UNTANGLING AND OPTIMIZATION OF UNSTRUCTURED HEXAHEDRAL MESHES

K. Kovalev[(*)], M. Delanaye[(**)], Ch. Hirsch[(*)]

kvk@stro.vub.ac.be

[(*)] Vrije Universiteit Brussel, Pleinlaan, 2, 1050 Brussels

[(**)] NUMECA Int., Av. Franklin Roosevelt, 5, 1050 Brussels

Abstract

An effective method for untangling and optimization of hexahedral unstructured non-conformal meshes is presented. The method has been developed as a part of the Numeca new mesh generator (HEXPRESS^{TM}).

The untangling algorithm is based on the successive analysis and correction of concave cells, via local untangling of sets of cells referencing a mesh node. The problem of untangling a hexahedral cell can be decomposed into untangling of the tetrahedra that constitute the cell. This subdivision is not unique and must be appropriately chosen to ensure the efficiency of the algorithm.

The optimization algorithm is also based on the quality analysis and optimization of the sets of cells attached to a mesh node. Successive optimization of such sets containing valid cells results in an overall mesh quality improvement. The local optimization of hexahedral cells is performed via optimizing a variational functional for the set of tetrahedra representing the cell. It is important that identical patterns for decomposition of a cell into tetrahedra are used in both untangling and optimization.

The results of applying the methods to various test cases including many complicated geometries of interest for industry. Some of the advantages and disadvantages are discussed.

Keywords: Mesh generation, unstructured, hexahedral, optimization, untangling, mesh quality, Jacobian positiveness.

1 Introduction

Automatic generation of unstructured meshes sometimes produces meshes with badly shaped and inverted elements. The case is even worse for hexahedral meshing because of the high flexibility of an hexahedron to become extremely distorted and its intrinsic difficulties of generating such meshes for complex configurations. Presence of invalid elements in the mesh may lead to major breakdown of the simulation algorithm. The development of quality improvement a posteriori tools is of prime importance.

Meshes consisting solely of simplicial elements (triangles or tetrahedra) are most likely to have their quality improved. This is due to existence of a linear mapping of the uniform triangle or tetrahedron onto an arbitrary triangular or tetrahedral cell. It enables one to detect poorly-shaped cells by evaluating the Jacobian of the corresponding mapping. The linearity is of extreme importance because the Jacobian of a linear mapping is constant. Therefore, evaluating the Jacobian of the mapping unambiguously answers the question whether or not a mesh element is inverted, i.e. has non-positive volume.

However, the situation for computational meshes consisting of non-simplicial elements is different. Particularly, no linear mapping exists for hexahedral elements to transform them onto the canonical element. Instead, the mapping between a unit cube and a hexahedral cell is trilinear and has complicated behavior throughout the cell. Also, no convexity relations can be effectively exploited because faces of a hexahedron are generally not planar and can be folded even when all 8 corners are convex. This makes the task of the quality improvement for hexahedral meshes much more complicated, as compared to simplicial and even quadrilateral meshes, because simple convexity relations can be easily established for the latter.

An effective method for untangling and optimization of hexahedral unstructured non-conformal meshes is presented in the paper. The method has been developed as a part of the NUMECA new mesh generator HEXPRESS™. It is able to untangle invalid (concave) cells and optimize valid but poorly-shaped cells resulting from grid generation process. The goal is to obtain a mesh with all convex cells.

The HEXPRESS™ grid generation process is based on a top-down approach and includes several stages. First, an initial non-boundary-conforming mesh is created and refined based on geometry particularities. The cells that fall outside or intersecting the domain are removed from the volume mesh. Next, the surface of the resulting staircase mesh is projected on the domain boundary and layers of buffer cells are inserted between the volume mesh and the corresponding surface mesh in order to obtain a body-conforming mesh. Concave and poorly shaped cells may occur during the projection step and usually are concentrated near the boundary. In the final stage, new untangling and optimization tools are applied to transform these cells to convex ones and recover a mesh of high quality. Optionally, layers of high aspect ratio cells for viscous flow computations may be inserted.

The grid generator is coupled with a new flow solver, which aggressively adapts the mesh based on local solution error estimation in order to obtain a mesh optimized for the particular flow solution. Subdividing a concave cell may result in new cells with negative volumes. The latter are unacceptable for the reasons of robustness and accuracy of the flow solver. That is why the automatic optimization procedure is necessary.

The paper is organized as follows. Section 2 describes the details of the untangling method applied to the hexahedral meshes. The problem is formulated and the solution procedure is described. Section 3 is devoted to the optimization of the hexahedral unstructured meshes. A detailed formulation of the problem is given and the guidelines to solve the problem are described. Section 4 presents some results of the untangling and optimization methods, discusses advantages and disadvantages of these approaches.

2 Mesh Untangling

2.1 Problem statement

We formulate the problem of untangling of an unstructured hexahedral mesh as follows:

Given: *Unstructured hexahedral non-conformal mesh with fixed boundary. Let* $\{V_I\}$ *be the set of internal non-hanging nodes of the mesh,* $\{V_B\}$ *- the set of non-hanging boundary nodes. The nodes from* $\{V_B\}$ *are not allowed to move.*

To find: *The new set of positions for the nodes from* $\{V_I\}$*, such that none of the cells in the mesh are concave, while the topology of the mesh is preserved.*

A node is called *hanging* if it results from refining a mesh edge into two edges or a face into four faces. The geometric position of a hanging node is determined by the positions of the vertices of the corresponding edge or face (Figure 1).

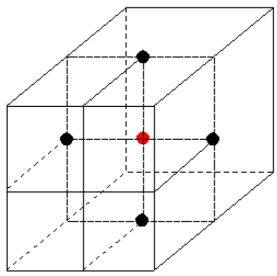

Figure 1: The four black nodes are the hanging nodes belonging to edges. The red node is a hanging node belonging to a face.

A cell is called *concave* if at least one of the tetrahedra constituting[1] the cell has negative volume (Figure 2). A cell is called *concave with respect to a node* if at least one of the tetrahedra constituting the cell and *referencing this node* has negative volume.

Due to multiple topological and geometrical limitations and dependencies the problem of global mesh untangling is too complicated to solve implicitly. Instead, it is decomposed into untangling of a number of sub-meshes with only one internal node. The problem is solved iteratively. The cells that surround each non-hanging node, at least one of which is concave with respect to the node, are successively untangled at each iteration.

[1] The pattern of subdivision of a cell into tetrahedra must be chosen a priori (see section 3.2).

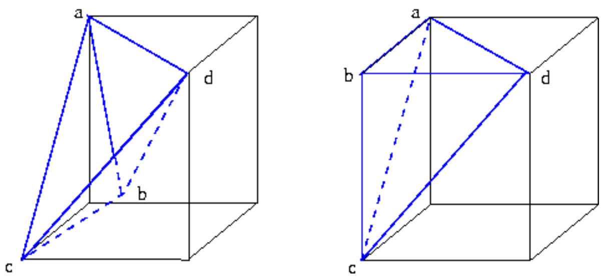

Figure 2: The cell on the left is concave because the volume of the tetrahedron ***abcd*** is negative (vertex ***b*** is behind the triangle ***acd***). It is concave with respect to the vertices ***a***, ***b***, ***c***, ***d***. Unlike the convex cell on the right, for which the same tetrahedron has positive volume.

Iterations are repeated until either no concave cells remain in the mesh or no concave cells can be untangled anymore. The problem of local untangling can be formulated as follows:

Given: *An arbitrary non-hanging node N belonging to* $\{V_I\}$ *and the ball of surrounding cells Ball(N).*

To find: *The new position for N, such that none of the cells in Ball(N) are concave.*

Ball(Node) is the set of all mesh cells that reference *Node*. *Node* cannot be hanging. *Kernel(Ball(Node))* is a sub-region of *Ball*, such that if and only if *Node* is positioned at any point belonging to *Kernel* then all *Ball*'s cells are non-concave with respect to *Node* (Figure 3). The notation (ball of mesh elements, kernel) is similar to that employed in 6.

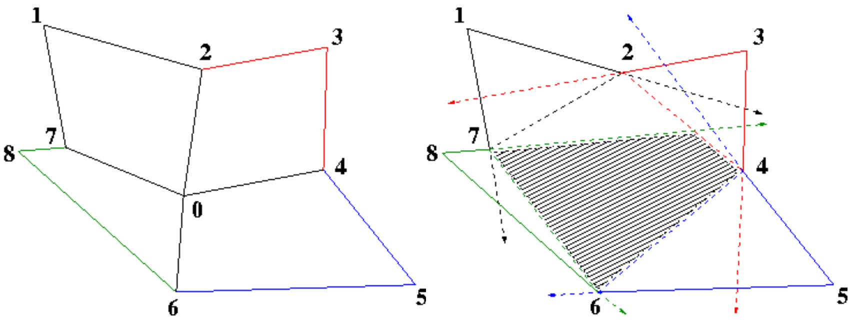

Figure 3: *Ball(Node 0)* contains 4 cells (left); *Kernel* (shaded region) results from intersection of the convexity regions of each cell bounded by the dashed lines of the cell's color (right).

The two-dimensional example demonstrates the idea of untangling for the case of quadrilateral 2-D meshes (Figure 3). The cells around the node *N* represent the corresponding ball - *Ball(N)* and the shaded region limited with dashed lines represents the kernel - *Kernel(Ball)* of the ball. Placing the node *N* at any location inside *Kernel(Ball)* guarantees convexity of *Ball(N)*, provided that the volume of *Kernel(Ball)* is positive. Negative volume of a kernel indicates that no valid convexity region exists and the untangling problem has no solution. Zero kernel volume indicates that the kernel is degenerate, i.e. it is collapsed to a point, a line segment, or a flat polygon (in 3-D only).

2.2 Algorithm

The simplex method 2 is an effective tool for solving local untangling problems. The method is generally used for solving linear optimization problems. If a linear function to be optimized on the set of *N* independent variables is given and a series of linear constraints for these variables is specified, the simplex method is capable to find the solution if it exists. The solution is always a point at the intersection of the linear constraints, because a linear function defined on a bounded domain can only reach its maximum at the boundary.

If the solution does not exist, it may be due to two possible reasons. First, the region bounded by the linear constraints may not be necessarily enclosed and, therefore, it is possible that the function reaches its maximum at the infinity. Second, the region bounded by the constraints may be empty.

The solution of the untangling problem must be a new position of the node, such that no concave cells remain in the ball. In order to properly set up the simplex problem, the series of linear constraints must be established and the optimized function must be defined. Each cell of the ball provides certain number of linear constraints (see Figure 3). In this 2-D illustration each quadrilateral cell can be subdivided into 2 triangles by each of the 2 diagonals. Each of these 4 triangles must have positive volume in order to guarantee convexity of the cell. The condition of volume positiveness of a triangle can be expressed by the coordinates $(x,y)_{0,1,2}$ of its vertices as follows:

$$\begin{vmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{vmatrix} \geq 0$$

This expression is linear with respect to any of the vertex coordinates. Therefore, as only 3 of 4 triangles constituting the cell reference the central node of the ball, 3 linear constraints result from each cell. In 3-D the number of constraints per cell depends on the pattern used to decompose the cell into tetrahedra: each tetrahedron provides one linear constraint. The choice of the decomposition pattern is addressed in section 3.2.

As we are only interested in the kernel itself, the choice of optimized function can be almost arbitrary. However, the gradient of the optimized function must be non-zero, otherwise no solution will be found. The new position must be located strictly within the kernel. If the point that results from the solution is located at the kernel boundary, it means that at least one cell referencing the node will be in an undetermined status between concave and convex. If this situation arises due to the kernel degeneracy, it is treated as "no solution". In all other cases, when the kernel is not degenerate, it is always possible to find a point located strictly within the kernel. This is achieved via combining several solutions that employ different optimized functions.

3 Mesh optimization

3.1 Problem statement

The problem of optimization of an unstructured hexahedral mesh can be formulated similarly to the untangling problem:

Given: *Unstructured hexahedral non-conformal mesh with fixed boundary. Let $\{V_I\}$ be the set of internal non-hanging nodes of the mesh, $\{V_B\}$ - the set of non-hanging boundary nodes. The nodes from $\{V_B\}$ are not allowed to move.*

To find: *The new set of positions for the nodes from $\{V_I\}$, such that the mesh quality, measured in a certain way, is higher than the quality of the initial mesh.*

It is rather complicated and time-consuming to apply optimization directly to the entire mesh. However, as the overall quality measure is contributed by every mesh cell, it is convenient to optimize the entire mesh by applying optimization to arbitrary limited sets of mesh cells with fixed boundary. The latter condition is necessary to prevent sub-mesh modifications from affecting the exterior mesh. In the scope of this work, optimization is applied to the balls of cells around each internal non-hanging mesh node. Therefore, the local optimization problem can be formulated as follows:

Given: *An arbitrary non-hanging node N belonging to $\{V_I\}$ and the ball of surrounding cells B(N).*

To find: *The new position for Node, such that the total quality of the cell in B(N) none of the cells in Ball(Node) are concave.*

The procedure is applied iteratively to the entire mesh until the increase in quality per iteration becomes lower than some threshold value.

3.2 Algorithm

The quality of a mesh cell is represented by the deformation energy density functional of the trilinear map between a unit cube and an arbitrary hexahedral cell 1:

$$I = \left[\frac{x_\xi^2 + y_\xi^2 + z_\xi^2}{a^2} + \frac{x_\eta^2 + y_\eta^2 + z_\eta^2}{b^2} + \frac{x_\varsigma^2 + y_\varsigma^2 + z_\varsigma^2}{c^2} \right] \cdot \frac{abc}{\sqrt{J}} \tag{1}$$

where $J = \begin{vmatrix} G_{00} & G_{01} & G_{02} \\ G_{10} & G_{11} & G_{12} \\ G_{20} & G_{21} & G_{22} \end{vmatrix}$, $G_{ij} = x_{\xi_i} x_{\xi_j} + y_{\xi_i} y_{\xi_j} + z_{\xi_i} z_{\xi_j}$, $\begin{cases} i, j = 0,...,2 \\ \xi_{0,1,2} = \xi, \eta, \varsigma \end{cases}$

The global functional value is contributed by each mesh cell. The functional of a cell, is the sum of the functional values computed separately for each tetrahedron constituting the cell. The decomposition of a hexahedral cell into the set of tetrahedra must be such that optimizing these tetrahedra would result in overall improvement of the cell quality. The decomposition suggested by Ivanenko in 1 (Figure 4) was chosen because it provides proper discrete analogue of the Jacobian of a hexahedral cell. As it was mentioned in section 2.2, the same decomposition is used in both untangling and optimization procedure.

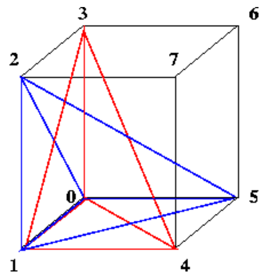

Figure 4: The hexahedron is decomposed into 24 tetrahedra, 2 tetrahedra correspond to each of 12 edges. For instance, red and blue tetrahedra are assigned to the edge 01.

An arbitrary ball of cells around a mesh node is optimized as follows. The direction of the highest gradient is identified as a search direction. The local minimum is now located on the line passing through the current node position and oriented along the search direction. Afterwards, a 1-D minimization problem is solved to find the location of the minimum and the node is repositioned. Figure 5 shows functional (1) plotted across an arbitrary straight line that passes through the kernel of a ball of cells. The functional becomes infinite at the positions where this line intersects the linear constraints. It means that optimization can never move a node behind any of the linear constraints and, therefore, a concave cell remains concave and a convex cell remains convex after optimization.

In practice, a combined untangling/optimization approach is implemented in the HEXPRESS™ mesh generator. First, concave cells are being untangled until no node can be displaced to produce new convex cells and after that, all convex cells are being optimized for several iterations. This combined procedure is repeated either for several times or until the mesh gains acceptable quality.

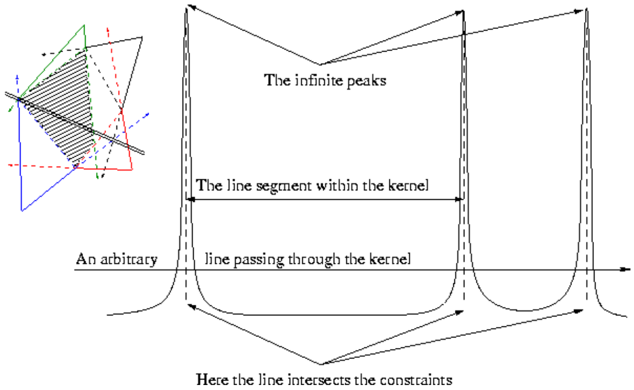

Figure 5: Functional plotted along a straight line intersecting the kernel of the ball (double black line on the left) has infinite barriers at intersections with linear constraints.

4 Results

The combined untangling/optimization approach was tested for various test cases in order to demonstrate its applicability to the geometries of industrial interest. The unstructured hexahedral mesh was generated in the exterior domain of the geometric configuration of a missile shown in Figure 6.

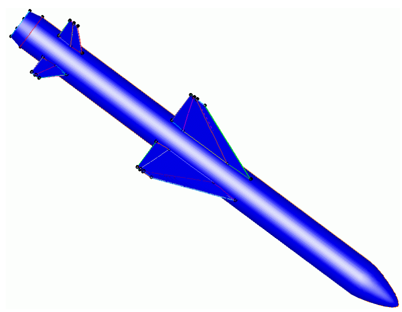

Figure 6: The geometry of a missile (courtesy from Aerospatiale).

Before the untangling/optimization step is performed, the mesh contains a significant percent of concave cells, some of which are highly distorted or have negative volume. Typically, these cells are concentrated near the surface of a model, especially in the vicinity of topological edges. Figure 7 shows the concave cells around one of the missile’s wings. The quality of concave cells ranges from rather acceptable up to totally distorted. The latter may even have negative volume. Figure 8 illustrates a close-up of a cluster of concave cells located near one of the rear wings of the missile. The distribution of quality of poorly-shaped cells shows that some of them are very distorted and a quality improving procedure is necessary.

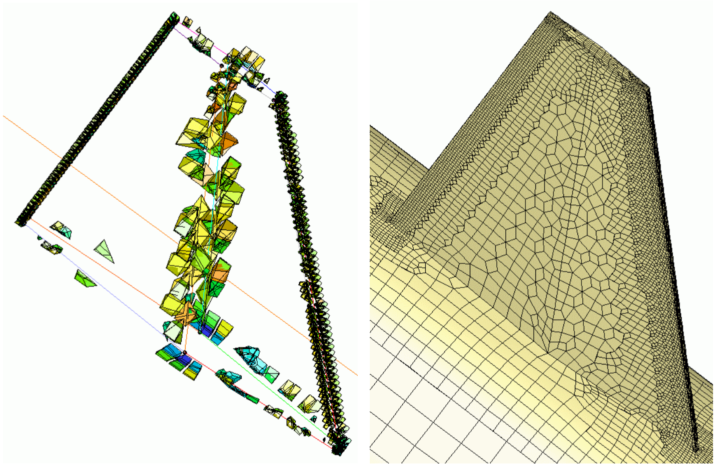

Figure 7: Concave cells are typically concentrated around topological edges of a wing of the missile (left). The corresponding surface mesh is shown on the right.

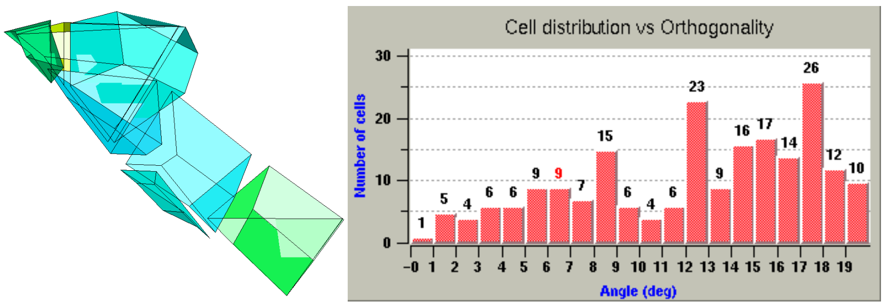

Figure 8: An example of a cluster of concave cells (left). On the right – the fragment of the cell quality distribution in the range between 0 and 20 degrees.

The described in the previous sections combined untangling/optimization approach is applied to the missile mesh as a final step of the generation process. The untangling sweeps and optimization iterations are combined in a complex sequence to make them as efficient as possible. This step has transformed 100 % of concave cells into convex ones.

Figure 9 compares the quality distributions for the mesh before and after optimization. It is worth noticing that only 3 cells (as compared to 205 initially) with dihedral angle lower than 20 degrees remain in the mesh. The optimization step consumes approximately the same CPU-time as the other steps.

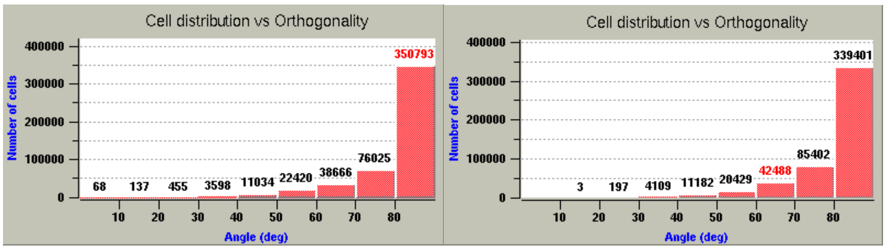

Figure 9: Quality diagrams of the missile mesh before (left) and after (right) quality improving step.

Test-case	**The total number of cells**	**Concave cells before optimization**		**Concave cells after optimization**	
1	54864	153	0.279 %	0	0.0 %
2	72764	86	0.118 %	0	0.0 %
3	75070	120	0.160 %	0	0.0 %
4	98056	299	0.305 %	1	0.0010 %
5	115662	246	0.213 %	2	0.0017 %
6	125477	161	0.128 %	0	0.0 %
7	198362	338	0.170 %	0	0.0 %
8	235370	497	0.211 %	0	0.0 %
9	265913	884	0.332 %	13	0.0049 %
10	301298	901	0.299 %	6	0.0020 %
11	440678	1344	0.305 %	7	0.0016 %
12	497672	3292	0.661 %	0	0.0 %
Missile	559169	5664	1.013 %	0	0.0 %
14	773268	2633	0.341 %	0	0.0 %
15	1130575	2205	0.195 %	24	0.0021 %

Table 1: The comparison of the populations of concave cells before and after optimization step for various test-cases.

This example demonstrates successful application of the untangling/optimization to the meshes generated around complex geometries. Table 1 gives a good picture of the average efficiency of the untangling method. It represents a set of several industrial complex geometries that were used to test and validate the untangling/optimization module. The cases are sorted by the total number of cells in a mesh. The table demonstrates that for relatively small test-cases, nearly or exactly 100 % of concave cells are being fixed. For larger meshes it is not so easy but the percentage of the concave cells remaining in a mesh after untangling/optimization step is very low and ranges between 0.001 and 0.005 %.

5 Conclusions

The paper presents a new approach for the quality improvement of hexahedral meshes based on low-cost simplicial untangling algorithm and optimization method. The results prove the efficiency of the combined approach at relatively low computational costs. Only the optimization part of the combined approach takes significant amounts of computational time, untangling, on the contrary, is very fast and cheap. It can not only untangle concave mesh cells but also improve their quality without optimizing them directly. This allows to save computational time.

We conclude that the combined approach is capable of improving the quality of meshes more efficiently than stand-alone optimization tools or optimization-based untangling algorithms 4. The main advantage of the combined approach is that similar mesh quality improvement can be achieved for shorter computational time as compared to pure optimization-based approaches. This is because the fast untangling method described here typically takes care of at least 80-90% of concave cells in a mesh before any optimization is applied. Therefore, the method usually untangles a majority of concave cells in a mesh very quickly, therefore providing valid meshes for minimum time.

References

1. Sergey A. Ivanenko, Harmonic mappings, Handbook of grid generation. Ed. By Joe F. Thompson, Bharat K. Soni, Nigel P. Weatherill, CRC Press 1998, pp. 8-1 – 8-43.
2. Sergey A. Ivanenko, Optimality principle for nondegenerate grids, Communications in applied mathematics series, Grid generation: new trends and applications in real world simulations, edited by S. A. Ivanenko, V. A. Garanzha, Russian Academy of Sciences, Computing Center, Moscow, 2001.
3. Numerical Recipes in FORTRAN. The Art of Scientific Computing. Second Edition, W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Camridge University Press, 1992, pp. 423-435.
4. Lori A. Freitag, Paul Plassman, Local optimization-based simplicial mesh untangling and improvement, International journal for numerical methods in engineering, John Wiley, vol. 49, num. 1, pp. 109-125.
5. Gregory Coussement, Euler-Lagrange formulation of a mesh optimization and adaptation method and unification with some other popular methods, 5th International Conference on Numerical Grid Generation in CFD and Related Fields, Mississippi, MS (USA), April 1-5, 1996.
6. P. Knupp, Hexahedral mesh untangling & algebraic mesh quality metrics, Proceedings, 9th International meshing roundtable, Sandia National Laboratories, 2000, pp. 173-183.