European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)

UNSTRUCTURED REMESHING USING AN EFFICIENT SMOOTHING SCHEME

Panagiotis I. K. Liakopoulos* **and Kyriakos C. Giannakoglou***

*National Technical University of Athens,
9, Iroon Polytechniou str., 157 80 Athens, Greece
e-mail: liakop@mail.ntua.gr
web page: http://velos0.ltt.mech.ntua.gr/research/

Key words: Unstructures grids, mesh movement, mesh smoothing, Laplacian smoothing, mesh untangling

Abstract. *In this paper, a method for adapting existing 2D and 3D unstructured meshes to modified domain boundaries is presented. The method simultaneously employs mesh smoothing, moving and untangling. It should be considered as an alternative to mesh regeneration often used in either optimization or unsteady flow computations with moving boundaries. With respect to the widely used Laplacian smoothing scheme, the proposed algorithm additionally employs a geometric constraint which limits the allowed moving space for each vertex and manages to overcome the generation of invalid/inverted elements even in the case of extreme boundary deformations. The resulting algorithm is easy to implement and has low computational cost. Here, it is demonstrated in 2D and 3D meshes under deformation, with triangular and tetrahedral elements respectively, including cascade meshes with differently treated periodic boundaries.*

1 INTRODUCTION

In computational mechanics, there is often the need for adapting an existing mesh to the modified boundaries of the solution domain. A typical example is the coupling of shape optimization algorithms with computational mechanics software (computational fluid dynamics, CFD, in our case); different candidate shapes are generated during the search for the optimal solution and these need to be meshed prior to resorting to the CFD software for their evaluation. Another example, though less troublesome due to the limited and controllable change in geometry between successive time-steps, is that of the numerical solution of unsteady flows in domains with moving boundaries, such as a multi-element airfoil or wing with moving slats and/or flaps; at each time-step, the mesh needs to be adapted to the new boundaries.

A typical way to satisfy the need of mesh adaptation to an updated domain is by regenerating it from scratch. Working with unstructured meshes, this is not smart at all;

the new mesh may probably have different number of nodes and different connectivity, so the interpolation of field quantities to the new mesh nodes introduces inaccuracies. In 3D domains, in particular, the mesh generation might be costly enough. For these reasons, efficient[1,2,3] and effective[4,5] methods which adapt/warp the existing mesh to the new boundary have been developed. To maintain the mesh quality, which may severely be damaged for pronounced boundary shape changes, an efficient mesh warping method should also perform mesh smoothing and untangling. The latter is necessary since a considerable change in the domain shape often yields invalid/inverted mesh elements.

In the literature, several approaches for triangular and tetrahedral mesh smoothing can be found; they range from simple ones, such as "plain"[1] or "smart"[2] Laplacian smoothing and angle based smoothing[3] up to complex ones such as optimization-based algorithms.[4] The advantage of the simpler methods is the low CPU cost whereas the complex ones are more robust, ensuring valid final meshes to be created. In the case of moving boundaries, an application of the spring analogy is frequently used; either linear or torsional springs or the combination of both[5] can be used. The existing methods perform mesh untangling after the creation of invalid elements which occurs during the change in the domain boundary shape.

The present work proposes a smoothing method which combines the simplicity of the Laplacian smoothing, overcomes the weakness of generating inverted elements, has low CPU cost and is easier to implement compared to complex methods such as optimization-based smoothing. $2D$ and $3D$ cases are presented to demonstrate the effectiveness of the proposed algorithm. Emphasis is put to cascade computational domains which additionally involve periodic boundaries.

2 LAPLACIAN SMOOTHING IN 2D

Laplacian smoothing is widely being used as a mesh smoothing method or for adapting an existing mesh to a modified boundary. It is based on the iterative movement of vertices towards the centroid of the polygon formed by the adjacent vertices[1] . Let $\overrightarrow{r}_i$ be the position vector of the ith internal vertex and $\overrightarrow{r}_j, j = 1...n_i$ the position vectors of the n_i vertices linked to i through an edge. The centroid m_i of the polygon surrounding $\overrightarrow{r}_m$ is located at

$$\overrightarrow{r}_{m_i} = \frac{1}{n_i} \sum_{j=1}^{n_i} \overrightarrow{r}_j \tag{1}$$

and the Laplacian smoothing updates $\overrightarrow{r}_i$ as follows (where ω is a relaxation factor).

$$\overrightarrow{r^*}_i = (1-\omega)\overrightarrow{r}_i + \omega \overrightarrow{r}_{m_i} \tag{2}$$

An often reported weakness of the Laplacian smoothing, though, is that it may produce invalid/inverted elements in the final mesh. An illustrative example is shown in fig. 1, right, where the centroid of the surrounding polygon lies outside the polygon. In such a

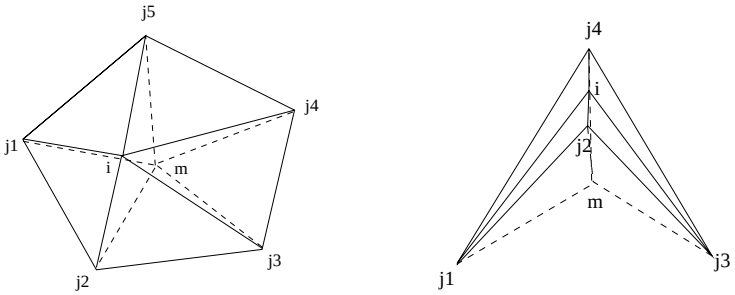

Figure 1: Left: The centroid of the surrounding polygon for the ith internal vertex. Right: The centroid of the surrounding polygon lies outside the polygon and this might transform valid triangles to inverted ones.

case, moving the ith vertex towards m_i would inevitably produce invalid elements so the need arises for a variant of the method which is capable of overcoming this problem.

In the past, a remedy to the weakness of the Laplacian smoothing through a modified algorithm which is often referred to as "smart" Laplacian smoothing[2] has been proposed. In this algorithm, at each iteration step, the movement of the ith vertex depends on the improvement or degradation of the quality of the elements that compose the surrounding polygon. For each node, its movement is allowed only if the quality of these elements improves. In the present work, an alternative approach is used through the definition of a modified polygon, as explained below.

3 DEFINING A MODIFIED SURROUNDING POLYGON

In order to safeguard the smoothing procedure and avert the invalid element creation, a definition of a modified polygon is proposed which prevents valid elements from inverting and leads to a compression of the already inverted elements' area.

Instead of the polygon defined by the adjacent vertices of i (fig. 1), a new polygon is formed by shrinking some edges and considering hypothetical vertices closer to i, at distance equal to the minimum length r_{ij} of all segments emanating from i (fig. 2). By doing so, the centroid of the new polygon lies always inside it, fig.2, when the triangles composing the polygon are not inverted. In case of inverted elements the same algorithm applies untangling them and producing valid ones.

The coordinates of the new (constrained) polygon's vertices are

$$\overrightarrow{r}_{j'} = \overrightarrow{r}_i + r_{i,min} \frac{\overrightarrow{r}_{ij}}{|\overrightarrow{r}_{ij}|} \tag{3}$$

where

$$r_{i,min} = min(|\overrightarrow{r}_{ij}|),\ j = 1...n_i \tag{4}$$

with the location of the constrained polygon's centroid $\overrightarrow{r}_{m_i'}$ given by eq. (1) in which $\overrightarrow{r}_{j'}$ is used instead of $\overrightarrow{r}_j$. Note that relaxation, eq. 2, can also be used, if necessary.

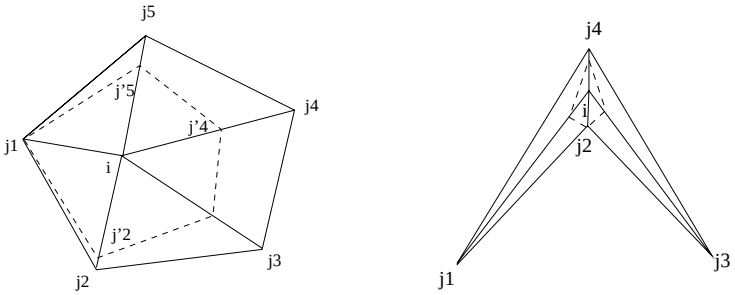

Figure 2: Left: The new constrained surrounding polygon. Right: Even in extreme cases, the feasible area lies within the surrounding polygon even in extreme cases.

The alteration of the control polygon safeguards valid elements from inverting by allowing them to move within a feasible area which is always within the polygon defined by the surrounding vertices (fig. 2, right).

The proposed algorithm performs also angle smoothing similar to[3] , since it moves the ith vertex so that the angles of the constrained polygon be adequately smoothed; thus, angle smoothing bears no additional CPU cost.

No other method to untangle inverted elements is necessary; the use of the centroid of the modified/constrained polygon proves satisfactory since a swift movement of the inverted element's vertices towards the centroid reduces the overall negative area (inverted elements) and aids the smoothing procedure to produce a valid mesh.

4 APPLICATIONS IN 2D CASCADES

The proposed method is used in two cascade applications with modified computational domains. Extension to $3D$ meshes follows in a subsequent section.

A quality metric is used to estimate the element quality of $2D$ triangular meshes prior and after remeshing. In particular, the mean-ratio metric[6] which is invariant under translation, rotation, reflection and uniform scaling of triangles[7,8] is used. The quality of a triangle q_e, scaled to the interval (0,1] (note that $q_e = 1$ for equilateral triangles), equals to

$$q_e = \frac{2}{3} \sum_{i=1}^{3} \mu_i \tag{5}$$

where

$$\mu_i = \mu(S_i) = \frac{det(S_i)}{\|S_i\|_F^2} \quad where \quad \|S\|_F = \sqrt{tr(S_i^T S_i)} \tag{6}$$

S_i is the product of the Jacobian matrix of each triangle and the inverse Jacobian matrix W^{-1} of the equilateral triangle whereas $\| \ \|_F$ is the Frobenius norm. So, for node i of triangle ijk S_i is defined by

$$S_i = A_i W^{-1} = \begin{bmatrix} x_j - x_i & x_k - x_i \\ y_j - y_i & y_k - y_i \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} \end{bmatrix}^{-1} \tag{7}$$

4.1 A case with fixed periodic boundaries

As a first example, the movement of an airfoil of a compressor cascade and the subsequent remeshing are examined. The airfoil is moved pitchwise and its new position is extremely close to the periodic boundary which remains fixed. For its new position, the proposed method is applied in order to produce a valid unstructured mesh. The initial mesh has a mean quality value $q_e = 0.915$ and standard deviation 0.088.

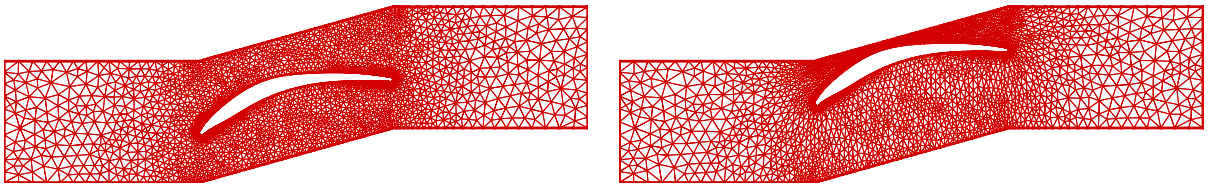

Figure 3: Left: Initial unstructured mesh for CFD computations in a $2D$ compressor cascade. Right: By pitchwise moving the airfoil and maintaining the periodic boundaries intact, the present method yields a valid mesh.

The result is shown in fig. 3, right, and is a valid mesh which, as expected, includes compressed elements in the narrow area formed between the airfoil and its closer boundary and expanded ones in the opposite area. The final mesh is worse than the initial one since it has $q_e = 0.812$ and $\sigma_e = 0.164$. The drop in the average mesh quality is due to the poor quality of the elements in the compressed area.

4.2 A case with free periodic boundaries

Because of the low element quality within the compressed area, the periodic boundaries of the cascade are, then, left free to move along with the airfoil. They are constrained, however, so as to remain periodic. It is clear that such a treatment increases the flexibility since it allows the airfoil to move even at a position which would, otherwise, itersect the (fixed) periodic boundaries. On the other hand, the pitchwise distances between the airfoil sides and the new periodic boundaries are expected to be evenly distributed, allowing triangles of higher quality to be formed. The final mesh, fig. 4, has a mean quality value $q_e = 0.840$ and standard deviation 0.154. In this case, there is a significant recovery of quality.

An even higher mesh quality can be achieved by letting the inlet and outlet cascade boundaries free to move pitchwise, as shown in fig. 5. The final mesh has $q_e = 0.909$ and $\sigma_e = 0.095$. By allowing the pitchwise movement almost full quality recovery is achieved (fig. 6).

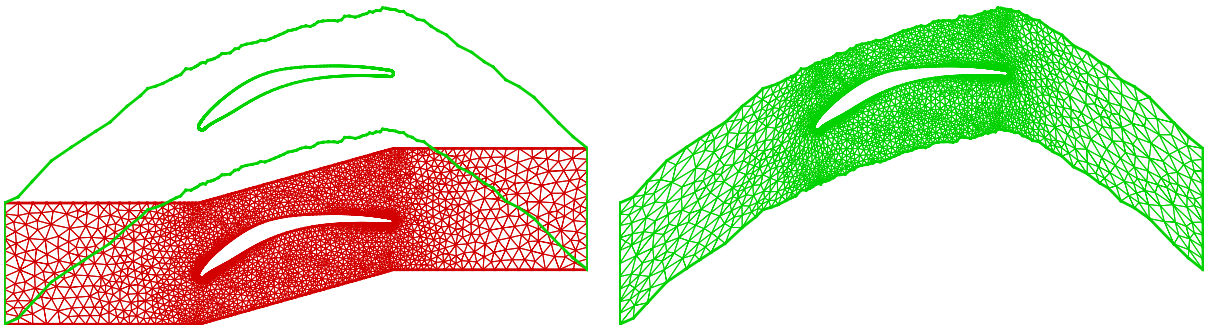

Figure 4: Left: Initial unstructured mesh for CFD computations in a $2D$ compressor cascade along with the final position of the cascade boundaries. Right: By pitchwise moving the airfoil, the periodic boundaries follow and a valid mesh of higher quality is finally produced.

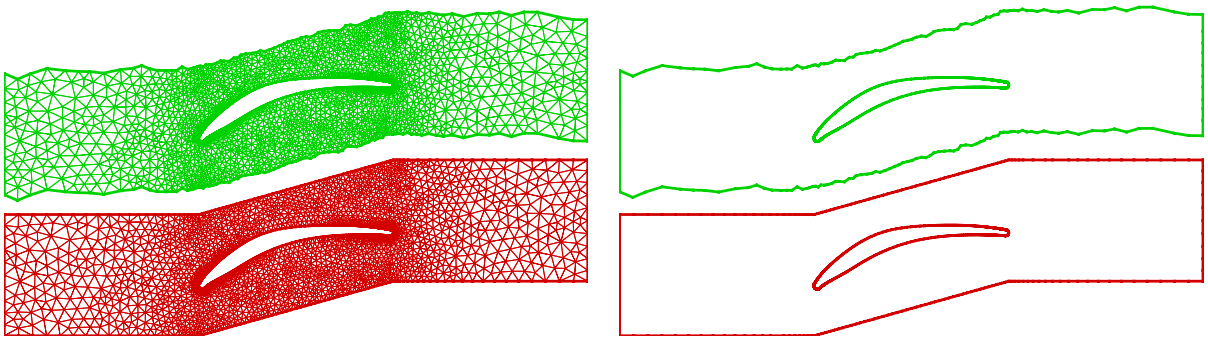

Figure 5: Left: Unstructured initial and final meshes in the compressor cascade. The periodic, inlet and outlet boundaries can move along with the airfoil. Right: Initial and final position of the boundaries of the two meshes, respectively.

4.3 An "extreme" case with unrealistic airfoil thickening

The last demonstration is an extreme case which mimics the situation that often occurs during, for instance, aerodynamic shape optimization problems handled by any stochastic method such as evolutionary algorithms. The randomized change of design parametric values may unexpectedly create unrealistic shapes, in particular during the early stages of the optimization task. In the present example, we exaggerate by creating a really thick "airfoil" (fig. 7) with maximum thickness slightly less than the cascade pitch. As it can be seen from this figure, the gap between the airfoil contour and the periodic boundaries is very thin and this occurs for a considerable chord length percentage. The proposed method is used for remeshing; the inlet,outlet and periodic boundaries are free to move.

As it was to be expected the final mesh , fig. 7, has a low mean quality value of $q_e = 0.443$ and $\sigma_e = 0.346$ which was due to the extreme compression of the space between the airfoil and the periodic boundaries. Nevertheless, the method achieves in producing a valid mesh.

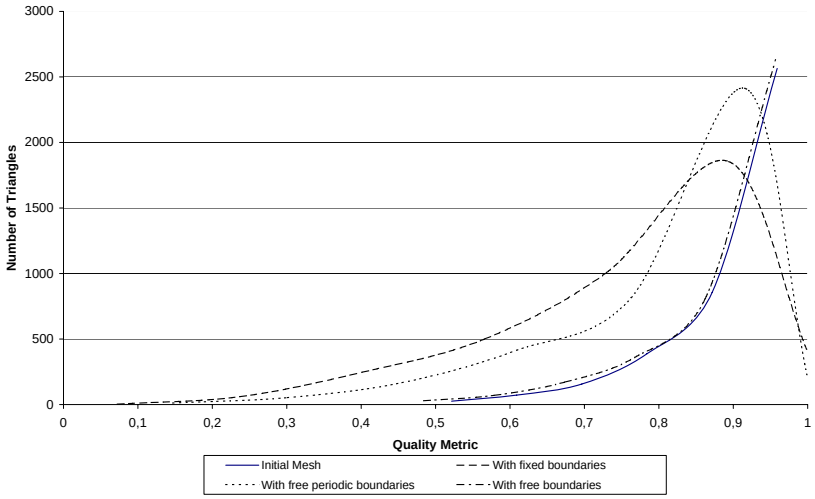

Figure 6: The quality distributions for the initial and the finsl meshes after movement with fixed boundaries, free periodic boundaries and free periodic boundaries with pitchwise inlet and outlet movement. The quality is recovered as the boundary is allowed to move along with the airfoil.

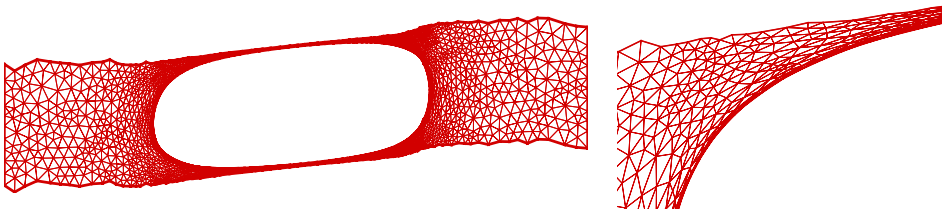

Figure 7: Left: Unstructured valid mesh with unrealistic airfoil thickening, after applying the proposed method. Right: A magnification of a mesh area close to the periodic boundary which shows the compressed elements formed between the periodic boundary and the airfoil.

5 EXTENSION TO 3D MESHES

The extension of the algorithm to $3D$ meshes is almost "straightforward" provided that, instead of the surrounding polygon, the surrounding polyhedron for each vertex is used. The same normalization applies to the modified polyhedron. $3D$ cases, though, weren't as simple as $2D$ ones. In almost every case, the application of the proposed algorithm left approximately 0.1% of inverted elements. The remedy to that problem is the use of an additional check before adjusting the coordinates of the ith vertex. The new check is similar to the one utilized whithin by the "smart" Laplacian algorithm[2] . The idea is to avoid creating more inverted elements, so if the movement of the ith vertex towards the polyhedron's centroid produces more inverted elements, the relaxation factor ω is cut in half.

An acceptable value of the relaxation factor ω^* is determined by a subdividing checking

loop which stops when the value of the relaxation factor is such that it doesn't invert any of the valid tetrahedra within the polyhedron. So

$$\omega^* = \frac{\omega}{2^n} \tag{8}$$

where n the number of cycles of the checking loop and ω the "basic" relaxation factor.

The mean-ratio metric[6] extends to $3D$ where the quality q_e of a tetrahedron, scaled to the interval (0,1], equals to

$$q_e = \frac{3}{4} \sum_{i=1}^{4} \mu_i \tag{9}$$

where

$$\mu_i = \mu(S_i) = \frac{det(S_i)^{\frac{2}{3}}}{\|S_i\|_F^2} \quad where \quad \|S\|_F = \sqrt{tr(S^T S)} \tag{10}$$

where, for each tetrahedral vertex,

$$S_i = A_i W^{-1} = \begin{bmatrix} x_j - x_i & x_k - x_i & x_l - x_i \\ y_j - y_i & y_k - y_i & y_l - y_i \\ z_j - z_i & z_k - z_i & z_l - z_i \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{6} \\ 0 & 0 & \frac{\sqrt{2}}{\sqrt{3}} \end{bmatrix}^{-1} \tag{11}$$

6 3D MESH APPLICATIONS

In order to demonstrate the effectiveness of the proposed algorithm, two $3D$ cases are presented. Remeshing is required due to the shape modification of an axisymmetric converging-diverging duct and pitchwise movement of the blade in a wedge cascade.

6.1 Shape transformation of a converging-diverging duct

For the definition of the new shape (fig. 8, right) a transformation of the starting duct with circular cross-section (fig. 8, left) is applied; practically along its axis the duct cross-section is transformed from circular to elliptic. The initial mesh has a mean quality value $q_e = 0.810$ and standard deviation $\sigma_e = 0.113$.

The lateral compression of the duct surface, while maintaining the starting internal node distribution, created inverted tetrahedra close to the moving boundaries which were iteratively eliminated, as can be seen in fig. 9. The final mesh has $q_e = 0.823$ and $\sigma_e = 0.116$ which shows that there is a complete quality recovery after the boundary's deformation.

6.2 Mesh movement in a wedge linear cascade.

In this case, the movement of a blade close to a linear cascade boundary is presented, fig. 10. The pitchwise movement of the wedge near the periodic boundaries forces the mesh to adapt around it and eliminate any invalid elements. The wedge shape with the

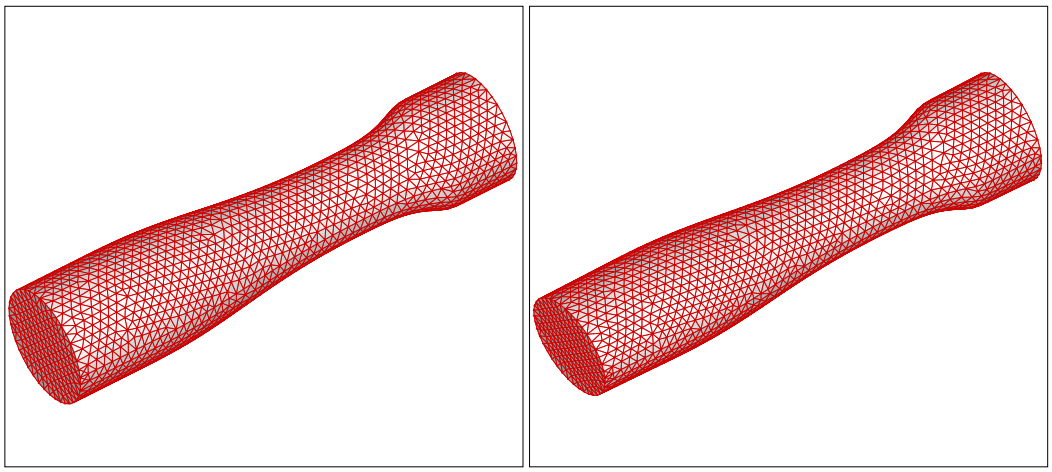

Figure 8: Tetrahedral meshes for the initial (left) and the modified (right) duct shape.

sharp angles is a perfect case to test the algorithm since Laplacian smoothing often fails at boundary elements that form very large or very small dihedral angles. The initial mesh has $q_e = 0.810$ and $\sigma_e = 0.113$. The final mesh has $q_e = 0.777$ and $\sigma_e = 0.153$, i.e. worse values, due to the intense movement of the wedge towards the boundary that produced numerous compressed elements of poor quality between the wedge and the periodic boundary. Apart from the compressed area, though, there is a quality recovery and a successful mesh movement.

7 CONCLUSIONS

A new mesh adaptation/smoothing/untangling method has been proposed. The proposed algorithm has a low CPU cost and is easy to apply. Several cases have been presented to demonstrate the algorithm's abilities in which the algorithm manages to produce valid meshes in a variety of boundary movements even in extreme cases. The extension to $3D$ needs an additional check similar to the one applied in "smart" Laplacian smoothing. On top of producing a valid mesh, the method succeeds in recovering the mesh quality that is degraded through intense boundary movements. In cascade meshes, the careful treatment of periodic boundaries may increase the quality of the final mesh.

8 ACKNOWLEDGEMENTS

This work was funded by the PENED01 program (Measure 8.3 of the Operational Program Competitiveness, of which 75% is European Commission and 25% national funding) under project number 01ED131.

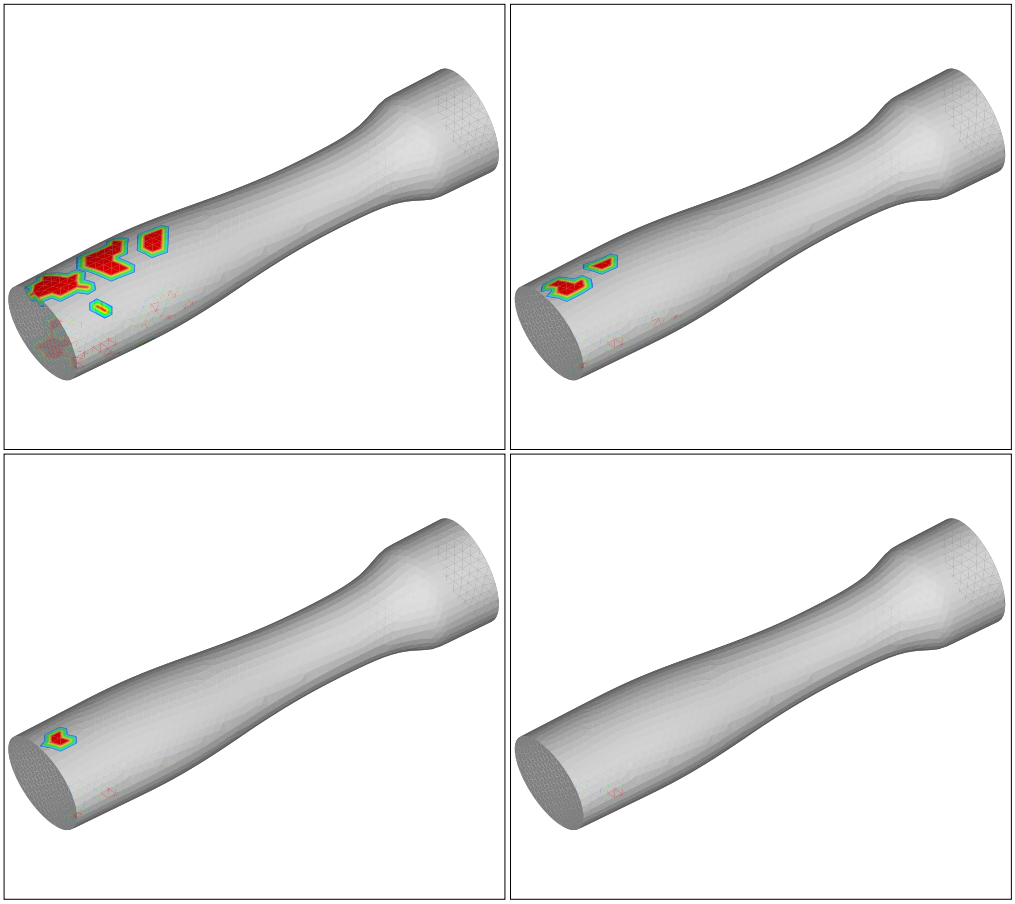

Figure 9: Eliminating inverted elements in the duct case. Areas with inverted elements are marked. The first to fourth iterations of the smoothing procedure are depicted from left to right and from top to bottom.

REFERENCES

[1] W. R. Buell and B. A. Bush, *Mesh Generation - A Survey*, J. Engng. Ind., Trans. ASME, pp. 332-338, (1973).

[2] L. A. Freitag, *On Combining Laplacian and Optimization-based Mesh Smoothing Techniques*, AMD Trends in Unstructured Mesh Generation, vol. 220, pp. 37-43, (1997).

[3] T. Zhou, K. Shimada, *An angle-based approach to two-dimensional mesh smoothing*, Proceedings of Ninth International Meshing Roundtable, pp. 373–384, (2000).

[4] Freitag L. A., Plassmann P. E., *Local Optimization-based simplicial mesh untangling and improvement*, Int. J. Numer. Meth. Engng **49**, 109–125, (2000).

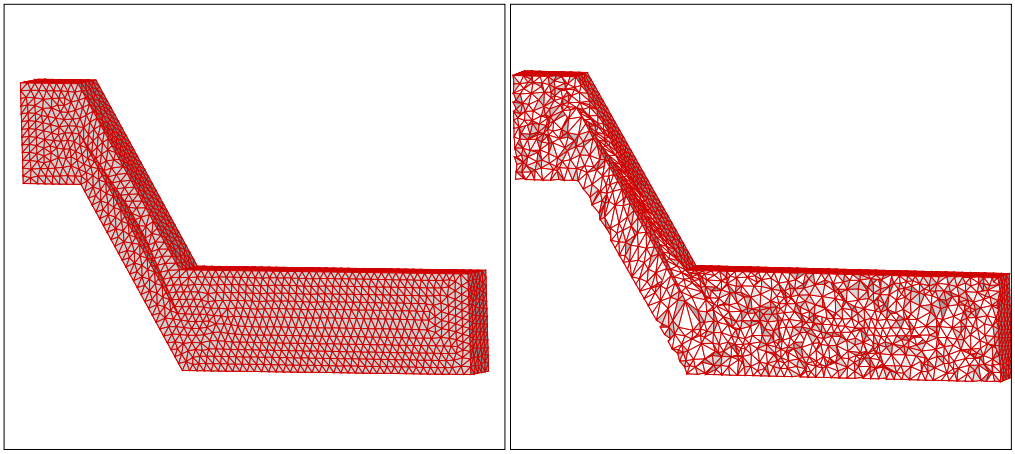

Figure 10: Left: The initial unstructured mesh for the wedge linear cascade. Right: A planar-cut view of the final mesh where the elements compressed between the wedge and the periodic boundary are of lesser quality than the rest of the mesh.

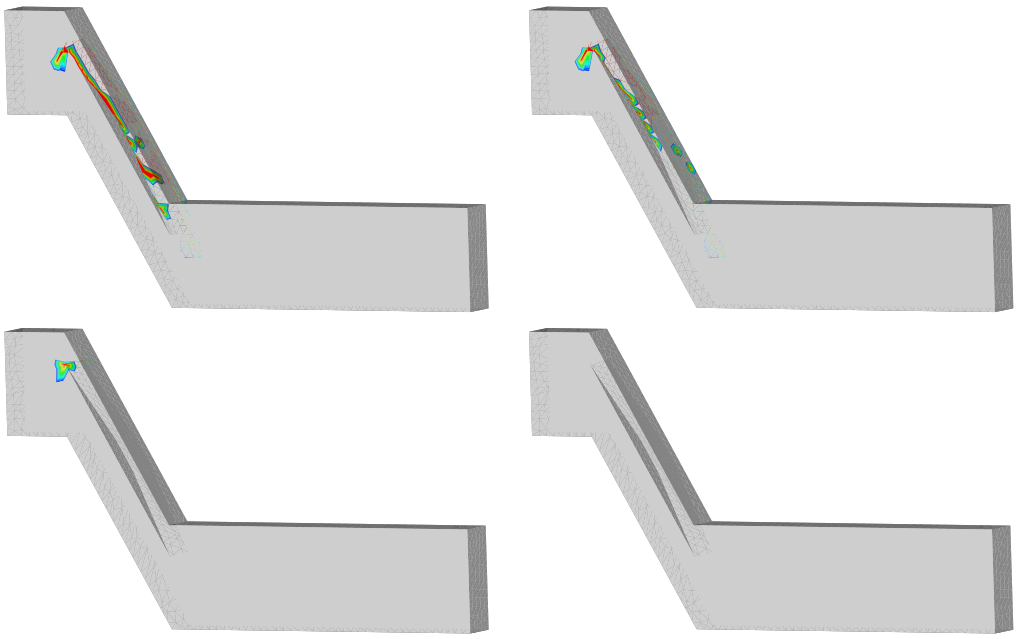

Figure 11: Eliminating inverted elements in the wedge cascade case.

[5] Burg, C. O. E., *A Robust Unstructured Grid Movement Strategy using Three-Dimensional Torsional Springs*, AIAA Paper 2004–2529, (2004).

[6] Liu, A. and Joe, B. - "Relationship Between Tetrahedron Shape Measures," BIT, vol. 34, pp. 268-287, 1994.

[7] Knupp P. *Algebraic Mesh Quality Metrics." SIAM Journal on Scientific Computing*, vol. 23, 193–218, (2001).

[8] J. Dompierre, P. Labbe, F. Guibault, and R. Camerero, *Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization*, Proceedings of the 7th International Meshing Roundtable, Dearborn, MI, (1998).