Paper 34

©Civil-Comp Press, 2012
Proceedings of the Eighth International Conference
on Engineering Computational Technology,
B.H.V. Topping, (Editor),
Civil-Comp Press, Stirlingshire, Scotland

Evaluation of Radial Basis Functions for the Deformation of Unstructured Meshes

P.E. Kouskouris and I.K. Nikolos
Department of Production Engineering and Management
Technical University of Crete, Chania, Greece

Abstract

The radial basis function (RBF) mesh deformation methodology is used in this paper to impose large deformations on two-dimensional unstructured hybrid meshes around airfoils, including translation and rigid rotation of the airfoil with respect to its original position. It is realized that, for such large deformations, some of the adopted RBFs failed to produce acceptable deformed meshes: the resulting meshes include large areas with overlapping elements. In order to improve the resulting mesh quality the RBF mesh deformation methodology is combined with a Laplacian smoothing operator, which is repeatedly called after the completion of the basic RBF mesh deformation procedure. A widely adopted metric for mesh quality is used to evaluate the quality of the deformed mesh, with and without Laplacian smoothing, for various radial basis functions. Results for unstructured hybrid meshes are presented, demonstrating the effect of the various RBFs as well as the application of Laplacian smoothing, in terms of grid quality metric values.

Keywords: mesh deformation, radial basis functions, unstructured hybrid meshes.

1 Introduction

Interest in computational fluid dynamics (CFD) technology on moving grids has increased during the last decade. The need of simulating flows past moving and/or deforming bodies is continually growing. Two key elements of this technology have been time-accurate discretization schemes on moving grids (also known as dynamic meshes), and robust algorithms for updating the position of the grid points in such meshes. Farhat [1] presents some of the theoretical and computational advances made in this area, highlights sample applications they have enabled in aero-elasticity and multidisciplinary optimization, and concludes with a brief discussion of specific barriers to progress. Since the reconstruction of the computational mesh is computationally unaffordable, the use of fast mesh deformation tools is required. Several methods are currently in use, such as the method of spring analogy, simple algebraic methods, transfinite interpolation (TFI), the solution of partial differential equations, the moving sub-mesh approach (MSA) and methods based on radial basis functions (RBFs). The latter exhibit small computational cost while in general maintain the overall quality of the original grid after the deformation; they are independent of the mesh connectivities and, therefore, structured, unstructured and hybrid meshes can be treated in the same manner.

The classic method, which uses the spring analogy to induce strain on the grid edges, was originally proposed by Batina [2] for unstructured grids and extended later by Robinson et al. [3] for structured grids. This method can handle large deformations, but because it is an iterative method, which simulates mesh generation in solving elliptic differential equation, proved time consuming, especially for large grids. Special cases of this approach include the use of Laplacian operators and Biharmonic functions [4]. Cizmas and Gargoloff [5] suggested a grid generation and deformation algorithm for wings with large deformations. Firstly, the spring analogy technique was applied to deform the nodes within a mesh layer and secondly, the layers were deformed in order to be perpendicular to the boundaries of the domain and to the surface of the wing. All these methods require solving a sparse system of equations for all grid nodes, which can be solved approximately for a small number of iterations. In addition, it is important to note that the grid re-meshing algorithm had the benefit of keeping unchanged the grid topology, which is advantageous for a parallel flow solver. Schuster et al. [6] and Bhardwaj et al. [7] used a simple algebraic method to deform the grid, by reordering the nodes along the grid lines in a direction perpendicular to the solid wall. This method can cause problems for complex geometries, as in this case is difficult to determine the vertical direction in which the deformation takes place. Moreover, the method is limited to small deformations while in large deformations results in a very low mesh quality. Eriksson [8] proposed a Transfinite Interpolation method (TFI) to redefine unique computational spaces, while Hartwich and Agrawal [9] combined the spring analogy approach with the TFI method for mesh redefinition in grids consisting of multiple blocks. The spring analogy is used to move all boundary edges for each block and, then, TFI method is applied to reconstruct the grids in deformed blocks. The method was further developed by Potsdam and Guruswamy [10], and included parallel processing.

The method presented in [11, 12, 13] uses Radial Basis Functions (RBF) to calculate the movement of grid nodes. The method requires the solution of a system of equations only for the boundary nodes. Displacement-based on RBF was also used by Jakobsson and Amoignon [14]. RBF-based methodologies exhibit small computational cost while maintain the overall quality of the original grid after moving. Morris et al. [15] used a domain element method with a geometric parameterization technique for application to CFD-based aerodynamic optimization. The parameterization uses RBFs to interpolate positions of the domain element and the grid coordinates, to provide simultaneous deformation of the design surface and its corresponding mesh.

Rendall and Allen initially [16] presented an RBF-based method which is independent of connectivity and produces high-quality meshes, besides the fact that it is expensive for large meshes. The efficiency of the technique improved by reducing the number of surface points used to define surface deformations, while the position error was corrected with a simple decaying perturbation, combining a primary basis function method and a secondary local correction method. Furthermore [17], they presented an efficient mesh motion method using RBFs where the surface deformations are smooth, with the computation results demonstrating that greedy algorithms offer an extremely effective method of cost reduction for this approach. They showed that RBF-based mesh deformation is very effective, preserving orthogonality and producing high quality meshes, but potentially expensive owing to the dependence of the entire volume mesh on a large number of surface points. A following study [18] proposed a method that utilizes an error function on the surface mesh to select a reduced subset of the surface points; this subset contains a sufficiently small number of points so as to make the volume deformation fast, while a correction function is used to correct non-included surface points.

Lefrancois [19] proposed the method of "moving submesh approach" (MSA), using the distortion of a pseudo-material in a sparse background grid to significantly reduce the computational cost. The grid is updated using an interpolation technique based on the theory of finite elements. The method is applicable both to structured and unstructured grids. In [20] a very simple low cost algebraic method was presented for the deformation of unstructured grids, which is capable of handling large deformations. In general the algebraic methods, which use simple or more complex interpolation techniques, have the important advantage of low computational cost and are easily adapted for parallel computations.

The motivation behind this work is the evaluation of the mesh deformation methodology based on radial basis functions (RBFs) in cases involving large deformations of unstructured hybrid meshes, typical of those used for the solution of viscous flow equations. Mesh deformation is a key element in various aerodynamic shape optimization methodologies using computational flow dynamics (CFD), as well as in fluid-structure interaction (FSI) methodologies, where the coupled calculations between computational structural mechanics (CSM) and CFD algorithms require the repeated deformation of the computational mesh.

RBF interpolation is used to derive the displacement of the internal fluid nodes given the displacement of the nodes on the solid boundary. There are several types of RBF functions, which are suitable for interpolating multivariate data. They can be divided in two basic groups: functions with compact and functions with global support. In the first case mesh nodes inside a circle (2D) or sphere (3D) with radius *R* around a centre are influenced by the movement of this center. On the contrary, functions with global support are not equal to zero outside a certain support radius but cover the whole interpolation space. In this work several types of RBF are utilized, such as Gaussian, Multiquadric, Inverse Multiquadric and Inverse Quadric, while the complete RBF deformation computation procedure is presented in detail. The grid which is deformed is a 2D hybrid grid around a NACA0012 airfoil, consisting of triangular an quadrilateral elements, a typical grid used for viscous computations around airfoils.

2 Methodology

2.1 RBF interpolation

RBF interpolation can be used to derive the displacement of the internal mesh nodes given the displacement of the mesh nodes on the deformed boundary. The interpolation function, *s*, describing the displacement in the whole domain, can be approximated by a weighted sum of basis functions

$$s(\boldsymbol{x}) = p(\boldsymbol{x}) + \sum_{i=1}^{N} \lambda_i \Phi\left(\|\, \boldsymbol{x} - \boldsymbol{x}_i \,\|\right) \tag{1}$$

where *s* is the RBF, *p* is a low degree polynomial, typically linear or quadratic, λ_i are the RBF coefficients, Φ is a real valued function called the basis function, $\|\boldsymbol{x}\|$ is the Euclidean distance, while $\boldsymbol{x}_i$ are the RBF centres ($\boldsymbol{x}_i$=[x_i, y_i, z_i]) in which the values of the displacement are known, in our case the nodes on the displaced boundary. The RBF consists of a weighted sum of radially symmetric basic functions Φ located at the centers $\boldsymbol{x}_i$ and a low degree polynomial *p*. Given a set of *N* points $\boldsymbol{x}_i$ and values f_i, the interpolation condition is defined as:

$$s(\boldsymbol{x}_i) = f_i \ , \qquad i = 1,\ 2,\ .\ .\ .\ ,\ N \tag{2}$$

The RBF is defined by the coefficients λ_i of the basis function along with the coefficients of the polynomial *p(**x**)*; these are the unknowns of the procedure. In the equation above, f_i are the enforced displacements at the *N* boundary nodes, which serve also as RBF centers, with the additional requirements

$$\sum_{i=1}^{N} \lambda_i q\left(\boldsymbol{x}_i\right) = 0 \tag{3}$$

for all polynomials *q* with a degree less or equal to that of polynomial *p* (with the minimal degree of polynomial *p* depending on the selection of the basis functions Φ). A unique interpolant is given if the basis function is conditionally positive definite function. For conditionally positively defined basis functions with an order $m \leq 2$ a linear polynomial can be used, with a result that rigid body translations are exactly recovered [11].

If we let $\{P_1, \ . \ . \ . \ ,P_L\}$ be a monomial basis for polynomials of the degree of *p*, and $c=(c_1, \ . \ . \ . \ , \ c_l)$ be the coefficients of *p(x)* in terms of this basis, then the interpolation condition can be rewritten in matrix form as a linear system

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix}\begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{4}$$

where ***A*** is the (N x N) interpolation matrix

$$A_{ij} = \Phi\left(\left\|x_{ki} - x_{kj}\right\|\right) \quad 1 \le i, j \le N \tag{5}$$

P is an (N x *4*) matrix defined by the constraint to interpolate exactly all first degree polynomials *p*

$$P = \begin{pmatrix} 1 & x^0_{k1} & y^0_{k1} & z^0_{k1} \\ 1 & x^0_{k2} & y^0_{k2} & z^0_{k2} \\ \cdots & \cdots & \cdots & \cdots \\ 1 & x^0_{kN} & y^0_{kN} & z^0_{kN} \end{pmatrix} \tag{6}$$

the vector ***f*** is given as

$$f = f(x_{ki}) \quad i = 1, \ldots, N \tag{7}$$

c is the vector of the polynomial coefficients, namely

$$p(\boldsymbol{x}) = c_1 + c_2 x + c_3 y + c_4 z \tag{8}$$

where $\boldsymbol{x} = (x, y, z)^{\mathrm{T}}$. This type of approximation is applied to the displacements in each one of the 3 directions *x, y, z*:

$$s_x(\boldsymbol{x}) = c_1^x + c_2^x x + c_3^x y + c_4^x z + \sum_{i=1}^{N} \lambda_i^x \Phi(\| \boldsymbol{x} - \boldsymbol{x}_{ki} \|) \tag{9}$$

$$s_y(\boldsymbol{x}) = c_1^y + c_2^y x + c_3^y y + c_4^y z + \sum_{i=1}^{N} \lambda_i^y \Phi(\| \boldsymbol{x} - \boldsymbol{x}_{ki} \|) \tag{10}$$

$$s_z(\boldsymbol{x}) = c_1^z + c_2^z x + c_3^z y + c_4^z z + \sum_{i=1}^{N} \lambda_i^z \Phi(\| \boldsymbol{x} - \boldsymbol{x}_{ki} \|) \tag{11}$$

2.2 Types of RBFs

Various RBF types are available in the literature suitable for interpolating multivariate data. They can be divided in two groups: *functions with compact* and *functions with global support*. When a RBF with compact support is used the mesh nodes inside a circle (2D) or sphere (3D) with support radius *R* around a center x_j are influenced by the displacement of the corresponding center:

$$\begin{aligned}\Phi(\xi) &= f(\xi), \quad 0 \le \xi \le 1 \\ \Phi(\xi) &= 0, \quad \xi > 1\end{aligned} \tag{12}$$

All compact RBF's are scaled with support radius *R* (*ξ=r/R*). Functions with global support are not equal to zero outside a certain radius; on the contrary they cover the whole interpolation space, which leads to dense matrix systems. In Table 1 various RBFs with compact support are listed [11, 12]. The first four are based on polynomials while the last four are based on the thin plate spline [21]. In Table 2 the RBFs with global support, used in this work are listed too. The MQB and IMQB methods use a parameter *a*, which controls the shape of the basis functions; a large value of a provides a flat function, while a small value gives a narrow cone-like function [11, 12].

No	*Name*	*Φ(ξ)*
1	CP C^0	$(1-\xi)^2$
2	CP C^2	$(1-\xi)^4\,(4\xi+1)$
3	CP C^4	$(1-\xi)^6\,((35/3)\xi^2+6\xi+1)$
4	CP C^6	$(1-\xi)^8(32\xi^3+25\xi^2+8\xi+1)$
5	CTPS C^0	$(1-\xi)^5$
6	CTPS C^1	$1+(80/3)\xi^2-40\xi^3+15\xi^4-(8/3)\xi^5+20\xi^2\log(\xi)$
7	CTPS C_a^2	$1-30\xi^2-10\xi^3+45\xi^4-6\xi^5-60\xi^3\log(\xi)$
8	CTPS C_b^2	$1-20\xi^2+80\xi^3-45\xi^4-16\xi^5+60\xi^4\log(\xi)$

Table 1: RBFs with compact support.

No	*Name*	*Abbreviation*	*Φ(r)*
9	Gaussian	GS	$\exp(-r^2)$
10	Multiquadric	MQB	$(r^2 + a^2)^{1/2}$
11	Inverse Multiquadric	IMQB	$(r^2 + a^2)^{-1/2}$
12	Inverse Quadric	IQB	$(r^2 + 1)^{-1}$

Table 2: RBFs with global support.

2.3 Laplacian smoothing

The Laplacian smoothing operator [22] is used for the improvement of the mesh quality. An iterative procedure takes place for the displacement of the mesh internal nodes. The coordinates of a node's new position are computed as the mean value of the coordinates of the centroids of all the adjacent triangular (2D) or tetrahedral (3D) elements

$$P_N = \sum_{i=1}^{N_E} \left(C_i^E \right) \Big/ N_E \tag{13}$$

where P_N is the corresponding coordinate of node's P new position, N_E is the number of the adjacent elements and C_i^E is the corresponding coordinate of the centroid of the i^{th} element. It is an operation that considerably improves the mesh quality by fixing the shape of highly distorted elements and smoothing the node distribution. The Laplacian smoothing operator consists of the following steps:

1. Construction of proper data structures, used for testing the validity of internal nodes displacement.

2. For each internal node the corresponding elements, connected to the node, are identified and their centroids are computed.

3. The mean values for the spatial coordinates of these centroids are computed, corresponding to the coordinates of node's new position.

4. The validity of the modified mesh is tested. The node is displaced provided that a valid mesh is produced.

5. Steps 2, 3 and 4 are repeated for all internal mesh nodes.

6. The complete procedure may be repeated to produce a smoother mesh.

2.4 Evaluation of mesh quality

Distortion metrics [23] can be used to do more than just determine the final quality of a mesh. They can also be used to guide smoothing operations by constraining the movement during Laplacian smoothing and to drive an optimization-based smoothing. Several metrics can be used for a quantitative evaluation of mesh quality, using various criteria. In this paper the ratio of the radii of the circumscribed and inscribed circles (2D) or spheres (3D) is used as a measure of the quality of each element, which is one of the most commonly used measures of mesh quality for tetrahedral elements [24].

For 3D meshes comprising tetrahedral elements the radius of the inscribed sphere $R_{inscribed}$ is given as [24]:

$$R_{inscribed} = \left(3V \right) \Big/ \left(\sum_{i=1}^{4} A_i \right) \tag{14}$$

where V is the total volume of the tetrahedral element and A_i is the surface of its *i*-th face. The radius of the circumscribed sphere is given as the solution of the following equation [25]:

$$\begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & d_{12}^2 & d_{13}^2 & d_{14}^2 & R_{circ}^2 \\ 1 & d_{21}^2 & 0 & d_{23}^2 & d_{24}^2 & R_{circ}^2 \\ 1 & d_{31}^2 & d_{32}^2 & 0 & d_{34}^2 & R_{circ}^2 \\ 1 & d_{41}^2 & d_{42}^2 & d_{43}^2 & 0 & R_{circ}^2 \\ 1 & R_{circ}^2 & R_{circ}^2 & R_{circ}^2 & R_{circ}^2 & 0 \end{vmatrix} = 0 \tag{15}$$

where R_{circ} is the radius of the circumscribed sphere and d_{ij} is the distance between i and j nodes of the tetrahedron. The corresponding *Ratio* is then defined as:

$$Ratio = R_{circ} / R_{insxribed} \tag{16}$$

For a regular tetrahedron its value is always 3 and the deviation of this value can be used as a measure of the quality of a tetrahedral element, or of the complete mesh by computing the distribution of elements it terms of the aforementioned ratio. For 2D meshes the value of the *Ratio* for an equilateral triangle is equal to 2. R_{circ} and $R_{inscribed}$ are given for triangles as

$$R_{circ} = \frac{(abc)}{\sqrt{(a+b+c)(b+c-a)(c+a-b)(a+b-c)}} \tag{17}$$

$$R_{inscribed} = \frac{1}{2}\sqrt{\frac{(b+c-a)(c+a-b)(a+b-c)}{(a+b+c)}} \tag{18}$$

where *a, b, c* are the lengths of the three triangle edges.

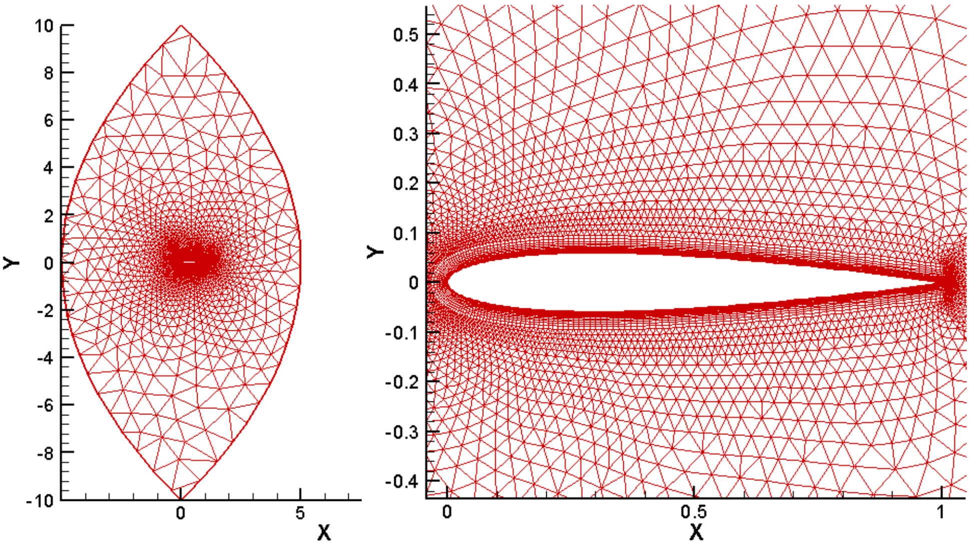

Figure 1: The hybrid 2D grid around a NACA0012 airfoil.

For the hybrid meshes used in this work (Figure 1) the distortion metrics refer only to the triangular (2D) and tetrahedral (3D) elements of the mesh. For the 2D cases considered in this work the percentage of triangular elements having a specific value of the quality metric was computed, using six categories regarding the values of the metric: (2.0-2.1), (2.1-2.2), (2.2-2.3), (2.3-2.4), (2.4-2.5), (>2.5). The more elements belonging to the first category the better the mesh quality, as such elements are closer to the equilateral triangles.

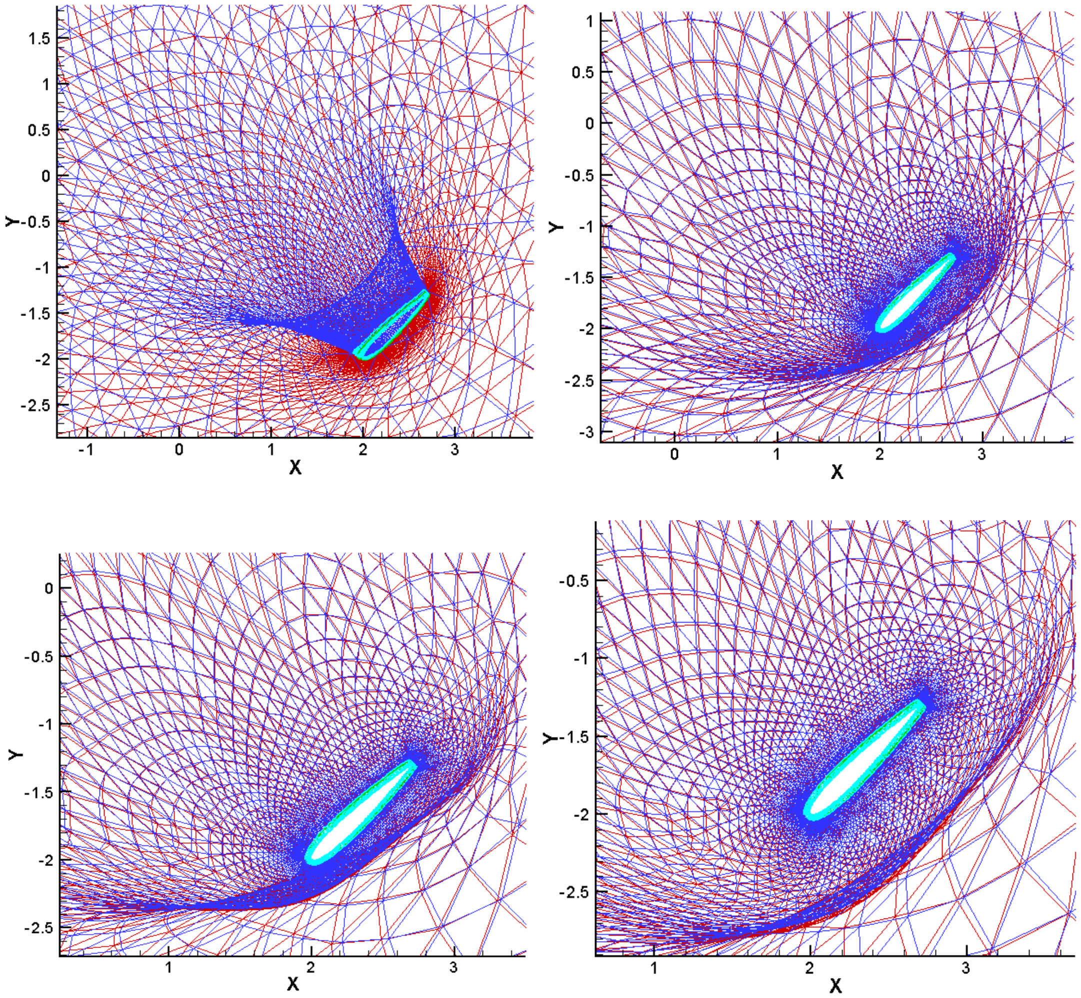

Figure 2: (Left to right - top to bottom: CP C^0, CP C^2, CP C^4, CP C^6) The results of the RBFs with compact support (R=5), for a linear airfoil translation of (2.0, -2.0) and an additional rigid rotation of 45°.

3 Computational results

The hybrid 2D grid was constructed around a NACA0012 airfoil with chord length equal to 1.0. The mesh consists of 6758 nodes, 7492 triangles and 2884 quadrilaterals (Figure 1). In order to obtain a mesh deformation the airfoil was translated by a vector (2.0, -2.0) with an additional rigid rotation of 45° for the RBFs with compact support. The corresponding translation for the RBFs with global support was set equal to (2.1, -0.5) with the additional rigid rotation of 45°. Other values of linear displacement and rotation were also tested, not shown here for brevity reasons. In the following figures the mesh resulting from the basic RBF procedure is shown in blue colour while the one after the Laplacian smoothing procedure is shown in red colour.

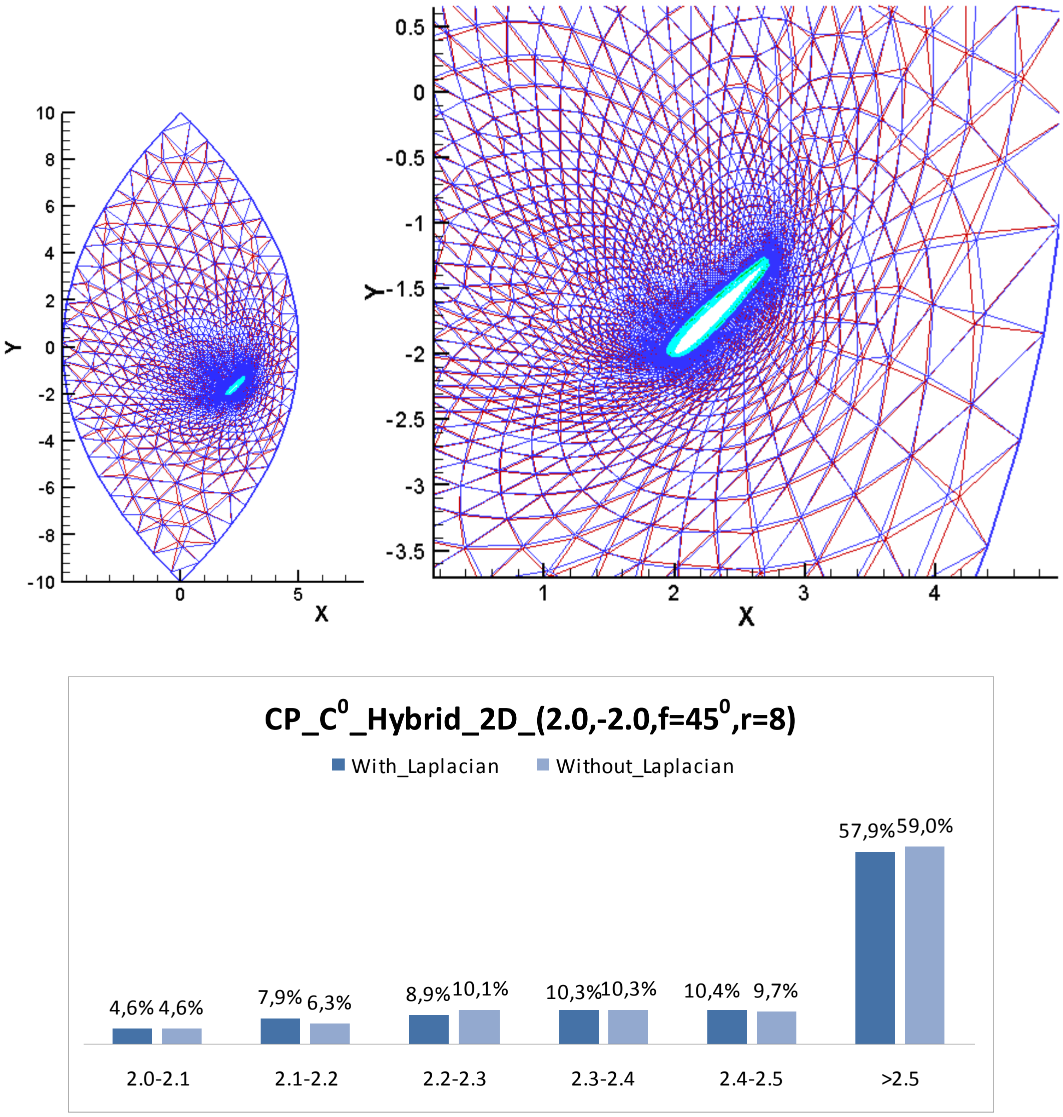

Figure 3: The results of the CP C^0 RBF with *R*=8, for a linear airfoil translation of (2.0, -2.0) and an additional rigid rotation of 45°.

Initially, the four first RBFs with compact support from Table 1 were tested for 3 different *R* values (2.0, 5.0, 8.0). For the first two *R* values (2.0, 5.0), mesh overlapping was observed, causing negative volumes, (Figure 2) and only for the large value of R the overlapping was avoided (Figures 3 - 6). The mesh quality improved with larger values of R, while the Laplacian smoothing procedure improved the mesh quality only for the better initial meshes; the smoothing procedure failed to improve badly distorted meshes as the one in Figure 3. The CP C^6 RBF showed the better behaviour compared to the rest RBFs with compact support for the 2D hybrid mesh.

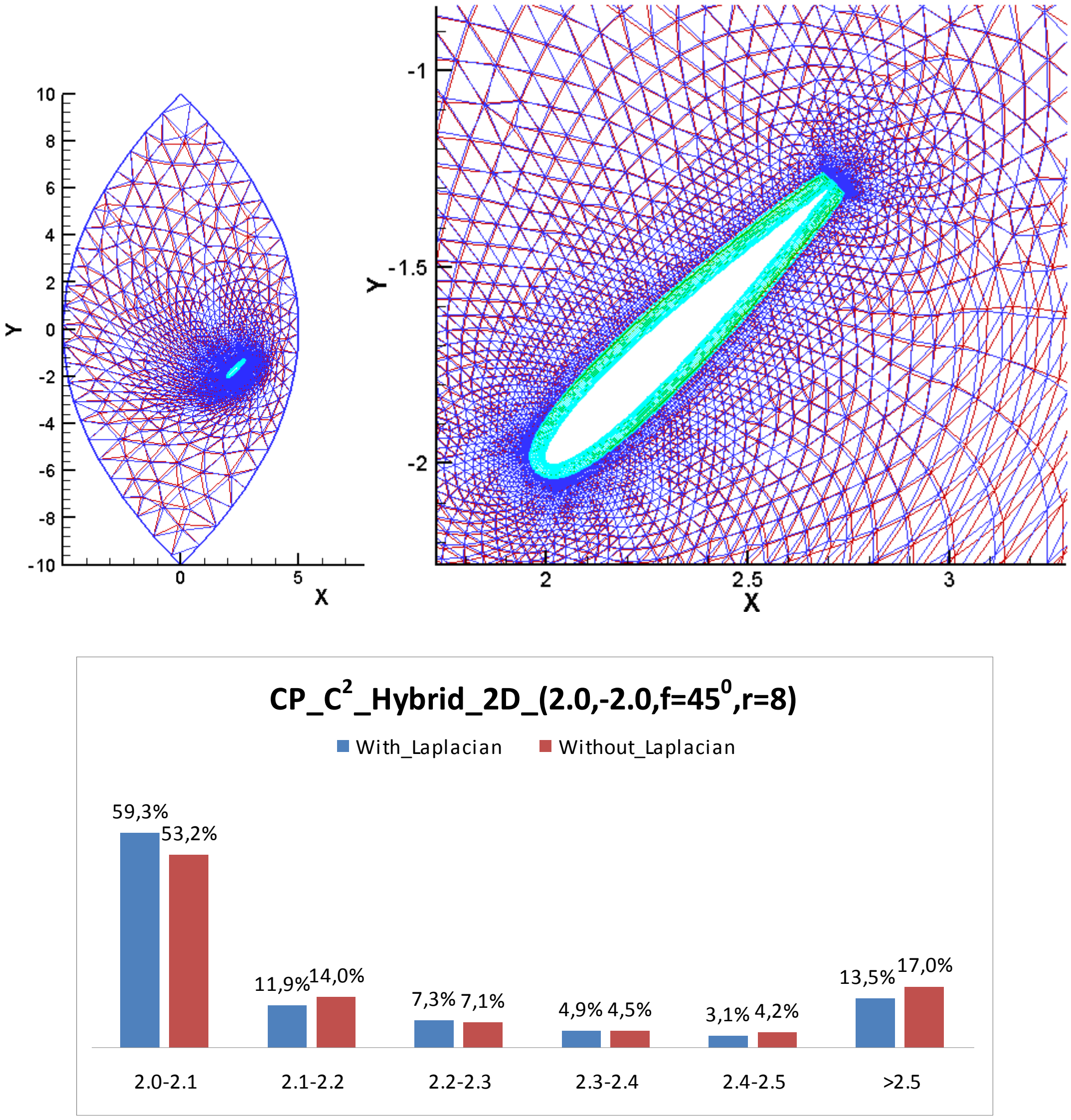

Figure 4: The results of the CP C^2 RBF with R=8, for a linear airfoil translation of (2.0, -2.0) and an additional rigid rotation of 45^o.

Concerning the RBFs with global support, all the functions of Table 2 were tested. Different values of linear and rotational displacements of the boundary nodes were tested, with the results only for the larger displacements presented here for brevity (Figures 7-10). The Gaussian RBF (Figure 7), resulted in overlapping elements, although the linear displacement of the boundary nodes was smaller than the one used for the RBFs with compact support. The Inverse Quadric RBF showed a slightly better behaviour for small boundary displacements but for the larger ones (Figure 8) the results were poor. Although the quality metrics are better than those of the Gaussian one, the final mesh has very dense regions, which are not acceptable for computational purposes. Inverse Multiquadric RBF showed a better behaviour for small and large displacements of the boundary nodes (Figure 9), while the Multiquadric RBF resulted in lower quality meshes (Figure 10). Similarly to the Gaussian one the Multiquadric RBF resulted in a non uniform mesh density around the displaced airfoil, although with better results than the former.

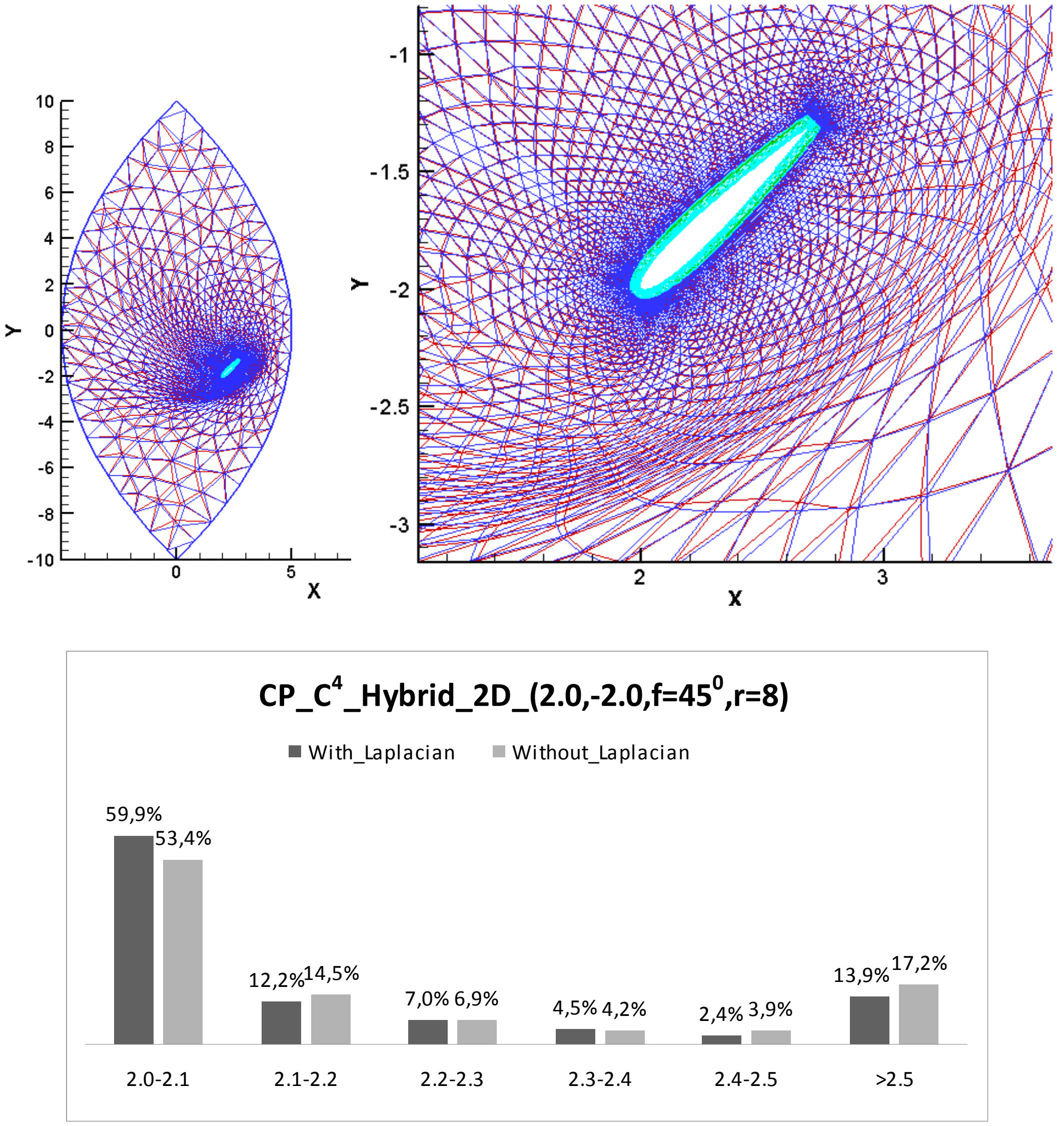

Figure 5: The results of the CP C^4 RBF with *R*=8, for a linear airfoil translation of (2.0, -2.0) and an additional rigid rotation of 45^o.

All the functions tested in this work succeeded in maintaining the shape of the grid close to the displaced airfoil almost unaffected (the one consisting of quadrilateral elements), which is very important for the correct computation of the viscous effects on the solid wall. The use of Laplacian smoothing was generally effective, although its effectiveness was not the same for all the cases considered. The Laplacian smoothing resulted in mesh quality improvements only when it was applied in good initial meshes. It cannot be used to cure overlapping elements or very distorted meshes with highly non-uniform densities.

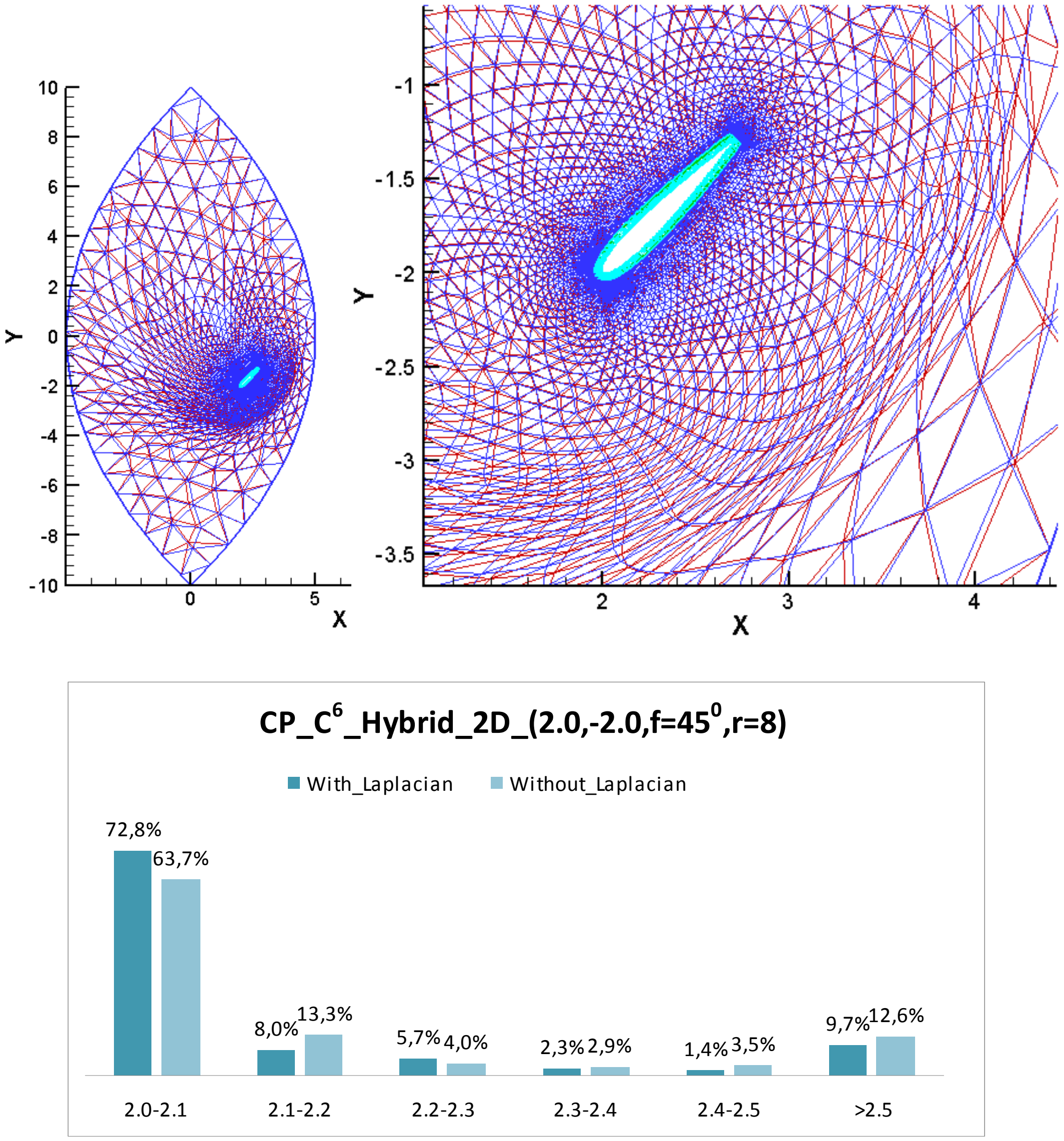

Figure 6: The results of the CP C^{6} RBF with *R*=8, for a linear airfoil translation of (2.0, -2.0) and an additional rigid rotation of 45^{o}.

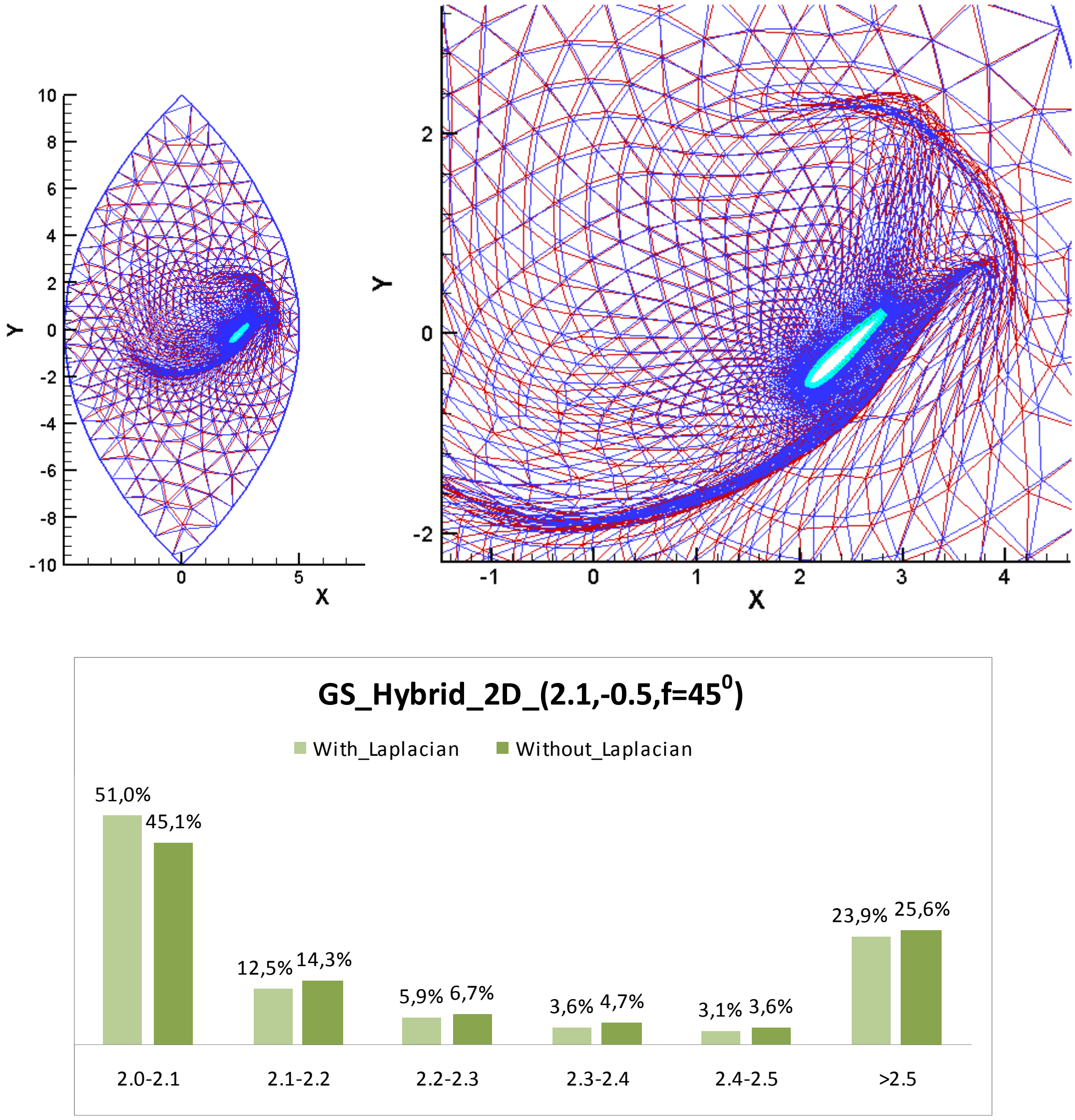

Figure 7: The results of the GS RBF for a linear airfoil translation of (2.1, -0.5) and an additional rigid rotation of 45^o.

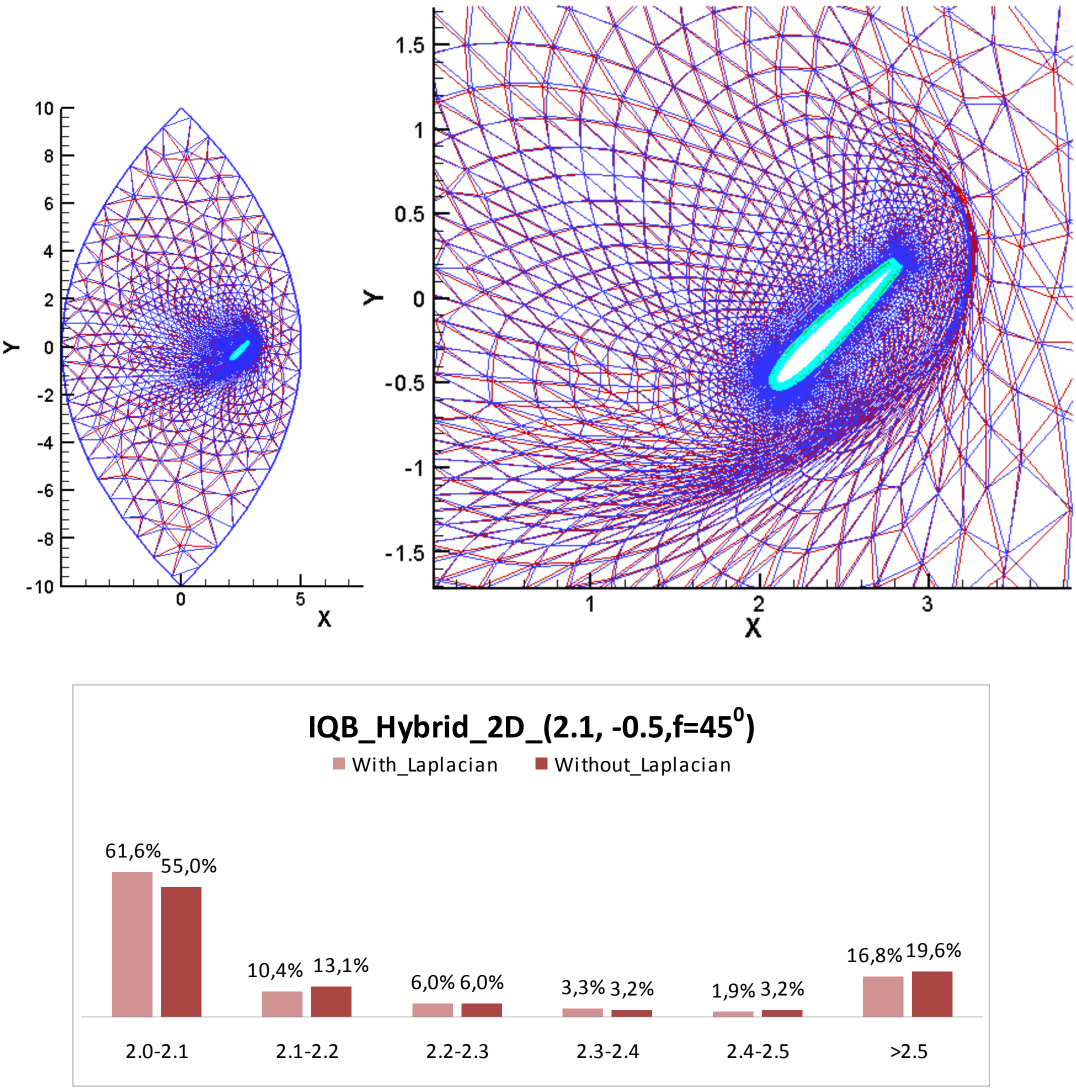

Figure 8: The results of the IQB RBF for a linear airfoil translation of (2.1, -0.5) and an additional rigid rotation of 45^{o}.

4 Conclusions

Concerning the two-dimensional hybrid meshes, for the RBFs with compact support, as the radius of influence *R* increases the mesh quality also increases, and the possibility of producing cells with negative volumes decreases. The use of Laplacian smoothing generally improves the mesh quality, but not in the same degree for all different RBF functions tested. The Laplacian smoothing cannot be effectively used to cure overlapping or highly distorted meshes.

Better results were obtained using RBFs with global support compared to those of RBFs with compact support and small values of *R*. However, compared to RBFs with compact support and large *R* value the latter provided smoother mesh deformations and higher values of the corresponding metrics. For RBFs with global support the Gaussian RBF may cause negative volumes in large mesh deformations, while the inverse quadric RBF provides poor results, with the inverse multiquadric RBF to be the best choice concerning the resulting mesh quality.

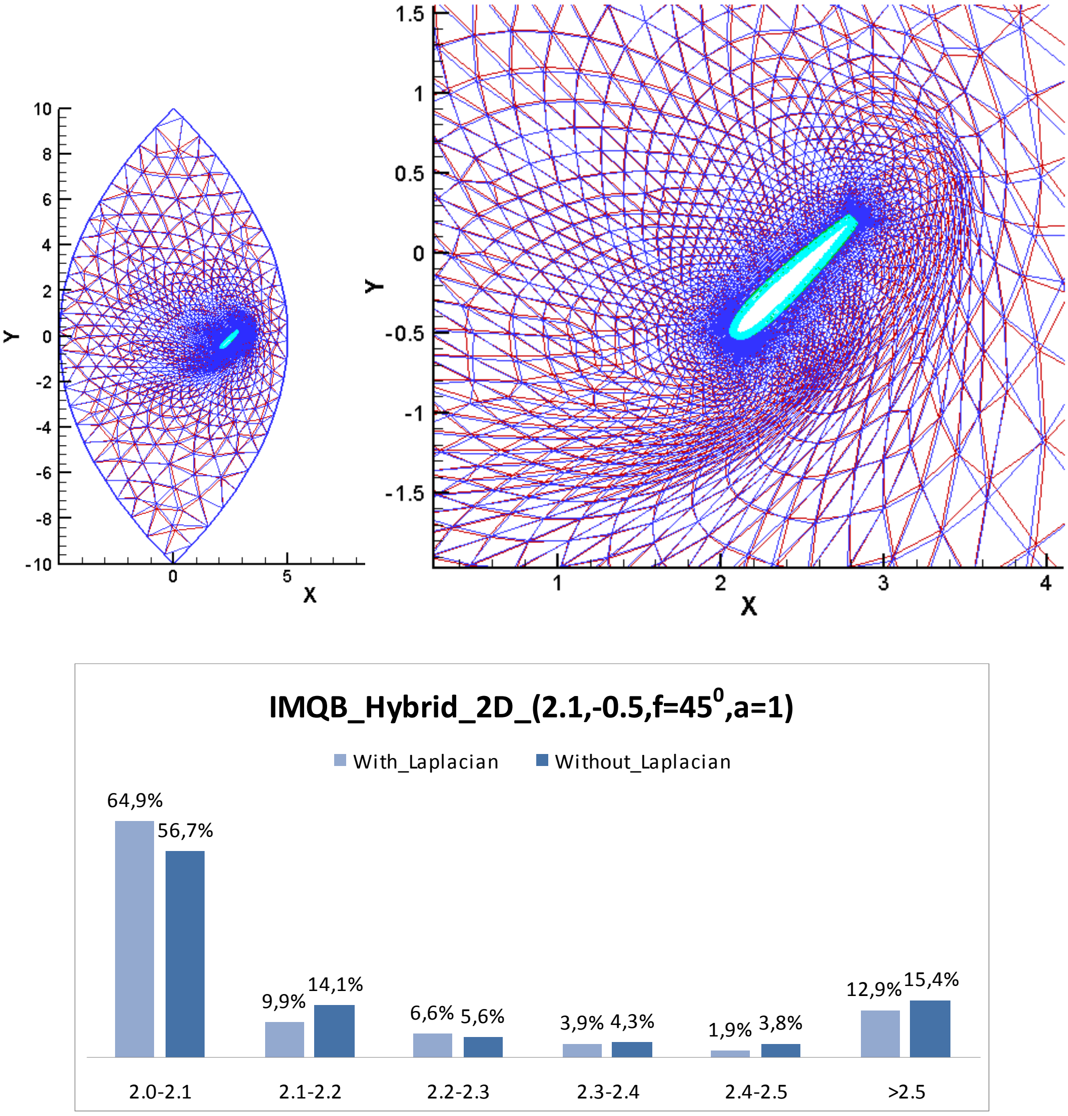

Figure 9: The results of the IMQB RBF for a linear airfoil translation of (2.1, -0.5) and an additional rigid rotation of 45^{o}.

The computational time of the procedure depends on the number of the boundary nodes. For two-dimensional meshes this can be kept as low as a few seconds. However for three-dimensional meshes that have been also tested (not presented here) with a large number of nodes on the deformed boundary the computational time increases considerably and a percentage of those boundary nodes should be used as RBF centres in order to decrease the computational effort.

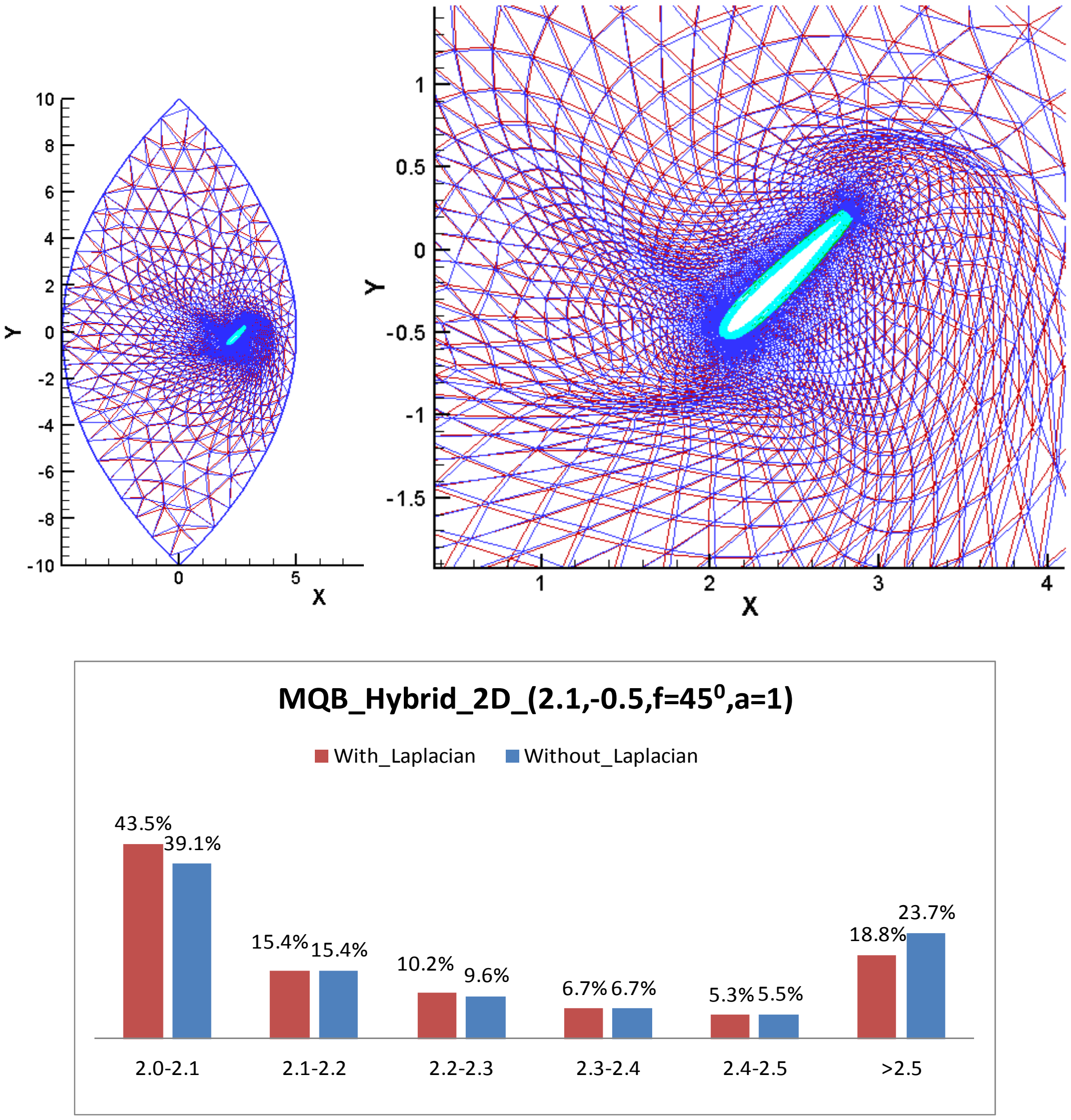

Figure 10: The results of the MQB RBF for a linear airfoil translation of (2.1, -0.5) and an additional rigid rotation of 45^{o}.

References

[1] C. Farhat, "CFD on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization", International Journal of Computational Fluid Dynamics, 19(8), 595-603, 2005.

[2] J.T. Batina, "Unsteady Euler algorithm with unstructured dynamic mesh for complex-aircraft aeroelastic analysis", AIAA paper AIAA-89-1189, 1989.

[3] B.A. Robinson, J.T. Batina, H.T.Y. Yang, "Aeroelastic analysis of wings using the Euler equations with a deforming mesh", Journal of Aircraft, 28(11), 781–788, 1991.

[4] B.T. Helenbrook, “Mesh deformation using the biharmonic operator”, International Journal for Numerical Methods in Engineering, 56, 1007–1021, 2003.

[5] P.G.A. Cizmas, J.I. Gargoloff, “Mesh generation and deformation algorithm for aeroelasticity simulations”, Journal of Aircraft, 45(3), 1062-1066, 2008.

[6] D.M. Schuster, J. Vadyak, E. Atta, “Static aeroelastic analysis of fighter aircraft using a three-dimensional Navier–Stokes algorithm”, Journal of Aircraft, 27(9), 820–825, 1990.

[7] M.K. Bhardwaj, K. Kapania, E. Reichenbach, G.P. Guruswamy, “Computational fluid dynamics/computational structural dynamics interaction methodology for aircraft wings”, AIAA Journal, 36(12), 2179–2186, 1998.

[8] L.E. Eriksson, “Generation of boundary-conformation grids around wing-body configurations using transfinite interpolation”, AIAA Journal, 20(10), 1313–1320, 1982.

[9] P.M. Hartwich, S. Agrawal, “Method for perturbing multiblock patched grids in aeroelastic and design optimization applications”, AIAA Paper AIAA-97-2038, 1997.

[10] M.A. Potsdam, G.P. Guruswamy, “A parallel multiblock mesh movement scheme for complex aeroelastic applications”, AIAA paper AIAA-2001-0716, 2001.

[11] A. de Boer, M. van der Schoot, H. Bijl, “New method for mesh moving based on Radial Basis Function interpolation”, in "Proceedings of the European Conference on Computational Fluid Dynamics", ECCOMAS CFD 2006.

[12] A. de Boer, M. van der Schoot, H. Bijl, “Mesh deformation based on Radial Basis Function interpolation”, Computers and Structures, 85, 784–795, 2007.

[13] A.H. Van Zuijlen, A. de Boer, H. Bijl, “Higher - order time integration through smooth mesh deformation for 3D fluid–structure interaction simulations”, Journal of Computational Physics, 224, 414–430, 2007.

[14] S. Jakobsson, O. Amoignon, “Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization”, Computers & Fluids 36, 1119–1136, 2007.

[15] A.M. Morris, T.C.S. Rendall, C.B. Allen, “CFD-based optimization of aerofoils using radial basis functions for domain element parameterization and mesh deformation”, International Journal for Numerical Methods in Fluids 58, 827–860, 2008.

[16] T.C.S. Rendall, C.B. Allen, “Efficient mesh motion using radial basis functions with data reduction algorithms”, Journal of Computational Physics, 228, 6231–6249, 2009.

[17] T.C.S. Rendall, C.B. Allen, “Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors”, International Journal for Numerical Methods in Engineering, 81, 89–105, 2010.

[18] T.C.S. Rendall, C.B. Allen, “Reduced surface point selection options for efficient mesh deformation using radial basis functions”, Journal of Computational Physics, 229, 2810–2820, 2010.

[19] E. Lefrancois, “A simple mesh deformation technique for fluid–structure interaction based on a submesh approach”, International Journal for Numerical Methods in Engineering, 75, 1085–1101, 2008.

[20] T. Gerhold, J. Neumann, “The Parallel Mesh Deformation of the DLR TAU-Code”, In “New Res. in Num. and Exp. Fluid Mech. VI, NNFM 96”, C. Tropea et al. (Eds.), 162–169, 2007.

[21] R. Franke, “Scattered data interpolation: tests of some methods”, Mathematics of Computation. 38, 181-200, 1982.

[22] L.R. Hermann, “Laplacian-isoparametric grid generation scheme”, ASCE Journal of the Engineering Mechanics Division, 102 (5), 749-756, 1976.

[23] S.A. Canann, J.R. Tristano, M.L. Staten, “An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes”, in "Proceedings of the International Meshing Roundtable", 479-494, 1998.

[24] W.-Y. Choi, D.-Y. Kwak, I.-H. Son, Y.-T. Im, “Tetrahedral mesh generation based on advancing front technique and optimization scheme”, International Journal for Numerical Methods in Engineering, 58, 1857-1872, 2003.

[25] L.M. Blumenthal, "Distance Geometry", Chelsea, 1970.