6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11–15 June 2018, Glasgow, UK

VISCOUS FLOW CALCULATIONS WITH THE SPECTRAL DIFFERENCE METHOD USING HIGH-ORDER MESHES

Fábio M. MOREIRA[1†], André R. B. AGUIAR[1‡], João Luiz F. AZEVEDO[2]

[1] Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, 12228-900, Brazil
[†]fabiom91@gmail.com, [‡]ufabc.andre@gmail.com

[2] Instituto de Aeronáutica e Espaço, São José dos Campos, SP, 12228-904, Brazil
joaoluiz.azevedo@gmail.com

Key words: CFD, High-Order Methods, Spectral Differences, High-Order Meshes

Abstract. The present work is concerned with assessing different numerical viscous flux schemes that could be used to solve flux discontinuities across cell interfaces. The aim is to verify the accuracy of all schemes, evaluate their robustness and computational efficiency in term of number of iterations and time required to satisfy a convergence criterion. The methods here considered are the first and second approaches by Bassi and Rebay, the interior penalty approach and the local spectral difference approach. Variations on ways to calculate terms not defined in these methods are also considered. The numerical experiments are focused on evaluating the schemes across a range of Reynolds numbers and their coupling with limiting techniques in order to obtain solutions to transonic flows.

1 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 have been 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 due to their inherently higher levels of dissipation and slower scaling of the solution accuracy with respect to mesh refinement. Moreover, high-order methods offer the possibility to reduce simulation costs for given solution accuracy levels, when compared to low-order schemes.

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 time integration, 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 high-order methods, a correct description of curved boundaries is mandatory.

The objective of this work it to assess numerical diffusive flux schemes and evaluate the impact of high-order boundaries in these simulations. The approach used was originally developed for the discontinuous Galerkin (DG) method by Bassi and Rebay [1]. Numerical experiments are performed on a laminar flow over a NACA0012 airfoil for validation. Furthermore, the Radial Basis Function (RBF) technique used to generate the curved meshes is tested for mesh movement for a heaving and pitching airfoil problem.

2 SPECTRAL DIFFERENCE METHOD FORMULATION

The flows of interest in the present work are assumed to be adequately modeled by the 2-D Navier-Stokes (NS) equations. These equations can be written in differential form as

$$\frac{\partial Q}{\partial t} + \frac{\partial E_c}{\partial x} + \frac{\partial F_c}{\partial y} = \frac{\partial E_d}{\partial x} + \frac{\partial F_d}{\partial y} \ . \tag{1}$$

The vector of conserved variables, Q, and the convective flux vectors, E_c and F_c, are given by

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{Bmatrix} \ , \qquad E_c = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{Bmatrix} \quad \text{and} \quad F_c = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{Bmatrix} \ . \tag{2}$$

Here, ρ is the density, u and v are the velocity components, e is the total energy per unit of volume and p is the pressure. The viscous flux vectors, E_d and F_d, are defined as

$$\begin{aligned} E_d &= \mu \begin{Bmatrix} 0 \\ 2\frac{\partial u}{\partial x} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ u[2\frac{\partial u}{\partial x} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})] + v(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \frac{C_p}{P_r}\frac{\partial T}{\partial x} \end{Bmatrix} \ , \\ F_d &= \mu \begin{Bmatrix} 0 \\ \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \\ 2\frac{\partial v}{\partial y} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \\ v[2\frac{\partial v}{\partial y} - \frac{2}{3}(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})] + u(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}) + \frac{C_p}{P_r}\frac{\partial T}{\partial y} \end{Bmatrix} \ , \end{aligned} \tag{3}$$

where μ is the dynamic viscosity coefficient, c_p is the specific heat at constant pressure and P_r stands for the Prandtl number of the fluid. T is the local temperature of the flow. The system is closed with the state equation for perfect gases and the ratio of specific heats, γ, is set as 1.4 for all computations in this work.

The spectral difference (SD) method employs a finite difference-like approach. First, in order to achieve an efficient implementation, all cells in the physical domain, (x, y), are

transformed into unit square elements in the computational domain. Such transformation can be written as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{s=1}^{K} M_s(\xi,\eta) \begin{pmatrix} x_s \\ y_s \end{pmatrix} , \tag{4}$$

where K is the number of points used to define the physical element, (x_s, y_s) are the Cartesian coordinates of such 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 is bilinear and the analytic expression can be easily found. However, in the case of higher order meshes, the number of points used to define a single cell increases. Considering a bi-polynomial representation, the transformation parameters can be calculated by numerically solving a linear system of size K.

The metric terms and the Jacobian matrix of the transformation can be computed in a pre-processing step and kept in memory in cases where the mesh is stationary. For moving mesh problems, the transformation must be calculated for each mesh position. The implementation follows the formulation presented in Refs. [2] and [3]. The governing equations in the physical domain are transformed into the computation domain and they are rewritten as

$$\frac{\partial \widetilde{Q}}{\partial t} + \frac{\partial \widetilde{E}_c}{\partial \xi} + \frac{\partial \widetilde{F}_c}{\partial \eta} = \frac{\partial \widetilde{E}_d}{\partial \xi} + \frac{\partial \widetilde{F}_d}{\partial \eta} , \tag{5}$$

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

$$J = \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix} . \tag{6}$$

For the current implementation, the flux vectors in the computational domain can be simplified from the general form as

$$\begin{aligned} \begin{pmatrix} \widetilde{E}_e - \widetilde{E}_d \\ \widetilde{F}_e - \widetilde{F}_d \end{pmatrix} &= \begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix}^{-1} \begin{pmatrix} E_c(Q) - E_d(Q,\nabla Q) \\ F_c(Q) - F_d(Q,\nabla Q) \end{pmatrix} \\ &= \begin{pmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{pmatrix} \begin{pmatrix} E_c - E_d \\ F_c - E_d \end{pmatrix} . \end{aligned} \tag{7}$$

In the standard element, two sets of points are defined, namely the solution points (SP) and the flux points (FP). As shown in Ref. [4], the stability of the method in a large array of problems is independent from the distribution of the SPs, meaning any criteria may be used in order to determine the location of these points. In the present implementation, the favored aspects are simplicity of implementation and computational efficiency. Therefore, an internal cell discretization that only requires dealing with one-dimensional problems is selected. The use of tensor products and the enforcement that the directions of the interpolations and derivatives should coincide greatly simplify the formulation of the method. An example of such distribution, for a 3rd-order SD scheme, is illustrated in Fig. 1.

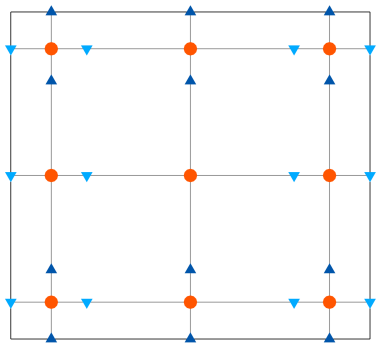


Figure 1: Possible solution point (orange circles) and flux point (blue triangles) distributions for the 3rd-order SD method.

The number of points in a cell is determined by the order of the interpolating polynomial required to achieve the desired solution accuracy. For an n-th order method, n^2 SPs are required, such that, in each direction, there are n points and an $n-1$ 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 n shifted from the interval $[-1,1]$ to $[0,1]$. In order to preserve the solution accuracy, n-degree polynomials are used to interpolate the fluxes and, hence, the $n+1$ flux points are selected to be the Legendre-Gauss-Lobatto points, defined by the roots of the Legendre polynomial of order $n-1$ plus the end points of the interval, similarly shifted to suit the $[0,1]$ interval.

Using the values at the n solution points and at the $(n+1)$ flux points, an $(n-1)$ degree solution polynomial and an n degree flux polynomial can be built using the following Lagrange basis, defined as

$$g_i(\xi) = \prod_{s=1,\, s\neq i}^{n} \left(\frac{\xi - \xi_s}{\xi_i - \xi_s} \right) \qquad \text{and} \qquad l_{i+\frac{1}{2}}(\xi) = \prod_{s=0,\, s\neq i}^{n} \left(\frac{\xi - \xi_{s+\frac{1}{2}}}{\xi_{i+\frac{1}{2}} - \xi_{s+\frac{1}{2}}} \right) \tag{8}$$

respectively. The reconstructed solution for the conserved variables in the computational domain is given by the tensor product of the two 1-D polynomials,

$$Q(\xi,\eta) = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\widetilde{Q}_{i,j}}{|J_{i,j}|}\, g_i(\xi) \cdot g_j(\eta) \ , \tag{9}$$

where $\widetilde{Q}_{i,j}$ represents the values of the conserved properties at the SPs. Similarly, the reconstructed flux polynomials take the following form

$$\widetilde{E}(\xi,\eta) = \sum_{i=0}^{n} \sum_{j=1}^{n} \widetilde{E}_{i+\frac{1}{2},j} \cdot l_{i+\frac{1}{2}}(\xi) \cdot g_j(\eta) \ , \ \widetilde{F}(\xi,\eta) = \sum_{i=1}^{n} \sum_{j=0}^{n} \widetilde{F}_{i,j+\frac{1}{2}} \cdot g_i(\xi) \cdot l_{j+\frac{1}{2}}(\eta) \ . \tag{10}$$

where $\widetilde{E}_{i+\frac{1}{2},j}$ and $\widetilde{F}_{i,j+\frac{1}{2}}$ are the values of the flux vectors at the corresponding FPs.

The reconstructed variables are element-wise continuous, but discontinuous across cell interfaces. On the internal FPs, the convective flux can be directly calculated from the conserved variables interpolated from the SPs. However, in order to ensure stability and conservation, a common numerical flux must be determined for both neighboring cells at their interface FPs. This can be accomplished by the use of a Riemann solver. In the present work, the Roe approximate Riemann solver is used as numerical flux function for the inviscid fluxes. There is a similar problem in determining a common viscous flux at cell interfaces which will be described in the following section.

Once the polynomial interpolation for the fluxes has been constructed, the derivatives of the fluxes are computed at the solution points using the derivatives of the Lagrange operators, l, as

$$\frac{\partial \widetilde{E}}{\partial \xi} = \sum_{i=0}^{n} \sum_{j=1}^{n} \widetilde{E}_{i+\frac{1}{2},j} \cdot l'_{i+\frac{1}{2}}(\xi) \cdot g_j(\eta) \ , \qquad \frac{\partial \widetilde{F}}{\partial \eta} = \sum_{i=1}^{n} \sum_{j=0}^{n} \widetilde{F}_{i,j+\frac{1}{2}} \cdot g_i(\xi) \cdot l'_{j+\frac{1}{2}}(\eta). \quad (11)$$

With both inviscid fluxes uniquely defined at all FPs and interfaces, the flux derivatives are computed at the solution points using the above described derivatives of the Lagrange operators. With the gradients of the fluxes calculated on the SPs, a time-stepping method can be invoked. The scheme implemented here is the explicit 2nd-order, 3-stage optimal strong stability preserving Runge-Kutta scheme described in Ref. [5].

Another important aspect of high-order methods is the representation of the boundary elements. The numerical tool developed for the present work considers a linear 2-D mesh and its geometry description as inputs provided by the user. A quadratic boundary representation of such input mesh is, then, created for the high-order simulations. This procedure allows one to accurately represent a generic 2-D geometry along with an unstructured domain discretization. Such features are necessary in order to make the high-order scheme performance comparable, or superior, to that of low-order methods by allowing coarser high-order meshes to be considered. Reference [6] has shown the influence of the high-order boundary treatment in 2-D inviscid simulations as it decreases the entropy error and improves the pressure distribution along the boundaries.

3 VISCOUS FLUX DISCRETIZATION

Just as it is required for the convective fluxes to be continuous across cell interfaces, the same condition must be enforced on the viscous fluxes in order to maintain conservation. However, unlike the inviscid fluxes, the viscous fluxes are functions not only of the solution but also of its gradient at a given point. Therefore, for the viscous fluxes to be equal on both sides of a face, both variables and their gradients need to be uniquely defined at cell interfaces.

There are several approaches to obtain these common properties and gradients, most of which originated in a discontinuous Galerkin context. An overview in that same DG context is given by Arnold [7]. More recently, Kannan [8] and Van den Abeele [9, 10] have adapted some of these methods for the spectral finite volume and the spectral difference methods. Here, we will focus on the second approach by Bassi and Rebay (BR2) [1].

Firstly, it must be noted that the derivatives of the primitive properties, necessary to calculate the viscous flux, cannot be calculated directly because the high-order polynomial reconstruction is only valid for the conserved variables. Using basic derivation rules, the derivatives of the primitive variables can be expressed from the derivatives of the conserved variables as follows

$$u = \frac{\rho u}{\rho} \qquad \Longrightarrow \qquad \vec{\nabla} u = \frac{\rho \vec{\nabla} \rho u - \rho u \vec{\nabla} \rho}{\rho^2} = \frac{\vec{\nabla} \rho u - u \vec{\nabla} \rho}{\rho} \tag{12}$$

$$v = \frac{\rho v}{\rho} \qquad \Longrightarrow \qquad \vec{\nabla} v = \frac{\rho \vec{\nabla} \rho v - \rho v \vec{\nabla} \rho}{\rho^2} = \frac{\vec{\nabla} \rho v - v \vec{\nabla} \rho}{\rho} \tag{13}$$

$$T = \frac{\gamma - 1}{R} \left(\frac{e}{\rho} - \frac{u^2 + v^2}{2} \right) \quad \Longrightarrow \quad \vec{\nabla} T = \frac{\gamma - 1}{R} \left(\frac{\rho \vec{\nabla} e - e \vec{\nabla} \rho}{\rho^2} - u \vec{\nabla} u - v \vec{\nabla} v \right) \tag{14}$$

Depending on the diffusive flux approach, the derivatives of the conserved variables might be calculated in two different ways. Both of them rely on using the Lagrange derivative coefficients, the same ones used to calculate the flux derivatives, to calculate the derivatives of the conserved variables. However, they differ on the values of the conserved variables from which they calculate the derivatives. The first approach calculates the gradient based on the values of the conserved properties at the FPs of the current cell, noted as Q, and they will be called $\vec{\nabla} Q$. the gradient can be, then, calculated as

$$\vec{\nabla} Q = \begin{bmatrix} \dfrac{\partial Q}{\partial \xi} \\ \dfrac{\partial Q}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{i=0}^{n} \sum_{j=1}^{n} Q_{i+\frac{1}{2},j} \cdot l'_{i+\frac{1}{2}}(\xi) \cdot g_j(\eta) \\ \displaystyle\sum_{i=1}^{n} \sum_{j=0}^{n} Q_{i,j+\frac{1}{2}} \cdot g_i(\xi) \cdot l'_{j+\frac{1}{2}}(\eta) \end{bmatrix} . \tag{15}$$

A second method calculates the gradient using the same coefficients, but replacing Q with $\widehat{Q}$ at face FPs, where $\widehat{Q}$ is the average of the vector of conserved properties at the interface. This new solution polynomial is called $\breve{Q}$ and, therefore, its gradient will be referred as $\vec{\nabla} \breve{Q}$. The gradient can be written as

$$\vec{\nabla} \breve{Q} = \begin{bmatrix} \dfrac{\partial \breve{Q}}{\partial \xi} \\ \dfrac{\partial \breve{Q}}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{i=0}^{n} \sum_{j=1}^{n} \breve{Q}_{i+\frac{1}{2},j} \cdot l'_{i+\frac{1}{2}}(\xi) \cdot g_j(\eta) \\ \displaystyle\sum_{i=1}^{n} \sum_{j=0}^{n} \breve{Q}_{i,j+\frac{1}{2}} \cdot g_i(\xi) \cdot l'_{j+\frac{1}{2}}(\eta) \end{bmatrix} . \tag{16}$$

The main difference between these two methods is that the first one results in a compact viscous flux calculation, while neighboring cell solutions are required to calculate the local gradients with the second approach. Furthermore, the averaging of the solutions does not guarantee the continuity of the gradients across cell interfaces. Therefore, gradients must be subjected to another procedure to ensure they are uniquely defined. Hence, the

common gradient at one face depends on the gradients calculated at the left and right cells, which might also depend on all the neighbors of these cells, generating an inconveniently large stencil for parallel computation. At internal FPs, however, these issues do not arise and any one of the two methods might be used at a similar computational cost.

In the BR2 approach, first presented by Bassi and Rebay in Ref. [1], the value of the conserved variables at the interface is still calculated as a simple average of the left and right values,

$$\widehat{Q} = \frac{1}{2}\left(Q_L + Q_R\right) \ . \tag{17}$$

The common gradient, however, is evaluated as an average of corrected left and right gradients. The correction imposed to these gradients makes them equivalent to calculating them using Q on all FPs, including interface FPs, except for the FPs at the face where the gradient is being evaluated. Due to the linearity of the derivative operator, these corrected gradients can be efficiently calculated simply by adding a function $\vec{\Lambda}$ to $\vec{\nabla}Q$. Here, $\vec{\Lambda}$ is the gradient of a polynomial ΔQ defined as $Q_L - Q_R$ at the FPs on the current face and 0 on all other FPs. Finally, the common gradient can, then, be written as

$$\widehat{\vec{\nabla}Q} = \frac{(\vec{\nabla}Q_L - \vec{\Lambda}_L) + (\vec{\nabla}Q_R - \vec{\Lambda}_R)}{2}, \tag{18}$$

where correction operators are defined by the following expression

$$\vec{\Lambda} = \begin{bmatrix} \dfrac{\partial Q}{\partial \xi} \\ \dfrac{\partial Q}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{i=0}^{n}\sum_{j=1}^{n} \Delta Q_{i+\frac{1}{2},j} \cdot l'_{i+\frac{1}{2}}(\xi) \cdot g_j(\eta) \\ \displaystyle\sum_{i=1}^{n}\sum_{j=0}^{n} \Delta Q_{i,j+\frac{1}{2}} \cdot g_i(\xi) \cdot l'_{j+\frac{1}{2}}(\eta) \end{bmatrix} . \tag{19}$$

It should be noted that, because ΔQ is 0 on almost all FPs, the calculation of $\vec{\Lambda}$ consists of only one multiplication per FP in which ΔQ is not 0.

4 RESULTS

4.1 Laminar flow over a NACA 0012 airfoil

A study considering viscous flow around a NACA 0012 airfoil is performed in order to validate the results obtained from the previous methodology. An unstructured mesh composed by quadrilaterals is used to represent the airfoil, as observed in Fig. 2. The flow solution is performed at Mach number 0.6 and Reynolds number 1000, based on the airfoil chord, for an angle of attack of $\alpha = 0$ deg. The calculations are performed with with a 3rd order (P2) method. The Mach number contours for this test case are shown in Fig. 3. The results were calculated with two different meshes. Hence, Fig. 3(a) presents the Mach number contours for the calculations with a linear mesh (Q1), whereas Fig. 3(b) presents the results for the calculations with a quadratic mesh (Q2). The basic features of the flow are clearly captured, namely the development of the boundary layer over the airfoil, followed by a wake behind it.

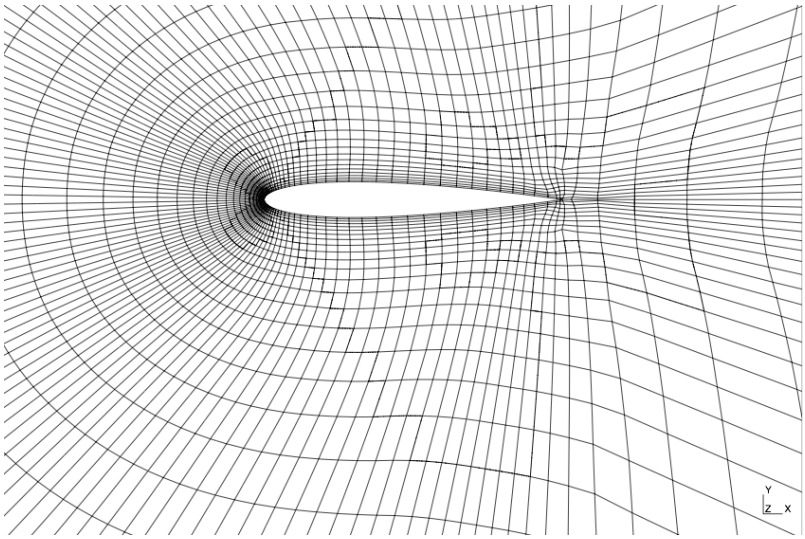
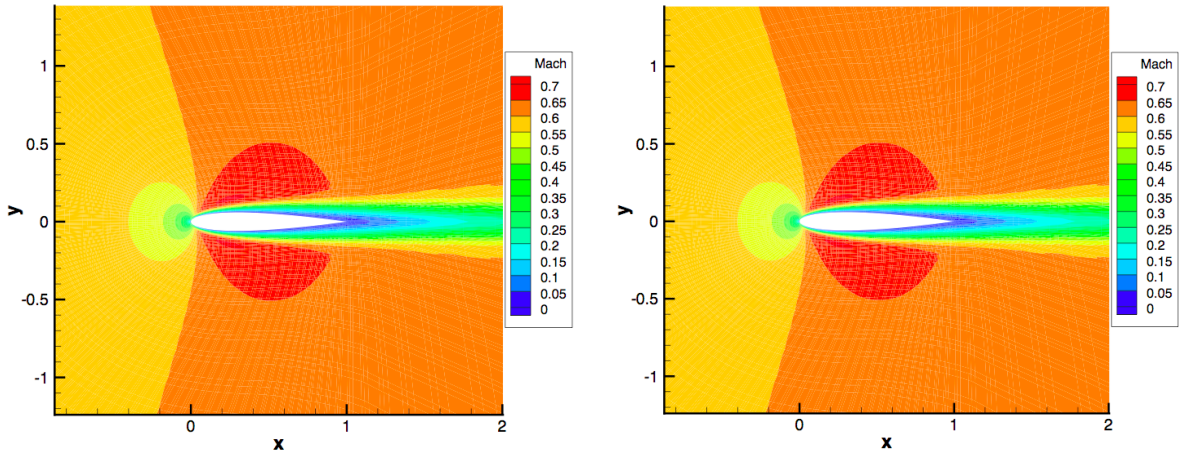


Figure 2: NACA 0012 airfoil mesh.

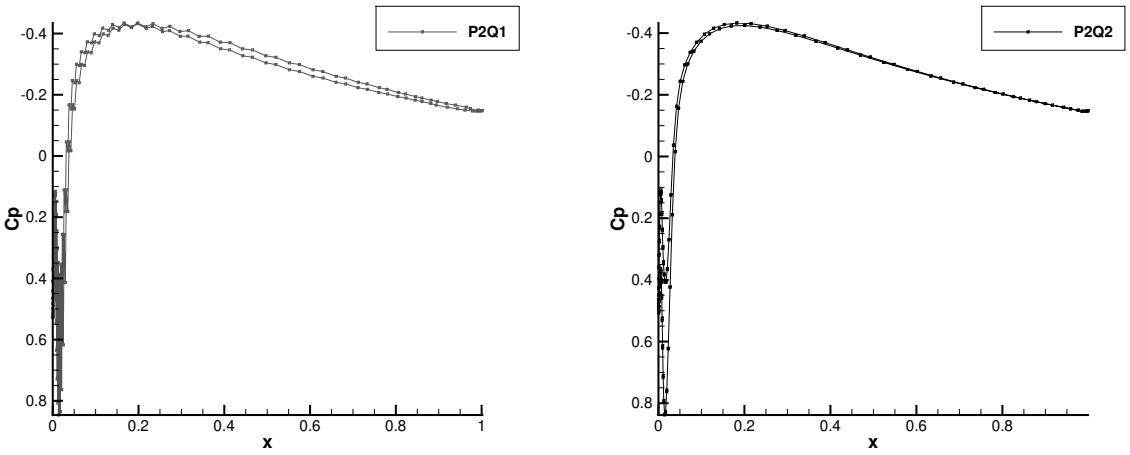


(a) Linear mesh results

(b) Second order mesh results.

Figure 3: Mach contours for a P2 spectral difference solution over a NACA 0012 airfoil.

A more detail analysis of the results, however, looking at pressure coefficient distributions over the airfoil surface, which are shown in Fig. 4, reveals that there is a noticeable mesh effect. The higher order mesh representation, with the Cp results shown in Fig. 4(b), allows the capture of equal values of Cp on the upper and lower surfaces of the airfoil, which is the expected result for a symmetric airfoil at zero angle of attack. On the other hand, calculations with the Q1 mesh yield differences in the pressure coefficient distribution between upper and lower surfaces of the airfoil. Moreover, one can also observe some small oscillations in the Cp distributions, both upper and lower surfaces, for the results with the linear mesh. The results for the Q2 mesh do not have such oscillations.



(a) Linear mesh results.

(b) Second-order mesh results.

Figure 4: Pressure coefficient distributions calculated with 3rd-order method using linear and quadratic meshes ($M_\infty = 0.6$, $Re = 1000$, $\alpha = 0$ deg).

4.2 Heaving and pitching airfoil movement

Unsteady aerodynamic simulations can be performed with the capability presnetly available in the code and some test cases are the subject of future efforts. A test case that considers a heaving and pitching airfoil is described in the 4th International Workshop on High-Order CFD Methods [11]. This problem is used to test the accuracy and performance of high-order flow solver for problems with deforming domains and, hence, it is suitable to validate the contributions of the present work. In particular, the test case consists of a NACA 0012 airfoil undergoing a smooth flapping-type motion, that is, a pitching motion along with a vertical displacement from rest up to a position one chord length higher. The flapping motion is performed for the total time of 2 dimensionless time units, by heaving and pitching the airfoil about a point located at 1/3 of the airfoil chord, as depicted in Fig. 5.

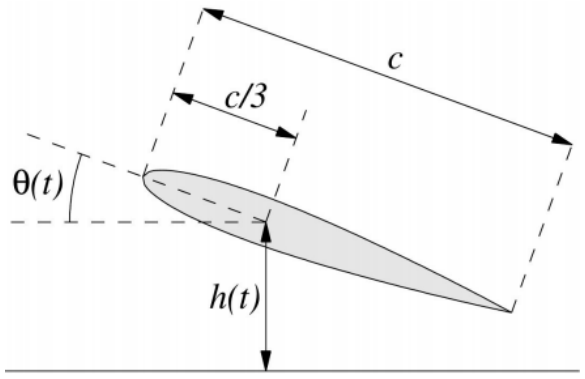


Figure 5: Definition of parameters for airfoil motion.

Table 1: Motion description for the pitching and heaving airfoil.

Case 1 - Pure heaving	Case 2 - Flow aligning	Case 3 - Energy extraction
h(t) = b_2(t)	h(t) = b_2(t)	h(t) = b_3(t)
$\theta(t) = 0$	$\theta(t) = A_2 \cdot b_1$(t)	$\theta(t) = A_3 \cdot b_1$(t)

The motion details are included in Table 1, where $A_2 = 60\pi/180$ and $A_3 = 80\pi/180$. Moreover, the motion is defined by the following expressions

$$b_1 = t^2(t^2 - 4t + 4) \ , \tag{20}$$

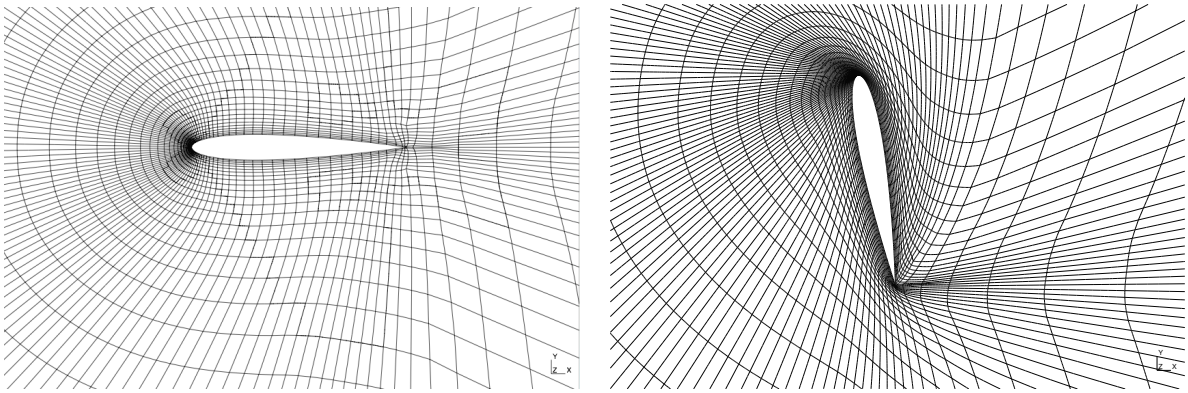
$$b_2 = t^2 \frac{(3-t)}{4} \ , \tag{21}$$

$$b_3 = t^3 \frac{(-8t^3 + 51t^2 - 111t + 84)}{16} \ , \tag{22}$$

where h(t) is the vertical displacement and $\theta(t)$ represents the pitching angle.

If one considers the angular displacement in cases 2 and 3, it is clear that $b_1 = 1$ when $t = 1$ and, when $t = 2$, $b_1 = 0$. Hence, the angular excursion of the airfoil starts at 0 deg., at $t = 0$, goes up to $A_2 = 60$ deg., or $A_3 = 80$ deg., depending on the test case, and finally returns to 0 deg. The heaving, though, is a single movement growing monotonically from 0 to 1 chord units for cases 1 and 2. For the last case, case 3, however, the movement still ranges from 0 to 1 chord units, but in a non-monotonic manner.

Some of the mesh movement results of interest for the NACA 0012 airfoil are presented in Fig. 6. Figure 6(a) shows the NACA 0012 airfoil at its initial configuration, for which the computational mesh is generated, whereas Fig. 6(b) presents the mesh at time $t = 1$ time units for the excursion previously described of case 3, *i.e.*, for an angle of attack $\theta = 80$ deg. The mesh is moved in the present study using the concept of radial basis

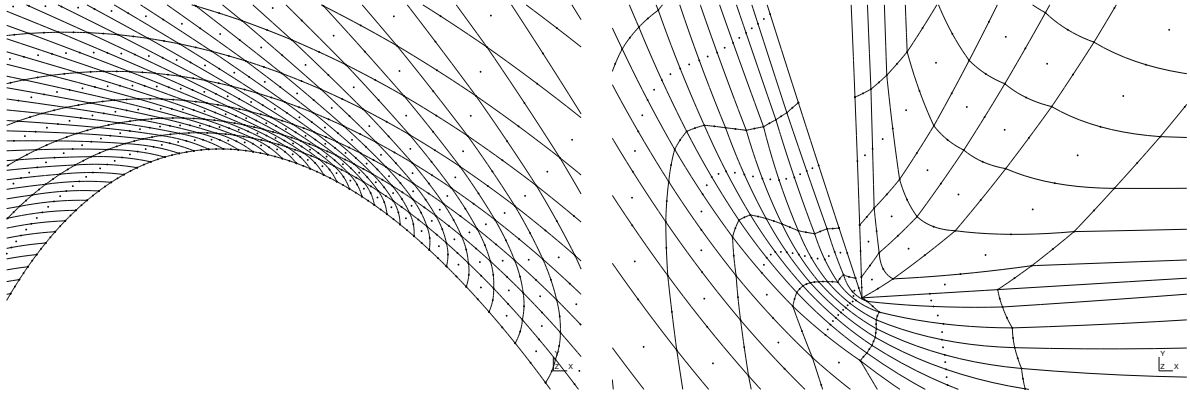


(a) Original NACA 0012 mesh. (b) NACA 0012 mesh after airfoil rotated by 80 deg.

Figure 6: Mesh movement via RBF for the NACA 0012 airfoil grid.

functions (RBFs) [12]. For this case, the mesh experiences a severe deformation. However, it is possible to observe in Fig. 7 that the mesh is still valid in the sense that there are no

intersecting edges even at the most problematic locations, *i.e.*, at the trailing and leading edges of the airfoil.



(a) NACA 0012 leading edge close-up view. (b) NACA 0012 trailing edge close-up view.

Figure 7: Assessment of cell validity after airfoil is rotated by 80 deg.

5 CONCLUDING REMARKS

The results obtained so far have demonstrated the capability of the radial basis function (RBF) technique to properly handle tangled and improper cells near the boundary, when a curving strategy is applied to the mesh. Furthermore, viscous simulations have been carried out with appropriate curved meshes and comparisons with their low order mesh counterparts have demonstrated the effectiveness of high-order meshes for viscous flows. Moreover, the RBF technique proved capable of handling the desired very large mesh displacements for the unsteady airfoil motion test cases.

ACKNOWLEDGMENTS

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

REFERENCES

[1] Bassi, F., Rebay, S., Mariotti, G., Pedinotti, S., and Savini, M., A High-Order Accurate Discontinuous Finite Element Method for Inviscid and Viscous Turbomachinery Flows, *Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, Technologisch Instituut, Antwerpen, Belgium, 1997, pp. 99–109.

[2] Wang, Z.J., Liu, Y., May, G., and Jameson, A., Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations, *Journal of Scientific Computing*, Vol. 32, No. 1, July 2007, pp. 3938–3956.

[3] May, G. and Jameson, A., A Spectral Difference Method for the Euler and Navier-Stokes Equations on Unstructured Meshes, *44th AIAA Aerospace Sciences Meeting*, Reno, NV, Jan. 2006.

[4] Van den Abeele, K., Lacor, C., and Wang, Z.J., On the Stability and Accuracy of the Spectral Difference Method, *Journal of Scientific Computing*, Vol. 37, No. 2, Nov. 2008, pp. 162–188.

[5] Spitieri, R.J. and Ruuth, S.J., A New Class of Optimal High-Order Strong-Stability-Preserving Time Discretization Methods, *SIAM Journal on Numerical Analysis*, Vol. 40, No. 2, 2003, pp. 469–491.

[6] Moreira, F.M., Jourdan, E., Breviglieri, C., Aguiar, A.R.B., and Azevedo, J.L.F., Implicit Spectral Difference Method Solutions for Compressible Flows Considering High-Order Meshes, AIAA Paper No. 2016-3352, *Proceedings of the 46th AIAA Fluid Dynamics Conference*, Washington, DC, June 2016.

[7] Arnold, D.N., Brezzi, F., Cockburn, B., and Marini, L.D., Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems, *SIAM J. Numer. Anal.*, Vol. 39, No. 5, May 2001, pp. 1749–1779.

[8] Kannan, R., *High Order Spectral Volume and Spectral Difference Methods on Unstructured Grids*, Ph.D. Thesis, Iowa State University, Ames, Iowa, 2008.

[9] Van den Abeele, K., *Development of High-Order Accurate Schemes for Unstructured Grids*, Ph.D. Thesis, Vrije Universiteit Brussels, Brussels, Belgium, 2009.

[10] Van den Abeele, K., Parsani, M., and Lacor, C., An Implicit Spectral Difference Navier-Stokes Solver for Unstructured Hexahedral Grids, AIAA Paper No. 2009-0181, *Proceedings of the 47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Orlando, FL, Jan. 2009.

[11] Wang, Z.J., Cagnone, J.S., Careni, D., de Wiart, C.C., Couaillier, V., Fidkowski, C., Galbraith, M., Hartman, R., Ollivier-Gooch, C., Persson, P.O., Hillewaert, K., Ekaterinaris, J., Huynh, H.T., Kroll, N., and Vincent, P., HiOCFD4 – 4th International Workshop on High-Order CFD Methods, 2015, https://how4.cenaero.be.

[12] Aguiar, A.R.B., Breviglieri, C., Santos, T.M., Azevedo. J.L.F., and Wang, Z.J., High-Order Meshes for Flow Simulations with a Spectral Difference Method, AIAA Paper No. 2017-4294, *Proceedings of the 23rd AIAA Computational Fluid Dynamics Conference*, Denver, CO, June 2017.