ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. J. Martin-Burgos, D. González-Juárez and E. Andrés-Pérez
Crete Island, Greece, 5–10 June 2016

A LAPLACIAN MESH DEFORMATION TECHNIQUE FOR SIMULATION-DRIVEN DESIGN OPTIMIZATION

M. J. Martin-Burgos[1], D. González-Juárez[1] and E. Andrés-Pérez[2]

[1] Fluid Dynamics Branch, Spanish National Institute for Aerospace Technologies (INTA)
Crta. de Ajalvir Km. 4.5. 28850 Torrejón de Ardoz
e-mail: {martinbj, gonzalezjd}@inta.es

[2] Engineering Department, Ingeniería de Sistemas para la Defensa de España S.A. (ISDEFE-INTA) & Technical University of Madrid (UPM).
Crta. de Ajalvir Km. 4.5. 28850 Torrejón de Ardoz
e-mail: eandres@isdefe.es, esther.andres@upm.es

Keywords: Mesh deformation, Aerodynamic shape optimization, Computational Fluid Dynamics, Partial differential equations.

Abstract. *A simple method to automatically adapt the computational grid to surface boundary deformations is presented. Mesh adaptation techniques can be used to reduce the number of re-meshing steps in aerodynamic shape optimization problems or during the solution of a coupled fluid-structure interaction problem. The method is inspired on the solution of the Laplace equation and the resulting algorithm resembles a fully explicit Jacobi iterative scheme, avoiding the need to invert a large matrix, in order to solve the resulting system of equations. The proposed method can withstand large deformations and also includes orthogonality correction to maintain the quality of the grid, especially in the boundary layer region. The proposed technique is presented in two-dimensional unstructured hybrid grids, but can be easily extended to three-dimensions.*

1 INTRODUCTION

Using Computational Fluid Dynamics (CFD) simulations, the mesh must be updated to conform the boundary surface modifications. The mesh update can be achieved by re-meshing, but the generation of the mesh, for industrial complex configurations, is a time-consuming operation and automatic grid generation is not always feasible. This issue particularly arises in the search of an optimal shape using CFD simulations, aero-elastic analysis, or control surface deflection problems. In addition, the use of different overlapped grids or chimera for the simulation of moving parts is a well established and versatile technique. However, the interpolation between zones is not intrinsically conservative and for some specific problems, such as fluid-structural interaction, the use of mesh deformation would be preferable instead, in order to reduce the numerical noise. To avoid the regeneration of the grid, mesh deformation or mesh update techniques can be considered as a fast and cheap alternative.

In the literature, there are several proposals for mesh deformation. The transfinite interpolation [1,2] is an algebraic method, generally used in structured grids, that interpolates the displacement of mesh points along arc mesh lines. This method is simple but not appropriate for unstructured grids. The advancing front method is another algebraic technique for unstructured grids [3]. In this method, grid points are sorted with increasing distance to the deformation surface. Then, new displacements for each point are scaled average from the displacements of the neighbor points, which have a smaller distance and which are thus already updated. This strategy leads to advancing the front of displacements as iso-surfaces of equal distances, which was found to be of advantage in order to transport the information orthogonal to the surfaces into the interior of the grid. The advancing front algorithm is simple and computationally very efficient. However, the robustness is uncertain for large deformations and it is unclear how to face multiple components, such as wing-flap configurations.

In the tension spring analogy [4], the edges of the computational grid are modeled as springs, whose stiffness are inversely proportional to the length of the edge, making short edges stiffer. This method is mathematically simple and can be solved with an iterative Jacobi solver. However, the spring-edge idealization will eventually fail, in some cases even for small displacements. This method prevents the colliding of the vertices, but there is no mechanism to prevent a vertex to cross and edge and the consequently collapsing element.

In order to mitigate overstretching and negative volume elements, the edge-spring is usually combined with a torsional spring analogy [5]. In two-dimensional grids, each vertex is attached to a torsional spring that prevents the faces from squashing. Unfortunately, the extension to three-dimensional grids is not straightforward, as it may lead to negative volume tetrahedrons, even if all tetrahedron faces are valid elements. There are some attempts to extend it to three-dimensions using more complex formulations and kinematic systems [6]. However, the spring-torsion analogy works under the assumption of relatively small displacements. When elements are nearly flat, the Partial Differential Equations (PDE) that governs the system is ill-conditioned, imposing a limited time step and convergence problems. Despite of its limitations, the spring analogy still remains popular thanks to its easy implementation.

The linear elasticity analogy [7] assumes that the computational mesh obeys the isotropic linear elasticity equations, much like a rubber. The modulus of elasticity is often taken to be inversely proportional to the cell volume or the cell aspect ratio. In this manner high aspect ratio cells, as those close to the boundary layer, are less susceptible to compression. The spring and linear elasticity approaches are compared in [8], where only the linear elasticity method is stable enough for large scale deformations of 3D hybrid Navier-Stokes grids. The linear elasticity approach allows significantly larger geometric deformations and can be naturally extended to three-dimensional grids. However, it requires the solution of a large system of equations, one for each vertex, and the computational effort is not diminished in comparison with the CFD simulations.

The concept of Radial Basis Functions (RBF) has been also applied to update the mesh [9-11]. In the RBF method, the mesh points are calculated through a global function that depends on the distance of the base points, which are usually the deformed surface. The RBF method does not require connectivity information, making it suitable for unstructured grids. However, without information of the elements shape, there is little control of the grid quality. Some authors propose the incorporation of a second set of base points to limit the change in orthogonality near the wall [12].

Partial Differential Equations (PDE) are often used for generating grids. The use of the Laplace equations is a popular approach to optimize smoothness and control the variation of cell volumes. Techniques based on PDEs can also be applied for updating the grid to a moving boundary [13], which can be naturally extended to three-dimensions [14]. This approach tackles the mesh motion as a boundary value problem by finding the mesh displacement field that emanates from the solution of an Euler-Lagrange equation. This works when the mesh is composed of approximately equal-sized elements and the motion of the interface boundary is in the order of the size of the elements; otherwise, results in an algorithm breakdown. In order to prevent an ill-conditioned system, a constraint condition is applied element wise; so larger elements absorb most of the distortion. Similar to the linear elasticity analogy, methods based on PDE can withstand a large amount of deformation, but require the solution of a very large equation system; often tackled with a gradient conjugate method, or one of its variants.

The tension spring and linear elasticity are closely related to the Laplace equation. In this work, a mesh deformation algorithm based on the Laplace operator is presented. The method is similar to the linear elasticity in terms of robustness and deformation withstanding, while it is as easy to implement as the spring analogy. The method can update very large deformations without losing grid quality and the resulting equation system can be easily solved with an explicit Jacobi algorithm.

2 PROPOSED APPROACH

2.1 Mesh deformation

The heat equation describes how thermal energy is distributed through a body until a steady state, governed by the Laplace equation

$$\nabla^2 u = 0$$

Let's consider the kinematic system represented in Figure 1, which illustrates a an interior vertex a connected to its neighbors x_i through the edges of an unstructured grid.

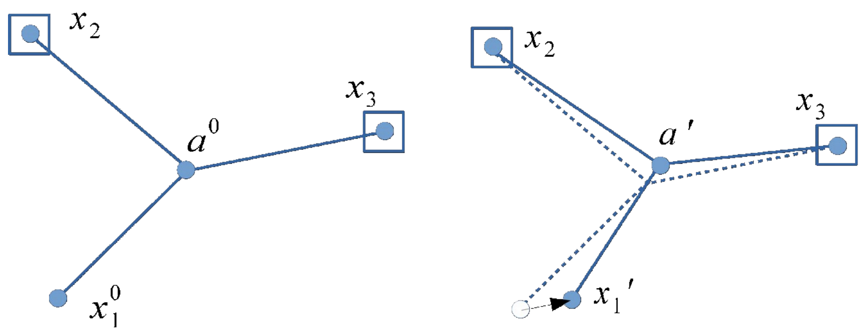

Figure 1: Kinematic system, where the vertex a is updated due to a perturbation of one of its neighbor.

Equivalently to the heat analogy, a mesh deformation rule is to minimize the variation of the relative vertex displacements $Min\{\partial(x_i\text{-}x_a)\}$, compared with the original ones. By imposing Dirichlet boundary conditions and using relative displacements as a field, this can be expressed as a problem to calculate the solution of the Laplace equation, which in the discrete form can be approximated as a linear system of equations:

$$\frac{\sum_i \left[(x_i' - x_a') - (x_i^0 - x_a^0)\right] w_i}{\sum_i w_i} \tag{1}$$

where the term w_i is the norm usually associated to the grid domain, such as the cell dimensions. The superscript x^0 indicates the original positions, while the superscript x' denotes the updated vertex positions. The above expression can be rewritten and be solved wth an explicit Jacobi algorithm:

$$x_a^{n+1} = \frac{\sum_i (x_i^n - x_i^0) w_i}{\sum_i w_i} + x_a^0 \tag{2}$$

where n is the iteration index. The norm selected in the test cases is the inverse squared length of the edge connecting two vertices:

$$w_i = \left| x_i^0 - x_a^0 \right|^{-2} \tag{3}$$

In this way, smaller elements, such as those close to the boundary layer, will remain stiffer, while deformations are absorbed by larger elements, which can sustain more strain. Alternatively, selecting the inverse of the dual area associated to the vertex or a weight calculated from the area of the associated elements, will achieve similar results. However, the edge distance turns out better for avoiding invalid elements in regions with high mesh stretching. In this formulation, displacements in each coordinate direction become decoupled and are solved independently. The norm are calculated from the original grid, before the deformation, which is taken as a reference. So, a good grid quality is expected from the original grid. The above expression corrects the relative position of the vertex, but it is insufficient to correct the skewness of the elements, or the orthogonality, which is the subject of the next section.

2.2 Mesh orthogonality correction

The concept of mesh orthogonality relates how a grid element faces an optimal angle. In convection-diffusion simulations a "false diffusion" is an error when the flow is not aligned with the grid lines, which is accumulated downstream. Orthogonality is a relevant mesh quality metric, in especial at the boundary layer, and might compromise the robustness and accuracy of the simulations.

The mesh deformation algorithm presented in Eq. (2) can sustain large deformations, but there is no information about the shape of the elements. It deals very well with translation transformations, e.g. if the airfoil is moved to the left. On the other hand, rotations present a more challenging problem and the orthogonality is required to be corrected, to maintain the cell lines perpendicular to the wall. This can be achieved by preserving the angle of the elements, as it is illustrated in Figure 2.

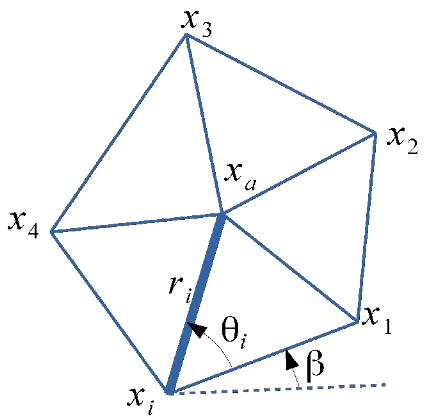

Figure 2: Kinematic system, where the vertex a is updated to maintain the angle θ of the element.

In a similar way to section 2.1, the aim is to minimize the variation of the angles after deformation: *Min{∂θ}* (this can be trivially rewritten as $\theta'=\theta^0$). The reference angles typically are 90º for quadrilaterals and 60º for triangles. In our test cases, the angles taken as reference are those of the original grid, before deformation. In two dimensions, this can be expressed as a coupled linear system of equations.

$$x_a^{n+1} = \alpha \frac{\sum_i \left[x_i^n + r_i^0 \cos(\theta_i^0 + \beta_i^n) \right] w_i}{\sum_i w_i} + (1-\alpha) x_a^n$$
$$y_a^{n+1} = \alpha \frac{\sum_i \left[y_i^n + r_i^0 \sin(\theta_i^0 + \beta_i^n) \right] w_i}{\sum_i w_i} + (1-\alpha) y_a^n \qquad (5)$$

where the weight w_i is the same as Eq. (4), the terms x and y are the Cartesians coordinates, while the angles β and θ are illustrated in Figure 2. The superscript β^n indicates the updated value at iteration level and β^0 the reference value, which in our test cases are the angles of the original mesh elements. The above formulation is ill-defined (there are oscillations between two local equilibrium states), and therefore requires a relaxation coefficient α to ensure the convergence; a value of $\alpha=0.5$ ensures the convergence. The algorithm corrects the orthogonality, but cannot prevent elements from squashing. The final algorithm comes as the combination of both approaches.

$$X^{n+1} = \gamma \cdot X_p^n + (1-\gamma) \cdot X_g^n \qquad (5)$$

where the term X_p are the updated coordinates from Eq. (3) that maintains the vertex relative positions, and X_g are the updated coordinates from Eq. (5) that corrects the orthogonality of the mesh. The value of γ highly depends on the severity of the deformation. The orthogonality correction has slow convergence, while the algorithm based on the relative positions is very robust. In the test cases presented in this work, the strategy employed is to start with $\gamma=0$, to give more strength to the relative position correction, then gradually increase the orthogonality correction, as the residuals drops to a final value of $\gamma=1$.

3 TEST CASE

The test case considered is a two dimensional hybrid grid for a RAE2822 airfoil, as shown in Figure 3, where the geometry is rotated 20º. The proposed grid deformation method is compared with the advancing front algorithm [3].

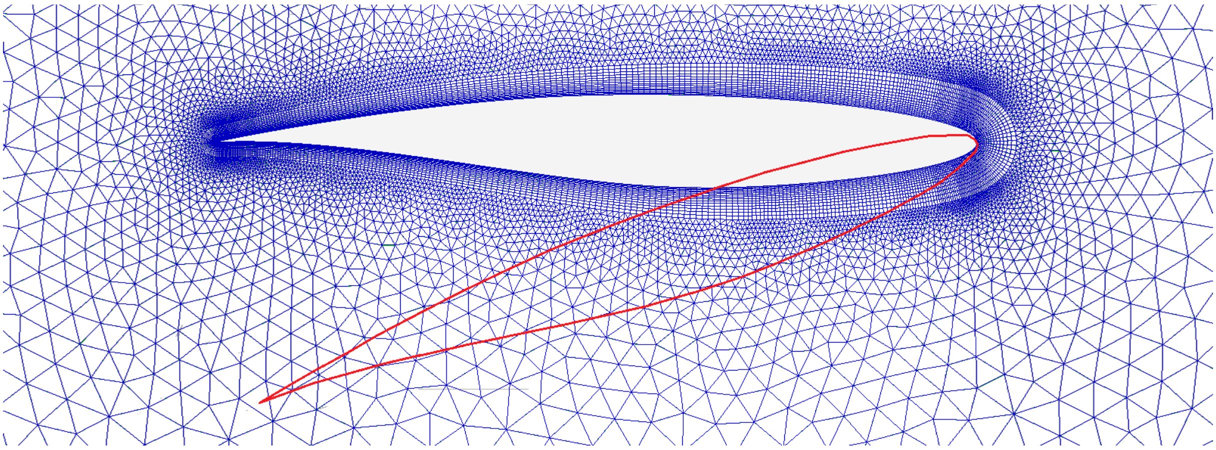

Figure 3: Original RAE2822 hybrid grid and the rotation tested for the mesh deformation.

Figure 4 and Figure 5 compare the mesh updates performed with the two different methods. Even when none of the methods turn out with invalid elements, there are disparities on the grid quality. The first one is the advancing front algorithm, which presents squashing cells for this test case. The Laplacian deformation, in comparison, is able to maintain the integrity of the elements, but the grid lines are not perpendicular to the wall, which might compromise the accuracy of the simulations. Both methods are then post-processed with the orthogonality correction presented in section 2.2, delivering good quality grids. Both post-processed grids lead to almost identical deformations, because the orthogonality correction is based on the Laplace operator, and therefore, the solution is theoretically unique. Disparities can only be explained at numerical level.

The advancing front algorithm is very fast and computationally efficient; in contrast, the Laplace deformation requires a long number of iterations. The orthogonality correction is then applied to both grids; because the starting grid is already close to the optimal, convergence is very fast.

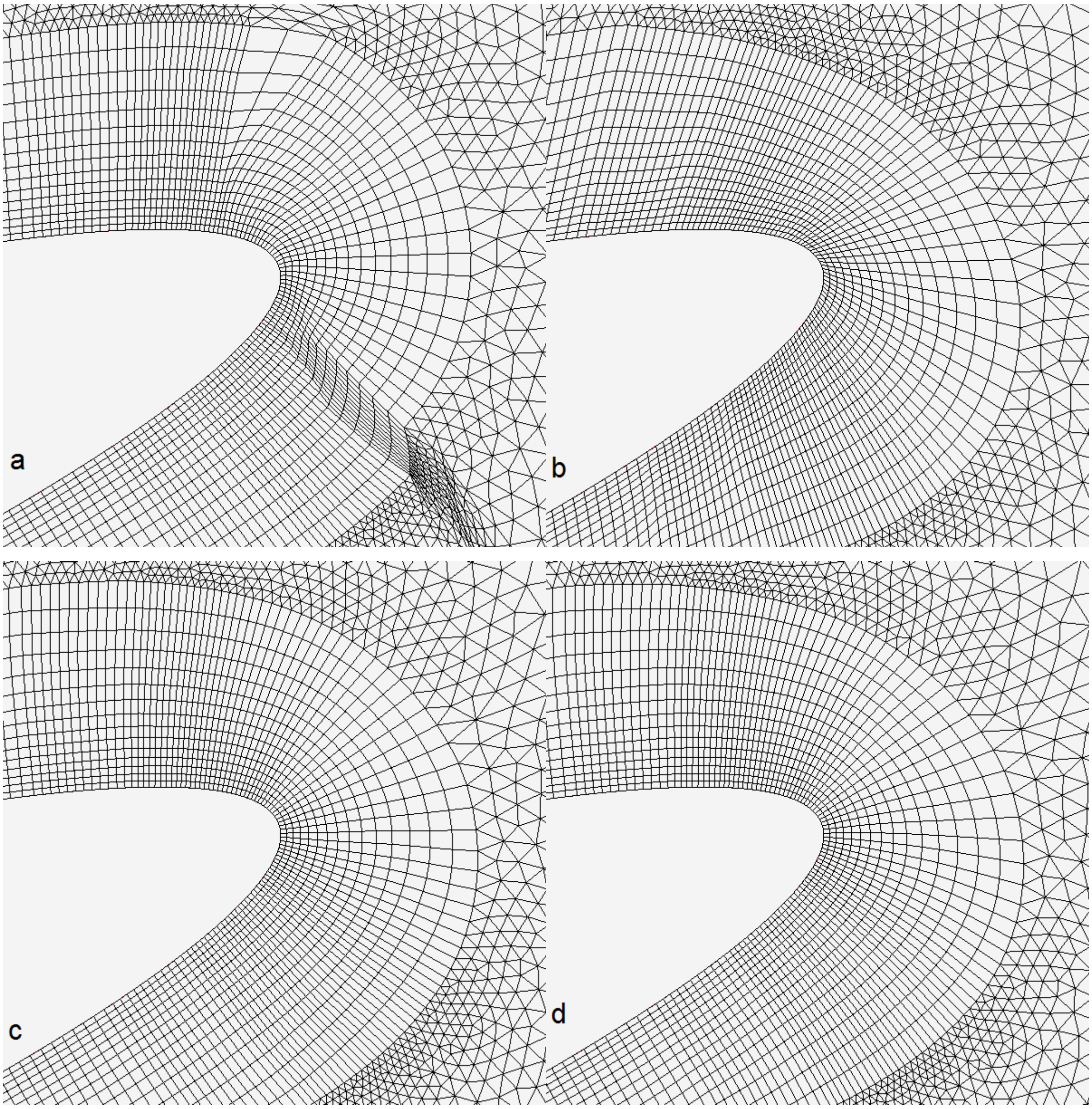

Figure 4: Details at the leading edge of the mesh deformation algorithm. a) Advancing front, b) Only Laplacian deformation, c) Advancing front with orthogonality correction, and d) Laplacian with orthogonality correction.

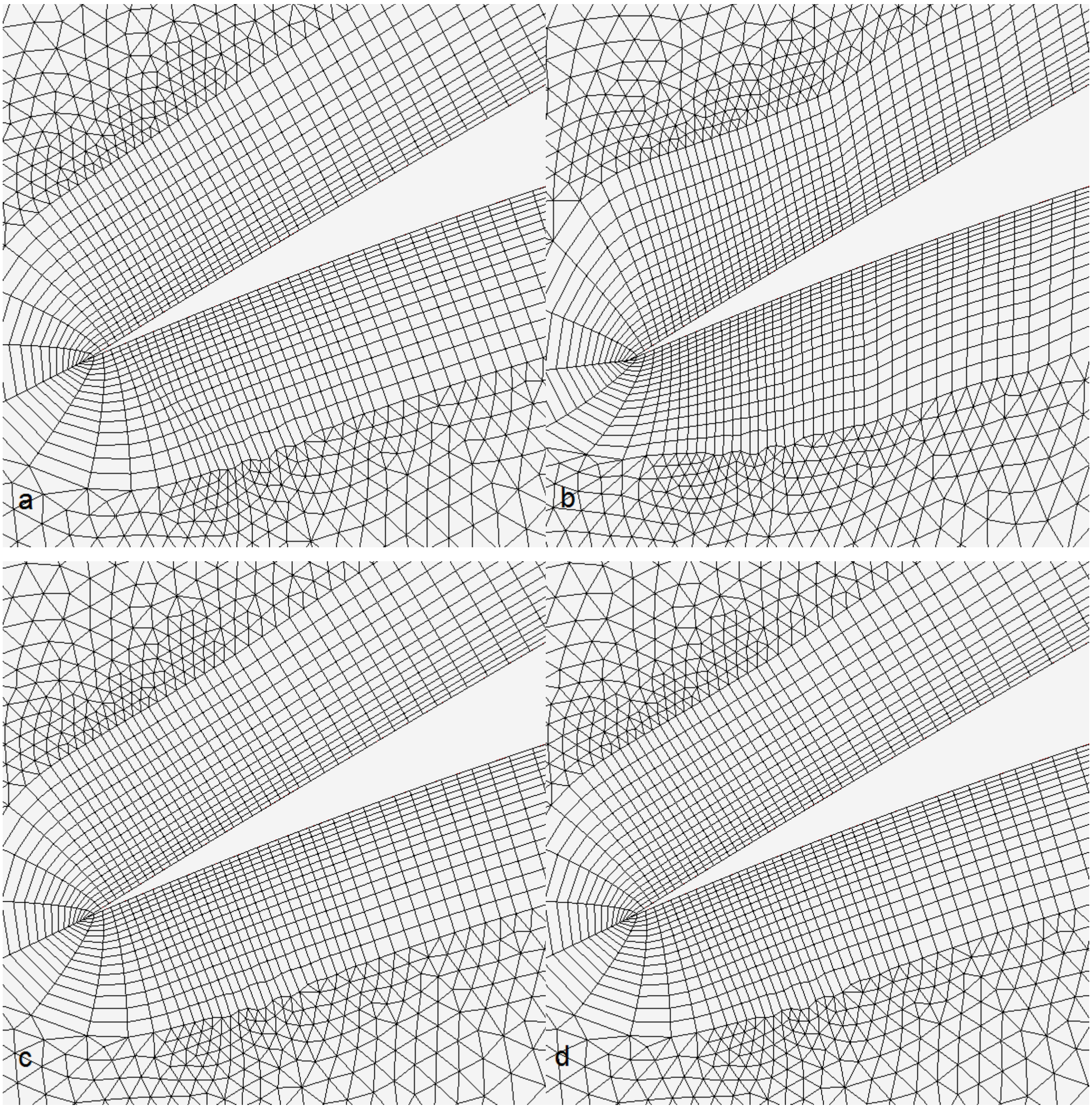

Figure 5: Details at the trailing edge of the mesh deformation algorithm. a) Advancing front, b) Only Laplacian deformation, c) Advancing front with orthogonality correction, and d) Laplacian with orthogonality correction. Note: c and d are almost identical grids, because the orthogonality correction is based on the Laplace operator, and therefore, the solution is unique.

The skewness is used to compare the grid quality with the methods exposed, as shown in Figure 6. As expected, the Laplacian deformation without orthogonality correction is not able to maintain the correct skewness of the elements. Hexahedrons are associated to the boundary layer and a value close to the original grid, previous to the deformation, indicates that most of the deformation is transferred to the prismatic layer, where is less critical. The presented algorithm, with orthogonality correction, shows a grid quality very close to the original.

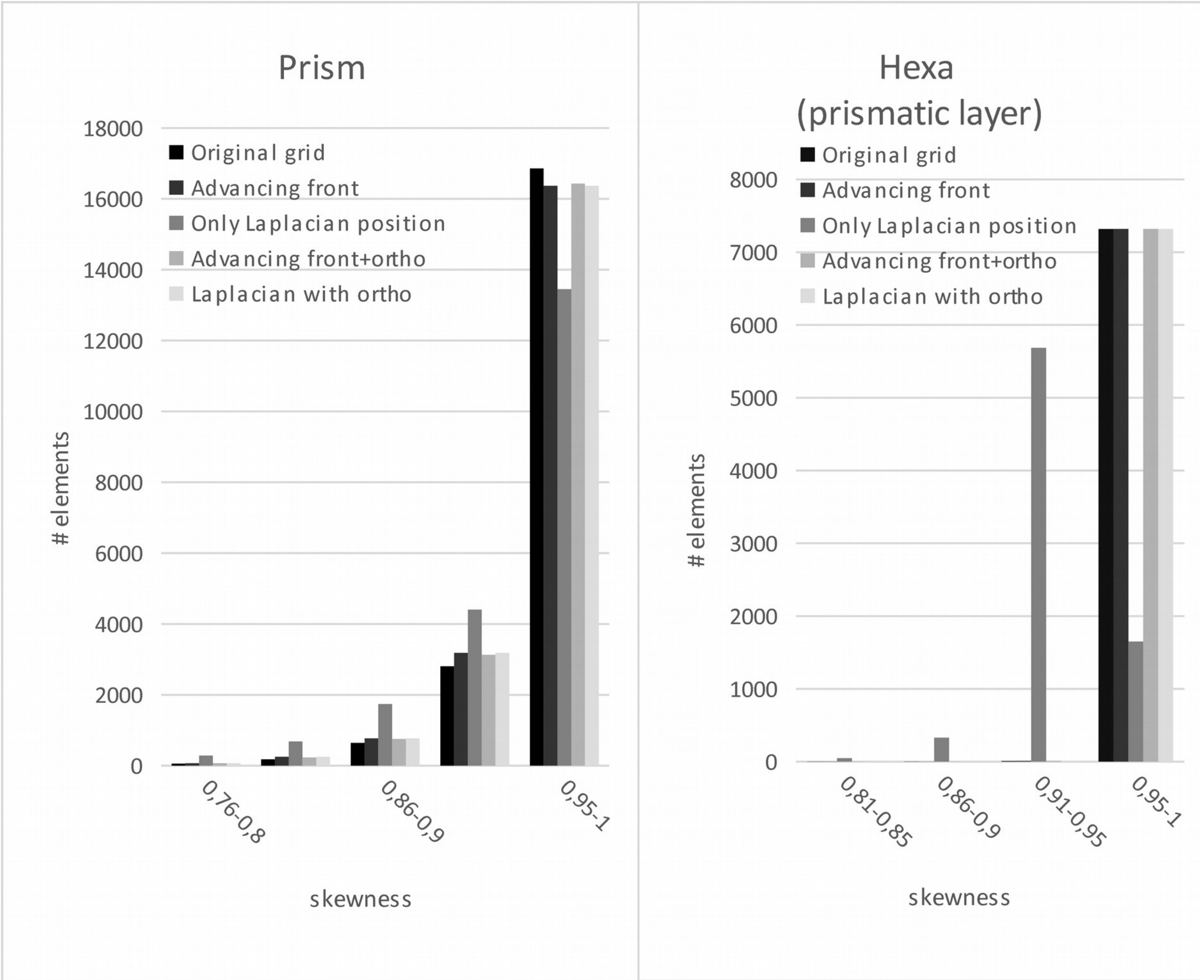

Figure 6: Comparison of the skewness, as a measurement of the quality of the grid.

4 CONCLUSIONS

A mesh deformation technique with orthogonality correction is presented in this work, inspired in the Laplace operator. The proposed method can maintain good grid quality, even in the case of large deformations. The method is very easy to implement, resulting into an explicit Jacobi iterative algorithm. Moreover, the algorithm is an element-based data structure ,and it is suitable for parallelization. In comparison, the advancing front algorithm is computationally more efficient, but the grid quality is uncertain. On the other hand, the Laplacian deformation requires an iterative algorithm. Using the grid delivered from the advancing front algorithm, and then applying the Laplacian deformation correction, to fix bad shaped elements, convergence is very quickly reached.

Future works would extend the proposed method to three dimensional grids and testing the method for multi-comonent grids. A further research will tackle how to improve the convergence rate to reduce the computational time; one posibility may be to take advantage of multigrid techniques.

REFERENCES

[1] L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K.J. Badcock, and B.E. Richards, “A grid deformation technique for unsteady flow computations”, International Journal of Numerical Methods in Fluids, vol. 32, Issue 3, 2000, pp.285-311.

[2] S.P. Spekreijse, B.B. Prananta, and J.C Kok, “A simple, robust and fast algorithm to compute deformations of multi block structured grids”, NLR-TP-2002-105, 2002.

[3] T. Gerhold, J. Neumann, “The parallel mesh deformation of the DLR TAU-Code”, vol. 96 of Notes on Numerical Mechanics and Multidisciplinary Design (NNFM), Springer Berlin Heidelberg, 2008, Chapter 20, pp. 162-169.

[4] Batina, J.T.,“Unsteady Euler Airfoil Solutions. Using Unstructured Dynamic Meshes”, AIAA Journal,. Vol. 28, No. 8, 1990, pp.1381-1388.

[5] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne, "Torsional Springs for Two-Dimensional Dynamic Unstructured Fluid Meshes", Computer Methods in Applied Mechanics and Engineering, Vol. 163, 1998, pp. 231-245.

[6] C. Degand, C. Farhat, A three-dimensional torsional spring analogy method for unstructured dynamic meshes, Comput. Struct. 80 (2002) 305.

[7] A. Masud, M. Bhanabhagvanwala, R. A. Khurram, “An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction”, Computers & Fluids 36 (2007) 77-91.

[8] Z. Yang, D.J. Mavriplis, “Unstructured dynamic meshes with higher order time integration schemes for the unsteady Navier-Stokes equations”, AIAA 2005-1222, 41th AIAA Aerospace Sciences Meeting and Exhibit, January 10-13, 2005/Reno, NV.

[9] A. de Boer, M.S. van der Schoot, and H. Bijl, "Mesh deformation based on radial basis function interpolation", Computers and Structures, Vol. 85, 2007, pp. 784{795.

[10] T.C.S. Rendall, and C.B. Allen, “Unified Fluid-Structure Interpolation and Mesh Motion using Radial Basis Functions”, International Journal for Numerical Methods in Engineering. Vol. 74(10), 2008, pp1519-1559.

[11] S. Jacobsson, and O. Amoignon, “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization”, Computers & Fluids 36 (2007) 1119-1136.

[12] T. Gillebaart, A.H. van Zuijleny, and H. Bijlz, "Radial basis function mesh deformation including surface orthogonality", 54th AIAA Aerospace Sciences Meeting, 4-8 January 2016, San Diego, California, USA, AIAA 2016-1674.

[13] A. Masud, M. Bhanabhagvanwala, R. A. Khurram, “An adaptive mesh rezoning scheme for moving boundary flows and fluid-structure interaction”, Computers & Fluids 36 (2007) 77-91.

[14] H. Kanchi, A. Masud, “A 3D adaptive moving scheme”, International Journal for Numerical Methods, in Fluids 54 (6-8), 923-944.

[15] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Optimization on Unstructured Meshes". AIAA Journal, Vol. 40, No. 6, June 2002, pp. 1155–1163