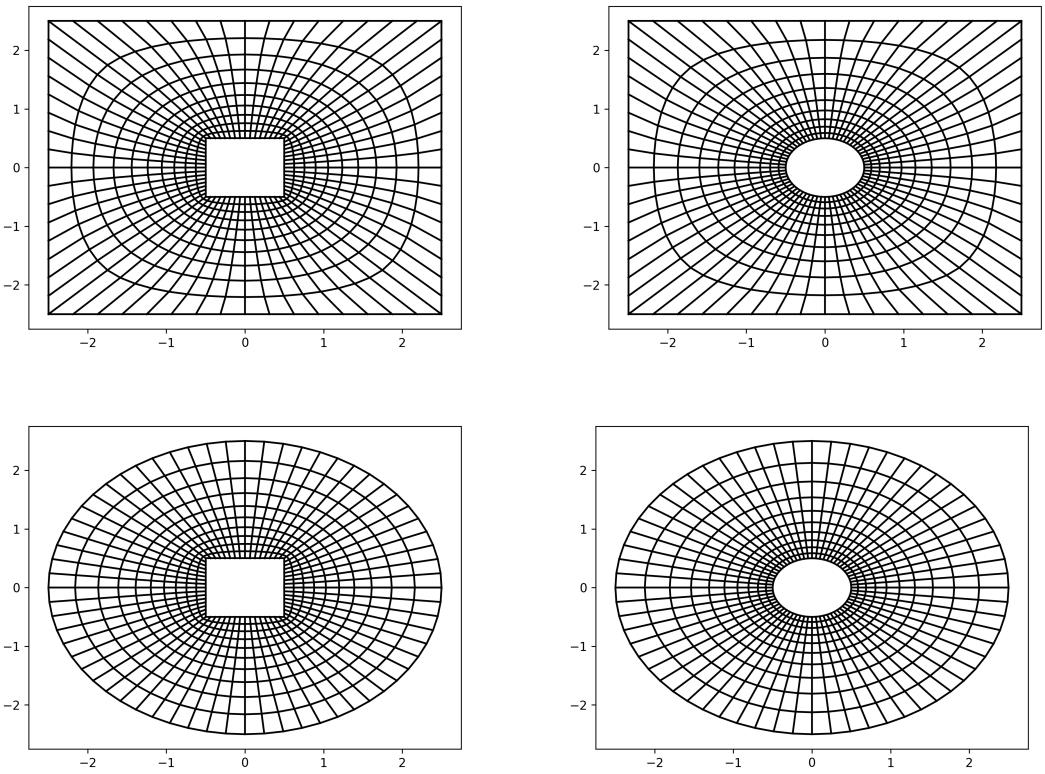


Figure 16 – Final grid to simple geometries

apply the generator to complex geometries. In Figure 17, comparing the reference grid, at left, with the final grid, at right, we see the improvement that our generator makes to previous grid, turning it smooth and orthogonal.

It shows that our generator is working, able to follow complex geometries and create and smooth and orthogonal grid. Still, our grid can be improved by refining the grid, increasing the number of points at the η direction.

That refinement costs more iterations to converge the program as exemplified in the Figure 18 where in upper part of the figure we present a less refined grid at left and a more refined grid at right (5 and 50 η lines, respectively) and in the lower part of the figure we present the evolution of the convergence for different refinements. It's also important to notice that not also the number of iterations was increased but also the computational power for each interaction given the complexity of the algorithm.

Figure 19 shows the evolution of the convergence for different outer and inner boundaries. It can be seen yet that, a more refined grid starts with a lower error when comparing it with a less refined.

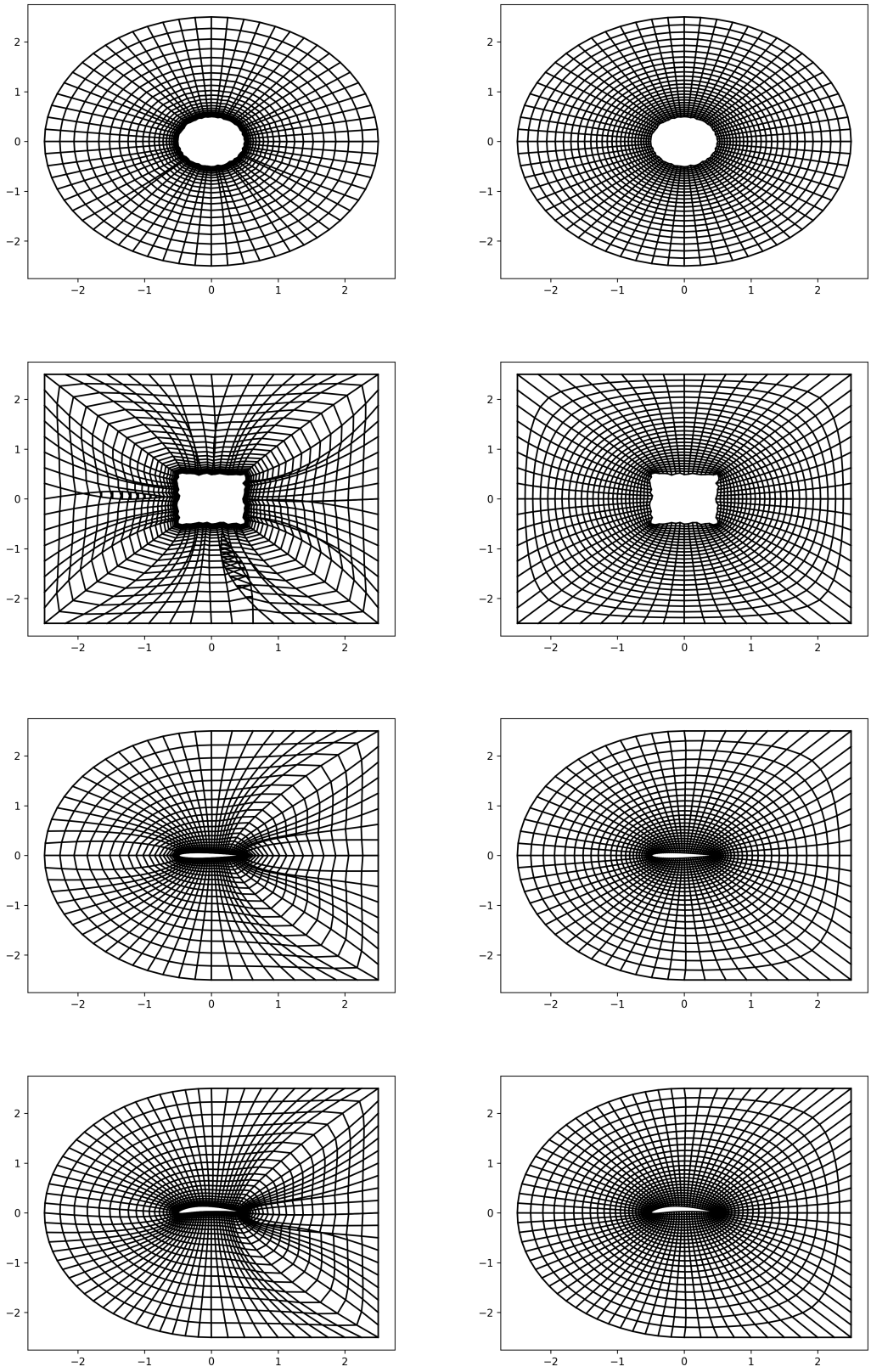
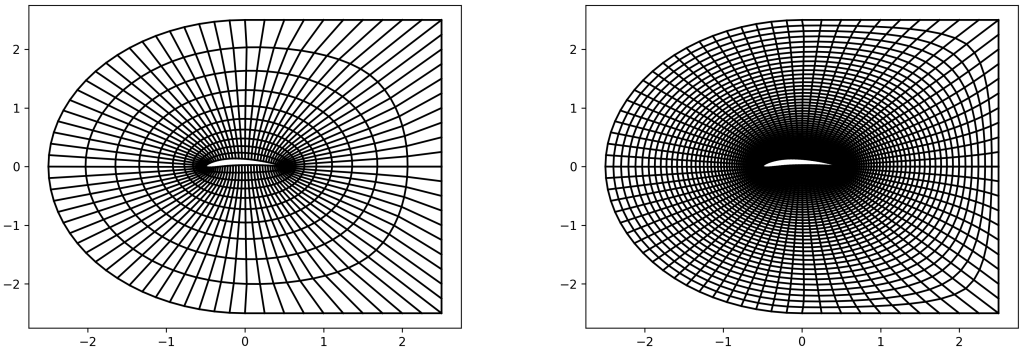


Figure 17 – Reference grid and final grid for more complex geometries



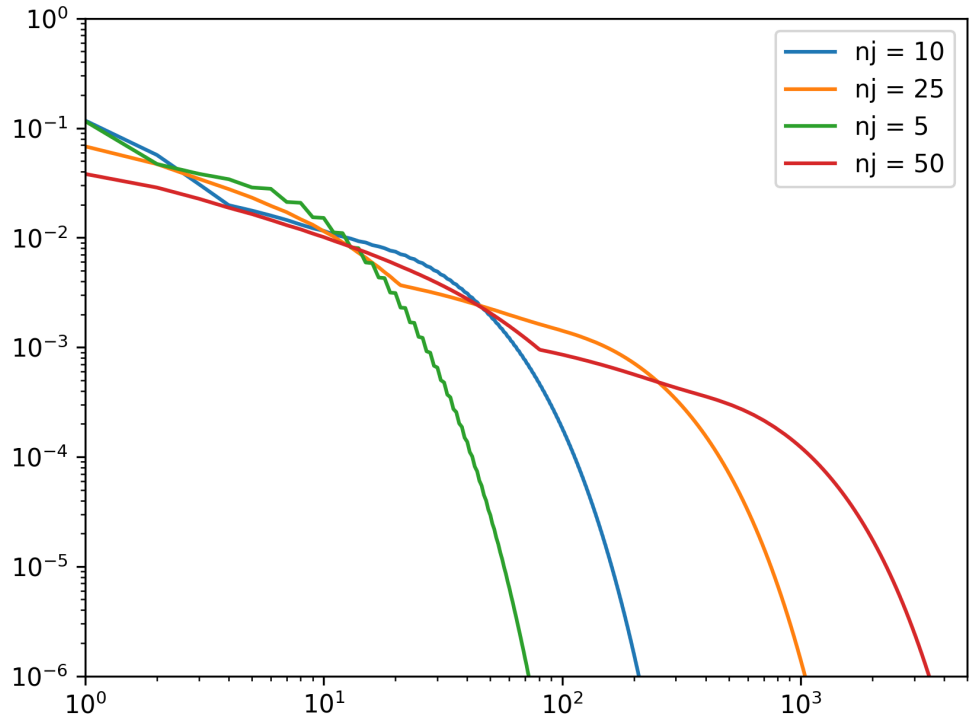


Figure 18 – Airfoil 8412 convergence related to the refinement

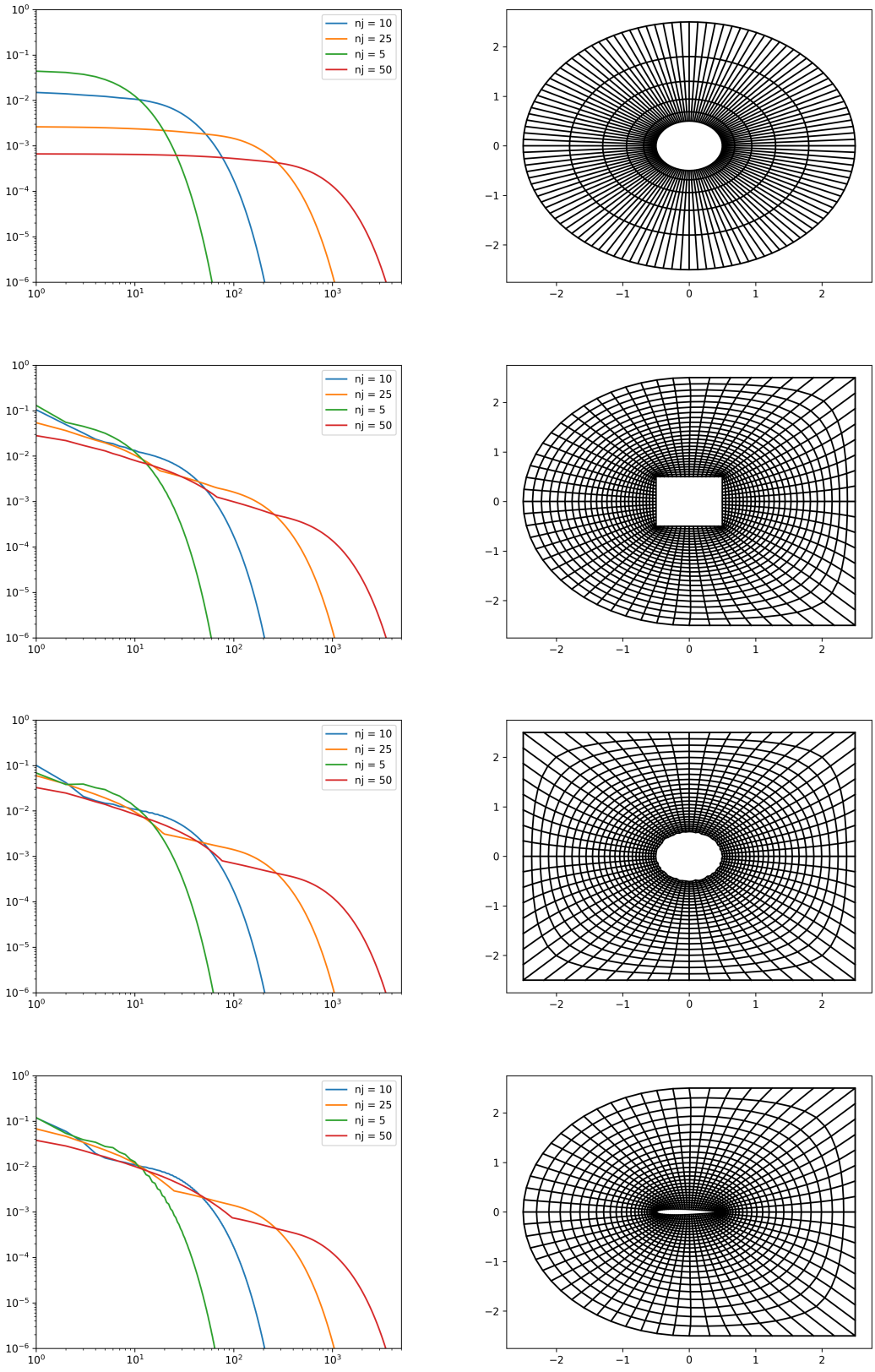


Figure 19 – Reference grid and final grid for more complex geometries

6 Final Remarks

In this chapter, we provide an overview of the main points covered throughout this work. In addition, we review the results and contributions of this work. Then, to finish this work, we describe and present possible future work related to the further steps, issues not addressed in the scope of this work, and solutions to limitations.

In this work we present an grid generator to fluid simulations by using elliptical PDEs in order to address an smoothly and orthogonal grid that follows complex geometries. We applied our generator to create grids between various boundaries from the simplest to more complex geometries, evaluating the convergence of the algorithm through the errors, comparing the number of interactions needed and evaluating through different refinement levels

6.1 Achievements

Resuming the objectives of this work, the development of a program to create a numerical grid at generalised coordinates using elliptical PDEs was reached. The resulted grids were presented in the previous chapter for 3 outer boundaries and 6 inner boundaries, reaching the objective of test of the program. Our generator converged for all the proposed cases, we notice that the convergence was proportional to the refinement of the grid at the η direction. We also notice that the error at the initial of our generator was bigger with more complex geometries, however, it does not had an significant impact in the algorithm convergence.

6.2 Future Work

In this section, we present possible future directions for this work, including topics not covered in the scope of this work and solutions to the current limitations presented.

First, the main limitation of this work was not to test our grid in real problems. An interesting work can be done by adding a real body contour as an inner boundary, applying our generator to create a grid, running a simulation in CFD and comparing with an experimental result.

Bibliography

ANDERSON, D.; TANNEHILL, J. C.; PLETCHER, R. H. **Computational fluid mechanics and heat transfer**. [S.l.]: Taylor & Francis, 2016. Cited 5 times in the pages 15, 19, 21, 22, and 23.

CESARIO, J. M. dos S.; FLAUZINO, V. H. D. P.; MEJIA, J. V. C. Metodologia científica: Principais tipos de pesquisas e suas caraterísticas. **Revista Científica Multidisciplinar Núcleo do Conhecimento**, p. 23–33, 11 2020. Cited in page 35.

CHAN, W. et al. Best practices in overset grid generation. In: **32nd AIAA Fluid Dynamics Conference and Exhibit**. [S.l.: s.n.], 2002. p. 3191. Cited 2 times in the pages 9 and 16.

CURRIE, I. G. **Fundamental mechanics of fluids**. [S.l.]: CRC press, 2016. Cited in page 29.

DJEDDI, R.; EKICI, K. A turbulent low-speed preconditioner for unsteady flows about wind turbine airfoils. In: **22nd AIAA Computational Fluid Dynamics Conference**. [S.l.: s.n.], 2015. p. 3202. Cited in page 21.

FLETCHER, C. A. **Computational techniques for fluid dynamics; Vol 1**. [S.l.]: Springer, Springer, NY (United States), 1991. Cited in page 19.

FLETCHER, C. A. **Computational techniques for fluid dynamics 2: Specific techniques for different flow categories**. [S.l.]: Springer Science & Business Media, 2012. Cited 3 times in the pages 15, 23, and 24.

GIL, A. C. et al. **Como elaborar projetos de pesquisa**. [S.l.]: Atlas São Paulo, 2002. v. 4. Cited in page 35.

HIESTER, H. et al. Assessment of spurious mixing in adaptive mesh simulations of the two-dimensional lock-exchange. **Ocean Modelling**, v. 73, p. 30–44, 2014. ISSN 1463-5003. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1463500313001844>. Cited 3 times in the pages 9, 21, and 22.

JOHNSON, F. T.; TINOCO, E. N.; YU, N. J. Thirty years of development and application of cfd at boeing commercial airplanes, seattle. **Computers Fluids**, v. 34, n. 10, p. 1115–1151, 2005. ISSN 0045-7930. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0045793005000125>. Cited 2 times in the pages 15 and 17.

KUNDU, P. K.; COHEN, I. M. **Fluid mechanics**. [S.l.: s.n.], 2002. Cited 3 times in the pages 9, 19, and 31.

LOMAX, H.; PULLIAM, T.; ZINGG, D. **Fundamentals of Computational Fluid Dynamics**. Springer Berlin Heidelberg, 2013. (Scientific Computation). ISBN 9783662046548. Disponível em: <https://books.google.com.br/books?id=PFvxCAAAQBAJ>. Cited in page 15.

MALISKA, C. R. **Transferência de calor e mecânica dos fluidos computacional .** [S.l.]: Grupo Gen-LTC, 2017. Cited 6 times in the pages 15, 16, 19, 20, 22, and 23.

MARCHI, C. H. et al. Highly accurate numerical solutions with repeated richardson extrapolation for 2d laplace equation. **Applied Mathematical Modelling**, v. 37, n. 12, p. 7386–7397, 2013. ISSN 0307-904X. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0307904X13001509>. Cited 2 times in the pages 20 and 21.

MOUKALLED, F.; MANGANI, L.; DARWISH, M. **The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®**. [S.l.: s.n.], 2015. v. 113. ISBN 978-3-319-16873-9. Cited in page 19.

NOACK, R. W.; ANDERSON, D. A. Solution-adaptive grid generation using parabolic partial differential equations. **AIAA journal**, v. 28, n. 6, p. 1016–1023, 1990. Cited 7 times in the pages 20, 21, 22, 23, 28, 35, and 38.

RUFFIN, S.; ZAKI, M.; SEKHAR, S. K. A normal ray refinement technique for cartesian-grid based navier-stokes solvers. **International Journal of Computational Fluid Dynamics - INT J COMPUT FLUID DYNAMICS**, v. 26, p. 231–246, 04 2012. Cited 2 times in the pages 21 and 22.

SCHNEIDER, F. A.; MARCHI, C. H. On the grid refinement ratio for onedimensional advective problems with nonuniform grids. In: **18th INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING**. [S.l.: s.n.], 2005. v. 11. Cited in page 19.

SILVA, A.; BARATA, J. Grid generation with boundary point distribution control on heterogeneous parallel architectures. **International Review of Aerospace Engineering**, v. 4, p. 48, 02 2011. Cited 2 times in the pages 21 and 22.

STEINBERG, S. **Fundamentals of grid generation**. [S.l.]: CRC press, 1993. Cited 4 times in the pages 21, 23, 24, and 25.

THOMPSON, J. F. Elliptic grid generation. **Applied Mathematics and Computation**, v. 10-11, p. 79–105, 1982. ISSN 0096-3003. Disponível em: <https://www.sciencedirect.com/science/article/pii/0096300382901886>. Cited 5 times in the pages 9, 17, 23, 24, and 27.

THOMPSON, J. F.; WARSI, Z. U.; MASTIN, C. W. **Numerical grid generation: foundations and applications**. [S.l.]: Elsevier North-Holland, Inc., 1985. Cited 6 times in the pages 9, 22, 24, 25, 27, and 35.

WANG, Y.; QUAINI, A.; CANIC, S. A higher-order discontinuous galerkin/arbitrary lagrangian eulerian partitioned approach to solving fluid–structure interaction problems with incompressible, viscous fluids and elastic structures. **Journal of Scientific Computing**, v. 76, 07 2018. Cited 2 times in the pages 9 and 20.

YU, N. Grid generation and transonic flow calculations for three-dimensional configurations. In: **13th Fluid and PlasmaDynamics Conference**. [S.l.: s.n.], 1980. p. 1391. Cited 2 times in the pages 9 and 17.

ZAKARIA, M. S. et al. A cartesian non-boundary fitted grid method on complex geometries and its application to the blood flow in the aorta using openfoam. **Mathematics and Computers in Simulation**, v. 159, p. 220–250, 2019. ISSN

0378-4754. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0378475418303094>. Cited 2 times in the pages 9 and 16.

ZANTE, D. E. V. et al. Progress in open rotor propulsors: The faa/ge/nasa open rotor test campaign. **The Aeronautical Journal (1968)**, Cambridge University Press, v. 118, n. 1208, p. 1181–1213, 2014. Cited in page 15.