High-Order Boundary Treatment for High-Order Methods in Computational Fluid Dynamics

André Ribeiro de Barros Aguiar, ufabc.andre@gmail.com[1]
Carlos Breviglieri, carbrevi@gmail.com[1]
Fábio Mallaco Moreira, fabiom91@gmail.com[1]
Eduardo Jourdan de Araújo Jorge Filho, eduardojourdan92@gmail.com[1]
João Luiz F. Azevedo, joaoluiz.azevedo@gmail.com[2]

[1] Instituto Tecnológico de Aeronáutica, DCTA/ITA, São José dos Campos, SP, 12228-900, Brazil
[2] Instituto de Aeronáutica e Espaço, DCTA/IAE/ALA, São José dos Campos, SP, 12228-904, Brazil

Abstract: *In the context of high-order numerical schemes in computational fluid dynamics (CFD), a meaningfully accurate solution can only be obtained if the corresponding high-order approximation of curved boundaries of the geometry is used. A study of the main issues and benefits of using such curved boundary treatment for high-order methods is developed in the present work. A methodology to project nodes from an original linear unstructured 2-D mesh onto a new curved mesh is detailed and its effectiveness is assessed using a high-order spectral difference (SD) scheme on standard compressible flow test cases. The methodology handles the complete description of the 2-D geometry as represented by a Non-Uniform Rational B-Spline (NURBS) curve entity. Flows of interest are assumed to be adequately modeled by the two-dimensional (2-D) Euler equations. Furthermore, the validation of the results is carried out using a subsonic flow over a 2-D cylinder, in which the issues arising from the use of straight-sided meshes is highlighted. The resulting framework is also applied to flows over a NACA 0012 airfoil configuration so as to demonstrate the capability implemented.*
Keywords: *Unstructured High-Order Meshes, Curved Boundaries, NURBS, Spectral Differences Scheme, CFD*

Introduction

High-order numerical schemes represent the natural extension of current computational fluid dynamics (CFD) methods, which were developed over the past thirty years for aerospace simulations. The current generation methods are mostly 2nd-order accurate and have achieved a level of maturity and robustness desirable for everyday deployment in aeronautical engineering scenarios. Likewise, several complementary methods were developed for time integration, convergence acceleration, shock capturing and for dealing with geometric complexities. However, there are many problems that cannot be fully simulated using low-order methods, such as vortex dominated flows. Moreover, high-order methods offer the possibility to reduce simulation costs for given solution accuracy levels, when compared to low-order schemes.

The precision and efficiency on the convergence order of the method is negatively affected by an unsuitable representation of the physical domain boundaries (Bassi and Rebay, 1997a). Therefore, a high-order method coupled with a low-order mesh with linear elements, for instance, will degrade the accuracy of the method near the domain boundaries. In order to overcome this problem and to fully benefit from the advantages of high-order methods, a correct description of curved boundaries is mandatory and, thus, high-order schemes must cope with mesh manipulation techniques that need to be superior, in comparison with its low-order counterparts.

Even though 2nd-order accurate methods are mainly used, for the past two decades, an active research community (Wang *et al.*, 2013) has put a great effort concerning the continuous development of high-order methods. Since high-order methods are currently not well established, companies that invest on the development of commercial mesh generation codes are not financially motivated to offer the same level of support given to linear mesh generation. As a consequence, commercial mesh generation software present significant limitations as regards to features on curved mesh generation.

In this sense, higher order mesh generators are not readily available for academia nor industry. Although an open source and well known high-order mesh generator called GMSH (Geuzaine and Remacle, 2009) does exist, the software is not sufficiently robust in the sense that a non-expert user is usually not comfortable to generate curved meshes. Moreover, GMSH has limited features as regarding to high-order elements such as limited choice of order and incapability of dealing with other than equally spaced node positioning which further leads to numerical instabilities for higher orders (Hesthaven, 1998).

There is room for improvement in many areas for high-order methods that must be pursued before they can compete with industrial-level CFD solvers. Computational resource requirements and run time are typical metrics used to classify a specific method or a combination of methods in a CFD solver. High-order schemes must cope with implicit schemes, limiters or filters and mesh manipulation techniques that also need to be superior, in comparison with the low-order counterparts. Therefore, a high-order method coupled with a low-order mesh with linear elements, for instance, will degrade the accuracy of the method near the domain boundaries. In order to overcome this problem and to fully realize the advantages of higher-order methods, a correct description of curved boundaries is mandatory. The solver is currently enabled with implicit methods for convergence acceleration whereas the mesh capabilities are still being recently developed.

The main contribution of the present work is to highlight the benefits of using a proper description of curved boundary near the geometry wall. This study was consolidated in terms of a library that outputs a high-order mesh given a ge-

neral unstructured linear mesh.The aforementioned solutions improves the scheme robustness, performance and domain applicability as will be discussed later.

The paper is organized as follows. Section 2.describes the inviscid flow formulation used for the test cases while Section 3presents the numerical formulation of the high-order SD scheme. Sections 4and 5give further simulation details whereas Section 6.addresses the high-order boundary treatment methodology aspects. Section 7.presents the numerical results computed with the SD method with both linear and curved meshes for the inviscid simulations and compare the results. Finally, Section 8.draws concluding remarks and discusses suggestions for improvements in the context of the present work.

The Euler Equations

The Euler equations describe the flow configuration for an inviscid, non-heat conducting fluid. They can be obtained by neglecting all viscous stresses and heat conduction terms present in the Navier-Stokes equations. The resulting system of first order partial differential equations is presented in its differential form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} = 0 \, . \tag{1}$$

The vector of conserved variables, $\mathbf{Q}$, and the convective flux vectors, $\mathbf{E}$ and $\mathbf{F}$, are defined in 2-D by

$$\mathbf{Q} = \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ \varepsilon \end{array} \right\} \, , \quad \mathbf{E} = \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\varepsilon + p)u \end{array} \right\} \, , \quad \mathbf{F} = \left\{ \begin{array}{c} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\varepsilon + p)v \end{array} \right\} \, . \tag{2}$$

Where ρ, u and v are respectively the specific mass of the fluid and the velocities in x and y directions. The system is closed by the equation of state for a perfect gas, where the pressure p can be obtained from

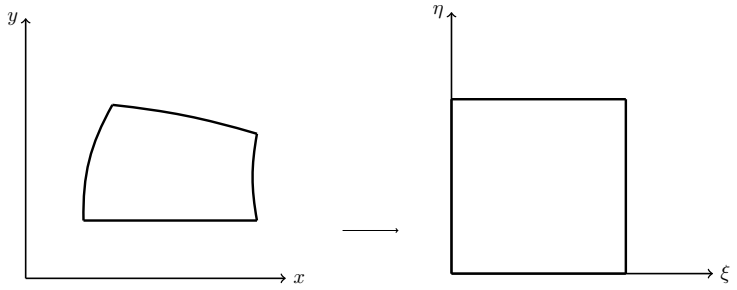
$$p = (\gamma - 1) \left[\varepsilon - \frac{1}{2} \rho (u^2 + v^2) \right] , \tag{3}$$

where the ratio of specific heats, γ, is set as 1.4 for all computations and ε is the total energy per unit volume defined in terms of the internal energy e and the kinetic energy as in equation 4.

$$\varepsilon = \rho \left(e + \frac{1}{2} \left\| \mathbf{v} \right\|^2 \right) \tag{4}$$

Numerical Formulation of the Spectral Difference Scheme

The Spectral Difference method (SD) applies a finite difference-like scheme within each cell of an unstructured mesh domain (Liu *et al.*, 2006). This work will perform studies on quadrilateral grids only. To achieve an efficient implementation, all elements in the physical domain, (x, y), are mapped into a unit standard square element in the computational domain (ξ, η) as depicted in Fig. 1.



(a) Physical Element (c) Standard Element.

Figura 1: Transformation from physical domain (x,y) to computational domain (ξ,η).

Each cell is then, transformed into the same standard element by a transformation expressed by a linear system with a total of K shape functions

$$\left(\begin{array}{c} x \\ y \end{array} \right) = \sum_{s=1}^{K} M_s(\xi, \eta) \left(\begin{array}{c} x_s \\ y_s \end{array} \right) \, , \tag{5}$$

and K is the number of points used to define the physical element, (x_s, y_s) are the Cartesian coordinates of these points, and $M_s(\xi, \eta)$ are the shape functions of the geometric transformation. In the case of a 1st-order linear boundary mesh, the transformation are bilinear and the analytic expression can be easily found. However, for the case of higher order meshes the number of points used to define a single cell increases and hence, a more well behaved transformation can be obtained. The metrics and the Jacobian matrix of the transformation can be computed in a pre-processing step and kept in memory given the stationary aspect of the mesh for the problems considered. The implementation follows the formulation presented by Wang *et al.* (2007) and May and Jameson (2006). The governing equations in the physical domain are, then, transferred into the computation domain, where they are rewritten as

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} = 0 \,, \tag{6}$$

where $\tilde{Q} = |J|\,\mathbf{Q}$ and J is the Jacobian matrix of the coordinate transformation given by

$$J = \frac{\partial(x, y)}{\partial(\xi, \eta)} = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} . \tag{7}$$

For practical computations, the inverse Jacobian is mainly used, and, for the current implementation, the flux vectors in the computational domain can be simplified from the general form as

$$\begin{pmatrix} \tilde{E} \\ \tilde{F} \end{pmatrix} = |J| \cdot \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} E \\ F \end{pmatrix} = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix}^{-1} \begin{pmatrix} E(\tilde{Q}) \\ F(\tilde{Q}) \end{pmatrix} . \tag{8}$$

In the standard element, two sets of points are defined, namely the solution points (SP) and the flux points (FP). An internal discretization that only requires dealing with one-dimensional problems was selected. An example of such distribution, in a third order cell, is illustrated in Fig. 2, where FP are represented by the blue squares while SP are represented by green circles.

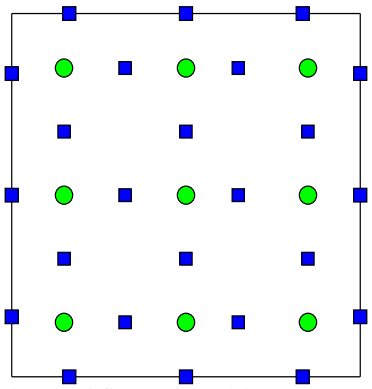


Figura 2: Set of solution points (green circles) and flux points (blue squares) distribution for the 3rd-order SD method.

The number of points in a cell is determined by the order of the interpolating polynomial required to achieve the desired accuracy. For a method of order p, p^2 SPs are required, such that, in each direction, there are p points and a $p-1$ degree polynomial can be reconstructed. The SPs are chosen to be the Gauss-Legendre points, which are defined as the roots of the Legendre polynomial of order p, defined by a recursive formula,

$$P_p(\xi) = \frac{2p-1}{p}(2\xi - 1)P_{p-1}(\xi) - \frac{p-1}{p}P_{pC}(\xi) \tag{9}$$

shifted from the interval $[-1, 1]$ to $[0, 1]$.

In order to preserve the solution accuracy, polynomials of degree n are used to interpolate the fluxes and, hence, $p+1$ flux points are selected to be the Gauss-Lobatto points, defined by the roots of the Legendre polynomial of order $p-1$ plus the end points of the interval, similarly shifted to suit the $[0, 1]$ interval.

Using the solution at p solution points, a $(p-1)$ degree polynomial can be built using the following Lagrange basis defined as

$$g_i(\xi) = \prod_{s=1,\, s\neq i}^{p} \left(\frac{\xi - \xi_s}{\xi_i - \xi_s} \right) . \tag{10}$$

Similarly, using the flux values at $(p+1)$ flux points, a p degree polynomial can be built for the flux using a similar Lagrange basis defined as

$$l_{i+\frac{1}{2}}(\xi) = \prod_{s=0,\, s\neq i}^{p} \left(\frac{\xi - \xi_{s+\frac{1}{2}}}{\xi_{i+\frac{1}{2}} - \xi_{s+\frac{1}{2}}} \right) . \tag{11}$$

The reconstructed solution for the conserved variables in the standard cell is the tensor product of the two one-dimensional polynomial,

$$Q(\xi,\eta) = \sum_{i=1}^{p}\sum_{j=1}^{p} \frac{\tilde{Q}_{i,j}}{|J_{i,j}|}\, g_i(\xi)\cdot g_j(\eta)\,. \tag{12}$$

Similarly, the reconstructed flux polynomials take the following form

$$\tilde{E}(\xi,\eta) = \sum_{i=0}^{p}\sum_{j=1}^{p} \tilde{E}_{i+\frac{1}{2},j}\cdot l_{i+\frac{1}{2}}(\xi)\cdot g_j(\eta), \tag{13}$$

$$\tilde{F}(\xi,\eta) = \sum_{i=1}^{p}\sum_{j=0}^{p} \tilde{F}_{i,j+\frac{1}{2}}\cdot g_i(\xi)\cdot l_{j+\frac{1}{2}}(\eta). \tag{14}$$

Flux Reconstruction

The reconstructed fluxes are continuous within the cell, but discontinuous across cell interfaces. Figure 3 depicts an example of two different reconstructed solutions that need to be associated with one flux at their interface. The u^- represents the solution from the left side cell whereas u^+ represents the solution from the right cell.

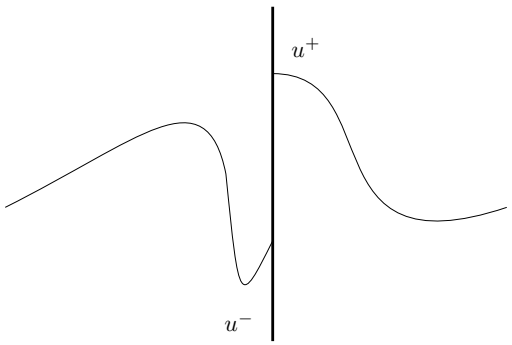


Figura 3: Discontinuous fluxes across an interface

For the inviscid fluxes, numerical flux is necessary in order to compute a common flux at interfaces to ensure conservation and stability. The Roe approximate Riemann solver detailed at Roe (1981) is considered in the present work. This numerical flux introduces artificial dissipation and upwind characteristics to the method which renders the numerical method stable. In order to compute the inviscid flux derivatives, the SD method computes the conservative variables at the flux points. The inviscid fluxes are, then, updated followed by the numerical flux calculations. Hence, the polynomial interpolation for the fluxes can be constructed. Afterwards, the derivatives of the fluxes are computed at the solution points using the derivatives of the Lagrange operators, l, as

$$\frac{\partial\tilde{E}}{\partial\xi} = \sum_{i=0}^{p}\sum_{j=1}^{p} \tilde{E}_{i+\frac{1}{2},j}\cdot l'_{i+\frac{1}{2}}(\xi)\cdot g_j(\eta), \tag{15}$$

$$\frac{\partial\tilde{F}}{\partial\eta} = \sum_{i=1}^{p}\sum_{j=0}^{p} \tilde{F}_{i,j+\frac{1}{2}}\cdot g_i(\xi)\cdot l'_{j+\frac{1}{2}}(\eta). \tag{16}$$

Time Discretization

Once the equations are discretized, an implicit marching method ought to be implemented in order to reach a reasonable convergence rate. The method used to solve the system was the generalized minimum residual (GMRES) algorithm, developed by Saad and Schultz (1986). In order to improve convergence, the incomplete LU factorization preconditioner with zero fill-in, ILU(0), was applied. The GMRES solver from the PETSc library (Balay *et al.*, 2015a,b) is used in this work.

High-Order Boundary Treatment

In order to render the high-order reconstruction process manageable, it is of great importance to consider curved meshes which better represent more complex geometries. This also has a side effect of reducing the required number of

degrees of freedom in the domain. Furthermore, the use of piecewise linear approximations of a curved boundary leads to imprecise solutions when using high-order schemes, as observed by Bassi and Rebay (1997b). Therefore, it is mandatory to use a precise description of the curved geometry of the boundary so as to obtain meaningful and accurate results. The point is that performing a correct boundary treatment ought to lead to more efficient results than refining the grid on its own as will be discussed in Section 7..

The approach used to generate the high-order mesh in the present paper consists of four subsequent steps, which can be summarized as creating the boundary representation of the model, generating a linear coarse mesh upon it, defining new nodes for each domain cell and, ultimately, projecting these new nodes onto the model geometry boundaries. This approach is considered an a-posteriori procedure as suggested in Peiró *et al.* (2013) and it overcomes some of the obstacles of directly generating a high-order mesh whilst taking advantage of the robustness of current linear mesh generators (Wang *et al.*, 2013). Although interior faces are not deformed, nodes are still properly placed so as to obtain a quadratic transformation within the whole domain.

A commonly used geometric entity that describes geometries in 2-D is the Non-Rational B-Spline (NURBS) curve. The procedures undertaken in this paper assume that geometries are generated through this type of B-spline and exported using the IGES (Initial Graphics Exchange Specification) (Gruttke, 1995) standard file exchange format. The IGES file format provides the necessary information to reconstruct the B-splines (de Boor, 2001) that recovers the geometry. The Rational B-spline curve definition is represented by Eq. (17) as follows:

$$B(u) = \frac{\sum_{i=0}^{n} w_i P_i N_{i,k}}{\sum_{i=0}^{n} w_i N_{i,k}} \,, \tag{17}$$

where u is the parameter value of the B-spline, that varies in an interval of 0 to 100% of the length of the curve. The fact that B-splines are defined as a function of its length parameter will be crucial on the node projection step. The weights w_i are associated with each control point, defined by P_i. An arbitrary amount of $n+1$ control points are used to describe the informations of a B-spline of degree k. The basis functions, $N_{i,k}$, used in the construction of the B-spline, use the information of a knot vector T_i to be recursively defined by the Cox-de Boor formula (de Boor, 2001):

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \leq u < u_{i+1} \\ 0, & \text{elsewhere} \end{cases} \tag{18}$$

$$N_{i,k}(u) = \left(\frac{u - T_i}{T_{i+k} - T_i} \right) N_{i,k-1}(u) + \left(\frac{T_{i+k+1} - u}{T_{i+k+1} - T_{i+1}} \right) N_{i+1,k-1}(u), \tag{19}$$

A file interpreter of standard IGES format was developed so that it could automate the process of acquiring the aforementioned key parameters of the B-spline, i.e., the control points, weights and the node vector. Afterwards, the parameters of the geometry were obtained from the file and the B-splines were rebuilt using Equation (17). Tests were carried out to certify that reconstruction took place correctly, i.e., in such a way that the error in this approach was acceptable as compared to other visualization tools. To do this, the original points of the geometry and the new points obtained were compared and the error in the coordinates were found to be around the same order of magnitude as the visualization tolerance from the CAD software.

Once the parametrized B-spline is reconstructed, a mesh on a CFD General Notation System (CGNS) standard format (Poirier *et al.*, 1998) is brought together with the geometry information. This parametrized B-spline will certainly cross the boundary nodes of the mesh that is, by construction, fitted over the geometry. The boundary nodes of the cells are easily computed from the mesh data structure and, thus, it can be compared to the B-spline with the aim of finding the parameter associated to that node. This has to be done using an iterative root finding method since it is not straightforward to find u given x and y. The bisection method was chosen due to its simplicity and robustness. Ideally, a single B-spline should cross all the boundary nodes. The algorithm becomes complicated when different B-splines belong to different nodes of the same cell. However, this work handles more than one B-spline within one cell as long as they follow a sequence.

The position of the new nodes can, then, be calculated so as to satisfy the required order of reconstruction in such a way that instability issues are handled properly. If the parameters are properly found, zeros of the Jacobi polynomials are used as the parameter to position the new nodes on the curved edge so as to obtain a stable positioning of nodes (Hesthaven, 1998). Figure 4 gives an example of the projection step for a second order triangular element. Note that edges located within the fluid region are not curved, and yet new nodes are placed on them. The exact same process can be applied to a quadrilateral element.

Once these new nodes are constructed, a new connectivity table can, then, be written in such a way that it respects the standard formats that handles high-order mesh. The open source GMSH software offers a relatively new *de facto* standard reference manual (Geuzaine and Remacle, 2009) that handles curved mesh specification. The way the output is written in the present work is based on this standard. In particular, the output is written as an unstructured mesh and the data structure that contains it has information about nodes, faces, neighboring and cell indexes.

Second order meshes are obtained for both triangular and quadrilateral elements thus far. The idea is to extend this capability to any general order by the end of this work. A coarse quadrilateral mesh on a cylinder is depicted in Figs. 5(a)

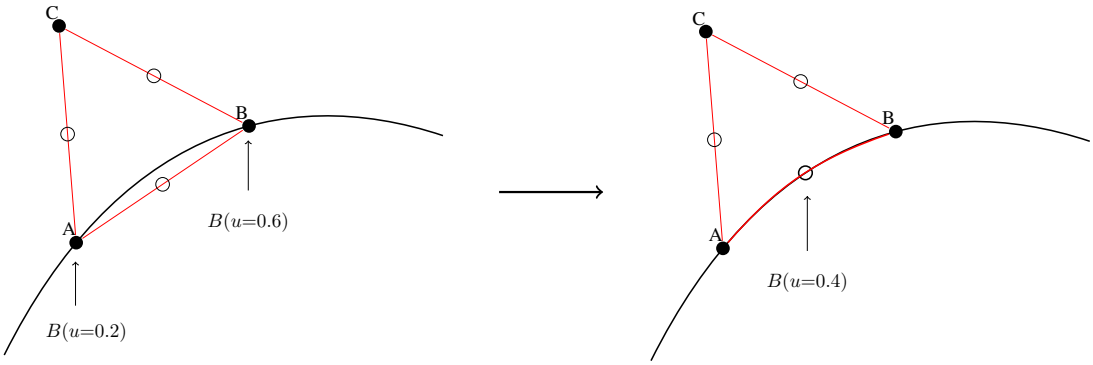
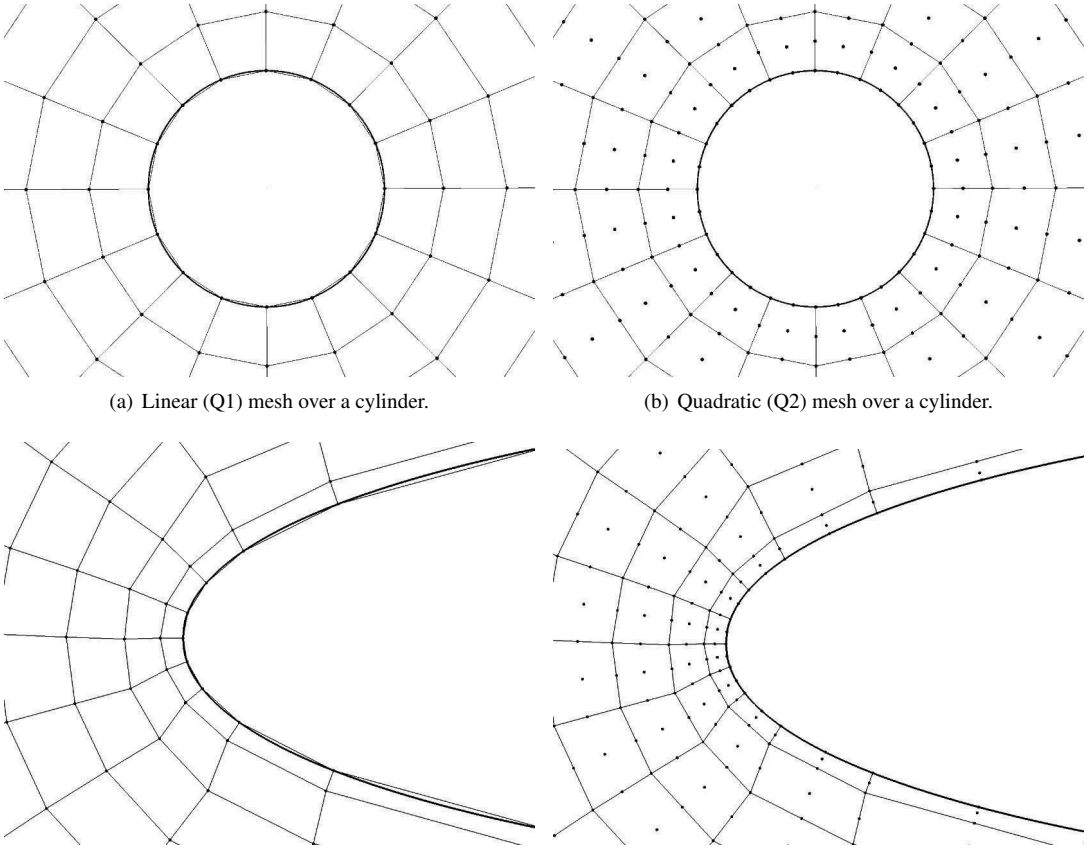


Figura 4: Node projection step into the B-spline

and 5(b) in order to illustrate the effect of node projection onto the geometry. Figure 5(c) shows a linear mesh example for the NACA 0012 airfoil, whereas Fig. 5(d) presents a quadratic mesh for the same geometry.

Note that the curved cells considered in this work belong to the family of Lagrangian elements, for which points inside the cell are used to obtain a complete transformation from physical to parameter space. For the following results, the notation Qp is used to determined the order of the mesh, where p represents the order of the polynomial in which the faces of the mesh are described and, hence, $Q1$ refers to linear meshes while $Q2$ is related to meshes composed of quadratic elements. Moreover, the number of degrees of freedom in a SD cell is the number of SPs existent in it (previously defined as $(p+1)^2$) times the numbers of variables of the problem. The total number of DoF in the solution follows this rationale for the whole computational mesh. A higher order mesh affects the coordinate transformation process which is performed once as a pre-processing step of the simulation.



(a) Linear (Q1) mesh over a cylinder.

(b) Quadratic (Q2) mesh over a cylinder.

(c) Linear (Q1) mesh over a NACA 0012 airfoil configuration.

(d) Quadratic (Q2) mesh over a NACA 0012 airfoil configuration.

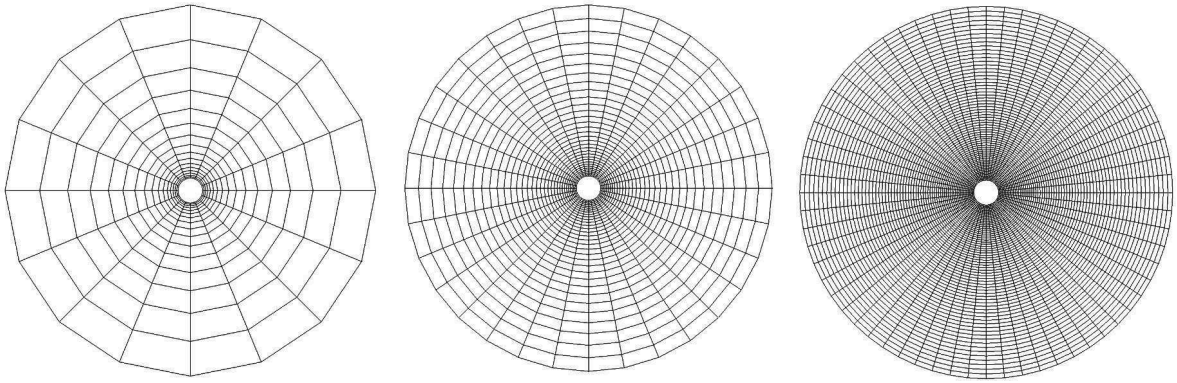
Figura 5: Curved mesh generation process. The thick line is the targeted geometry where boundary nodes are projected.

Results

Circular Cylinder

An important validation for the curved meshes is the case of a subsonic inviscid flow over a circular cylinder. A flow with freestream Mach number of 0.2 is simulated and the solution for the Euler equations in this domain is expected to be

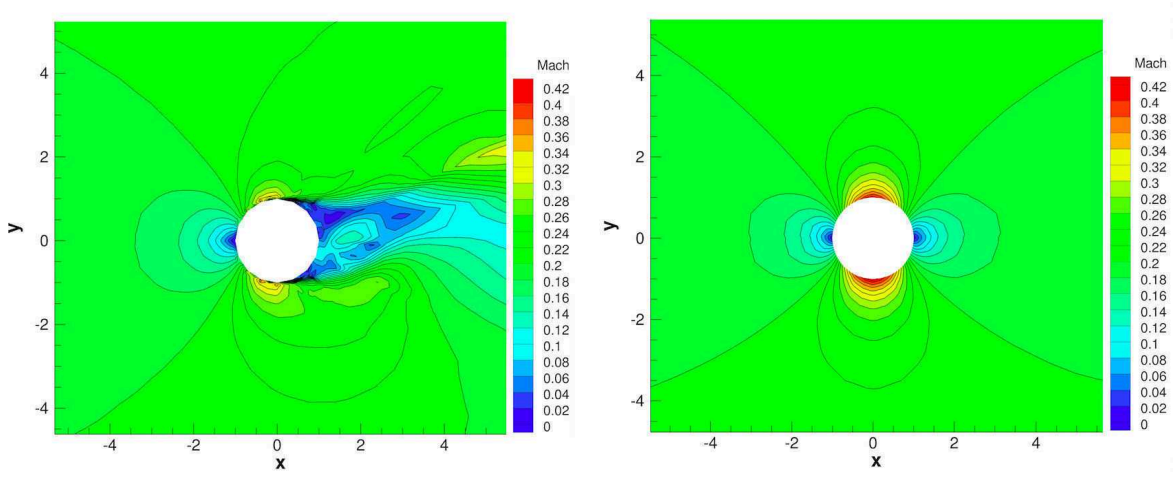
twice symmetrical, given that compressibility effects are small for this freestream Mach number. This test demonstrates that spurious results can be observed for high curvature geometries, even though a continued sequential refinement is performed over the linear mesh. The correct representation of the geometry using higher order meshes, however, handles this effect properly. A quadrilateral linear mesh composed by 16 cells in both radial and azimuthal directions (16×16) is defined. Afterwards, even finer meshes with 32×32 and 64×64 cells are considered to assess the effects of using 2nd-order meshes. These meshes can be observed in Fig. 6. The resulting $Q2$ curved mesh is shown in Fig. 5(b) of Section 6. The outer boundary of the mesh is located at a distance of 8-diameter units. The following results are obtained



(a) 16×16. (b) 32×32. (c) 64×64.

Figura 6: Meshes for the cylinder configuration.

with a 3rd-order method, in which the solution is reconstructed by a 2nd-order polynomial (P2). In this sense, the order of the polynomial that reconstructs the mesh is compatible with the one that reconstructs the solution. The results in Fig. 7 show that a linear mesh exposes the flow to kinks in the geometry in such a way that the flow can separate, even though an inviscid formulation is used for the present computations. Hence, in the Q1P2 case, the order of the polynomial used to reconstruct the solution is higher than the one used to reconstruct the geometry and, furthermore, the geometry is linear. Therefore, lower-order erroneous solutions build-up from the coarse geometry representation. When a curved, quadratic mesh representation is considered, the results are clearly symmetric and the numerical oscillations are effectively handled, *i.e.*, they are eliminated. The second test uses on a finer mesh, *i.e.*, the 32×32 mesh shown in Fig. 6(b), in order to clarify



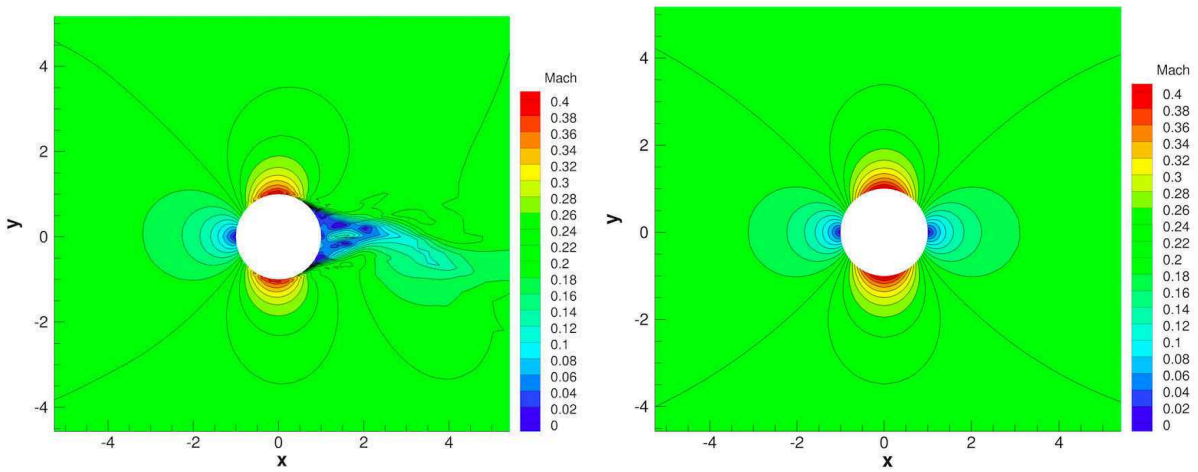
(a) Q1P2. (b) Q2P2.

Figura 7: Mach contours for a 16×16 mesh using a 3rd-order SD method (P2) considering both linear (Q1) and quadratic (Q2) meshes

the fact that sometimes it is not enough to refine a linear mesh to properly represent a highly curved geometry. The solution observed in Fig. 8 shows that spurious solutions do not vanish for a finer linear mesh, as opposed to what happens for the 2nd-order mesh. To take this analysis even further, an even finer mesh of 64×64 cells is created, as indicated in Fig. 6(c). Even though the solution in Fig. 9 is more symmetric in a top-to-bottom comparison, results for the linear mesh are highly asymmetric in the horizontal direction. However, the results for the curved mesh remain consistent to what is expected for such flow conditions and geometry.

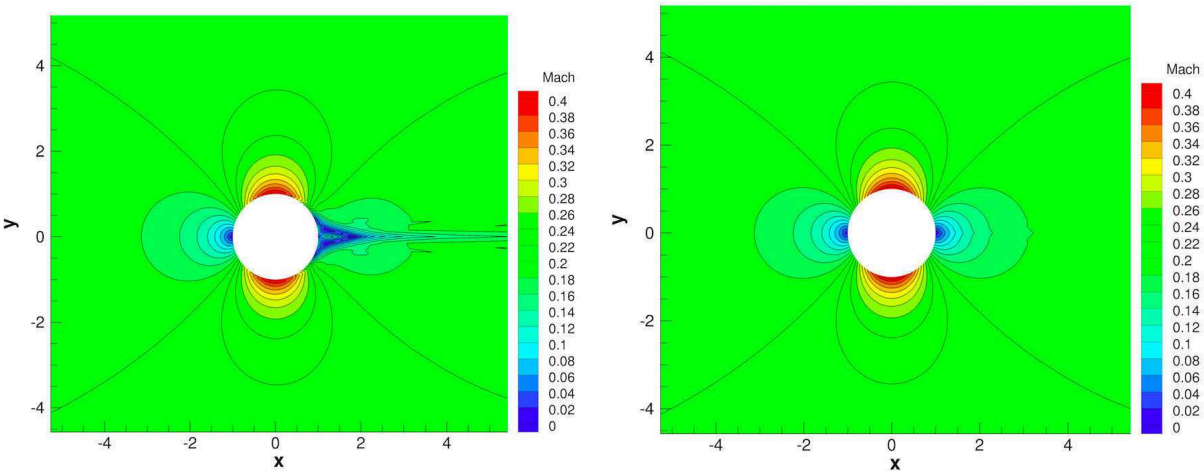
NACA 0012 Airfoil

An inviscid subsonic NACA 0012 airfoil flow simulation is performed with for 2nd- and 3rd-order SD method using the implicit time marching scheme and linear and quadratic meshes. The freestream Mach number value of $M_\infty = 0.6$



(a) Q1P2. (b) Q2P2.

Figura 8: Mach contours for a 32×32 mesh using a 3rd-order SD method (P2) considering both linear (Q1) and quadratic (Q2) meshes



(a) Q1P2. (b) Q2P2.

Figura 9: Mach contours for a 64×64 mesh using a 3rd-order SD method (P2) considering both linear (Q1) and quadratic (Q2) meshes

and $\alpha = 0$ deg. angle-of-attack are used, which ensure a smooth flow throughout the domain. The original Q1 mesh is composed of 4619 quadrilateral cells and is presented in Fig. 10. The geometry of the airfoil is modified to have a collapsed trailing edge. The farfield boundary is located at a distance of 1000 chords from the airfoil, using a *C*-type grid topology. A quadratic Q2 mesh was obtained from the original Q1 mesh. An upper limit for the CFL number of 10^{10} is used for both the 2nd- and 3rd-order simulations. Figure 11 shows the convergence history for all the orders considered for this test case. All simulations converge in nearly 30 iterations, therefore with $1/500$ of the number of iterations required for convergence of the simulations with an explicit time marching scheme, as discussed in Moreira *et al.* (2015). In the 3rd-order (p2) simulation with quadratic mesh (q2), each iteration for the implicit scheme lasts from 5 to 10 seconds depending on how much Krylov sub-iterations were made in the GMRES. Therefore, the complete simulation finishes in less than 5 minutes, running in serial mode for a single CPU.

Figure 12 shows the density contours for the NACA 0012 subsonic simulations. The flow property contours share the same color map range and number of levels. Hence, a qualitative comparison can be made from the solutions at different orders of accuracy. There are some small differences between the 3rd-order solution and the 2nd-order one. In particular, the 3rd-order method solution achieves slightly smoother contour lines than the results obtained with the 2nd-order method. An important measure of the enhancement of the results due to curved meshes are the magnitude of entropy error generated at the surface of the airfoil. For the Euler subsonic simulation, any entropy result observed is caused by spurious solutions introduced into the domain. By means of comparison, we may observe in Fig. 13 that the process of increasing the order of mesh representation has effectively reduced the entropy error along the airfoil by over an order of magnitude. The entropy error for the linear mesh increases while the entropy generated by the curved wall is reduced accordingly. The logarithmic scale results in a much more visible effect of a change in entropy when it is small, which explains the highly oscillatory aspect of the curved mesh results. Essentially, both cases oscillates in the same order of magnitude at the trailing edge. It is also important to notice that, for the case of third order methods (P2), a second order curved mesh (Q2) is compatible in terms of polynomial order. For a fourth-order method (P3), however, a

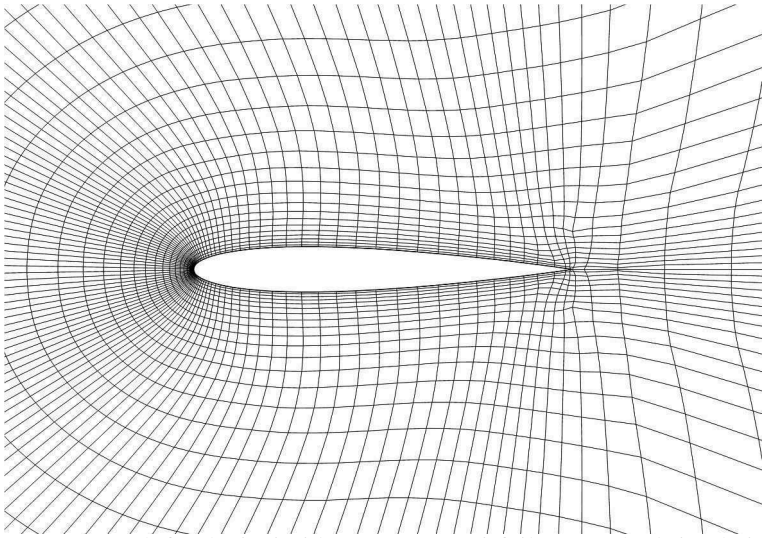


Figura 10: Mesh for the inviscid NACA 0012 airfoil SD method simulation.

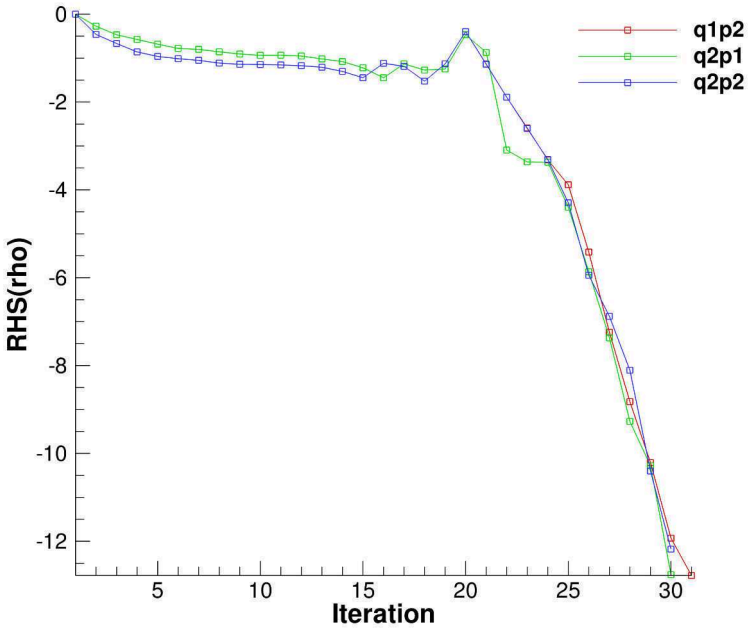
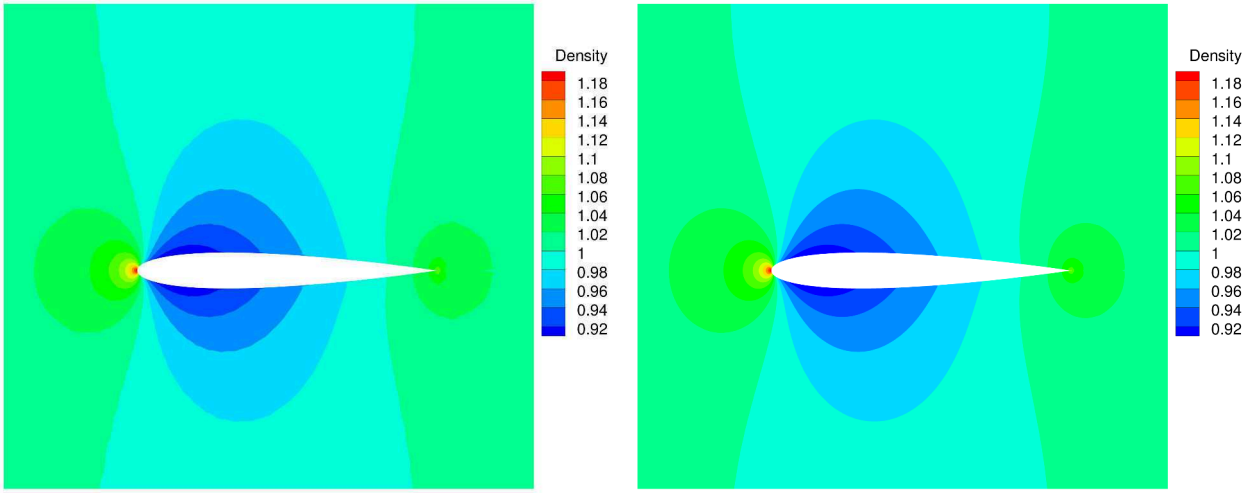


Figura 11: NACA 0012 convergence history.



(a) 2nd-order. (b) 3rd-order.

Figura 12: Density contours for the NACA 0012 airfoil inviscid simulations at $M_\infty = 0.6$ and $\alpha = 0$ deg.

third order mesh (Q3) is necessary so as to capture the whole advantages of the curved boundary treatment. The fact that

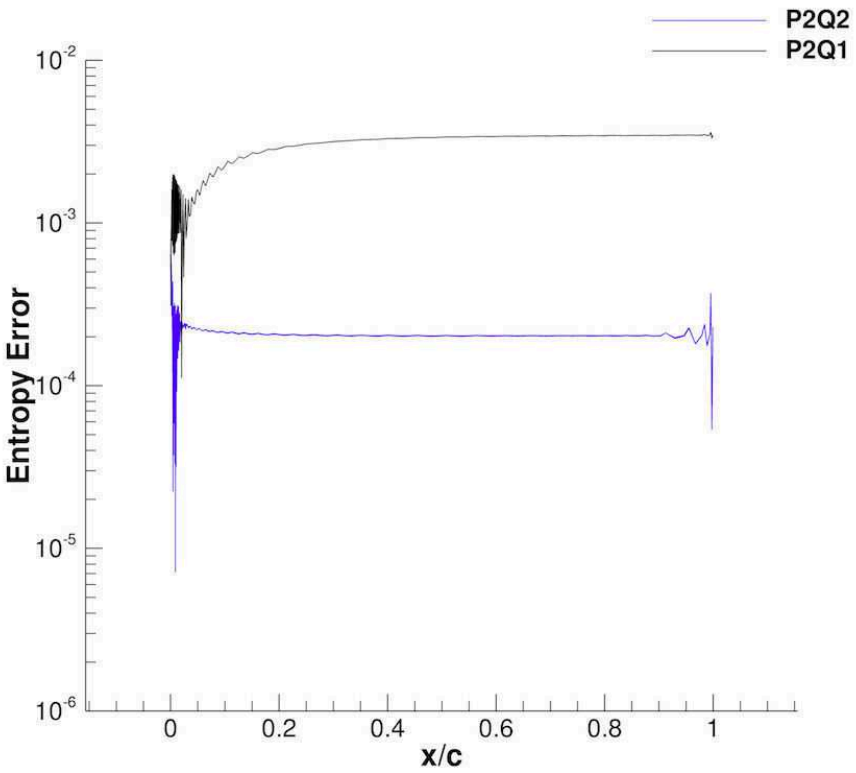


Figura 13: Comparison of entropy error along the surface of a NACA 0012 airfoil for the inviscid 2nd-order SD method. Both linear (P2Q1) and quadratic (P2Q2) meshes are considered.

linear meshes establish spurious results into the solution leads to another important and practical observation which is the behavior of the pressure coefficient Cp. The results for the linear Q1 mesh on Fig. 14 depicts an oscillatory aspect for the curve whereas this oscillation is controlled with the use of curved Q2 mesh. This can be understood considering the fact that linear meshes are perceived as corners to the flow. The results are clearly behaving in such a way that the fluid encounters repeatedly compression and expansion corners, which leads to an alternating pattern, in such a way that, on average, the result is correct. The better suited representation of the Q2 mesh reduces the entropy generated by the wall. Moreover, and more importantly, it fixes the spurious oscillatory results at the wall.

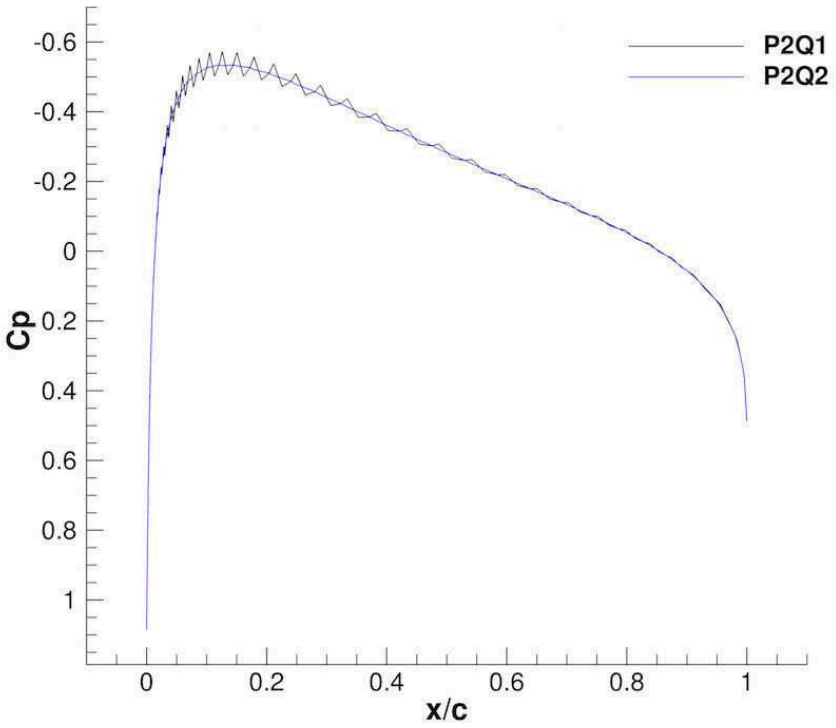


Figura 14: Pressure coefficient at Solution Points along the wall of a NACA 0012 airfoil for inviscid simulations at $M_\infty = 0.6$ and $\alpha = 0$ deg.

Conclusions

The present work addresses a new technique that was developed to manipulate linear meshes and to generate high-order curved computational grids. The technique has proved itself reliable in the sense that, once the geometry is obtained, it can effectively project the mesh nodes onto an exact description of the model. In addition, the curved meshes are written in a general GMSH standard format, which gives them an important aspect of portability. The results obtained for the cylinder configuration shows that oscillations may occur if the geometry is not properly represented even in the case of well refined meshes. The results for NACA 0012 airfoil configuration shows that, although entropy generated by the linear wall may seem negligible, the aerodynamic coefficients may be strongly affected. The curved meshes, as expected, reduce spurious results and lead to more consistent solutions.

On the other hand, efforts are being dedicated to enhance the robustness of the method, since complex geometries are still difficult to handle. Another restriction of this study lies on the fact that only second order meshes were obtained, which, for practical purposes, limits proper simulation up to only third order methods. In order to overcome this restrictions, ongoing developments are being carried out in which the main idea is to generalize the methodology for increasing order of meshes to continue to obtain practical results. Lastly, moving mesh techniques shall be used to avoid curve or surface overlapping for highly stretched meshes.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support for the present research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grants No. 309985/2013-7, No. 400844/2014-1 and No. 443839/2014-0. The work is also supported by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under Research Grant No. 2013/07375-0. The support provided by Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, through graduate scholarships for all graduate student authors, is also greatly appreciated.

REFERENCES

Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zampini, S. and Zhang, H., 2015a. "PETSc Web page". http://www.mcs.anl.gov/petsc. URL http://www.mcs.anl.gov/petsc.

Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zampini, S. and Zhang, H., 2015b. "PETSc users manual". Technical Report ANL-95/11 - Revision 3.6, Argonne National Laboratory. URL http://www.mcs.anl.gov/petsc.

Bassi, F. and Rebay, S., 1997a. "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations". *Journal of Computational Physics*, Vol. 131, pp. 267–279.

Bassi, F. and Rebay, S., 1997b. "High-order accurate discontinuous finite element solution of the 2D Euler equations". *Journal of Computational Physics*, Vol. 138, No. 2, pp. 251–285.

de Boor, C., 2001. *A Practical Guide to Splines*. Springer-Verlag, New York.

Geuzaine, C. and Remacle, J., 2009. "GMSH: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities". *International Journal for Numerical Methods in Engineering*, Vol. 79, No. 11, pp. 1309–1331.

Gruttke, W.B., 1995. *The Initial Graphics Exchange Specification (IGES) v. 6.0*. IGES/PDES Organization.

Hesthaven, J.S., 1998. "From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex". *SIAM Journal on Numerical Analysis*, Vol. 35, pp. 655–676.

Liu, Y., Vinokur, M. and Wang, Z.J., 2006. "Spectral difference method for unstructured grids I: Basic formulation". *Journal of Computational Physics*, Vol. 216, pp. 780–801.

May, G. and Jameson, A., 2006. "A spectral difference method for the Euler and Navier-Stokes equations on unstructured meshes". In AIAA Paper No. 2006-0304, *Proceedings of the 44th AIAA Aerospace Sciences Meeting*. Reno, NV.

Moreira, F.M., , Breviglieri, E.J.C., Aguiar, A.R.B. and Azevedo, J.L.F., 2015. "High-order compressible flows simulations with the unstructured spectral difference method". In AIAA Paper No. 2015-3195*, Proceedings of the 22nd AIAA Computational Fluid Dynamics Conference*. Dallas, TX.

Peiró, A.G., Roca, X., Peraire, J. and Sarrate, J., 2013. "High-order generation on CAD geometries". In *Proceedings of the VI International Conference on Adaptive Modeling and Simulation*. Lisbon, Portugal.

Poirier, D., Allmaras, S.R., McCarthy, D.R., Smith, M.F. and Enomoto, F.Y., 1998. "The CGNS system". In AIAA Paper No. 98-3007, *Proceedings of the 29th AIAA Fluid Dynamics Conference*. Albuquerque, NM.

Roe, P.L., 1981. "Approximatte Riemann solvers, parameter vectors, and difference schemes". *Journal of Computational Physics*, Vol. 43, No. 2, pp. 357–372.

Saad, Y. and Schultz, M.H., 1986. "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems". *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, pp. 856–869.

Wang, Z.J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., Kroll, N., May, G., Persson, P., van Leer, B. and Visbal, M., 2013. "High-order CFD methods: Current status and perspective". *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, pp. 811–845.

Wang, Z.J., Liu, Y., May, G. and Jameson, A., 2007. "Spectral difference method for unstructured grids II: Extension to the Euler equations". *Journal of Scientific Computing*, Vol. 32, No. 1, pp. 3938–3956.

RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.