POLYHEDRAL MESH OPTIMIZATION USING THE INTERPOLATION TENSOR

Stefano Paoletti

Adapco, Melville, NY, U.S.A. stefano@adapco.com

ABSTRACT

A method for improving the quality of general conformal polygonal and polyhedral meshes is presented. The method is based on local optimization of a cell quality function that can be derived from the inertia tensor of the arrangement of nodes that belong to the cell. A definition of global mesh quality is also presented as a function of the quality of the cells in the mesh. The cell quality function is related to the ability of the arrangement of points to correctly interpolate test fields.

Keywords: mesh optimization, smoothing, interpolation tensor, condition number, polygons, polyhedra, meshless.

1. INTRODUCTION

In the last decade finite element and finite volume methods have become common tools in analyzing complex physical and engineering problems. In many cases automatic mesh generation techniques do not consistently produce meshes that can be used to solve such problems, because the *quality* of such meshes does not fulfill the minimal requirement of the solver.

Smoothing is one of the many techniques that can be applied to improve the quality of meshes, together with others such as topology modifications, local remeshing, adaptivity etc.

In recent years optimization-based smoothing has shown a large potential to address the improvement of meshes through node-movement, though it is important to remember that node-movement techniques alone cannot fully solve the mesh quality problem, since even after a complete optimization, particular meshes may have a quality that still does not fulfill the requirements of the solver. However, optimization-based smoothing is certainly a very useful tool in the hands of an analyst who has to resolve a specific problem and does not have the option of building a mesh with a different topology.

In the present paper we present an attempt to address the problem of improving a general conformal polyhedral (or polygonal in 2D) mesh by isotropic smoothing.

Optimization of two-dimensional polygonal and three-dimensional polyhedral meshes is a relatively unexplored field. Polyhedral meshes have started to be available only recently, with the introduction of reliable face-based finite volume methods and related polyhedral mesh generators in the CFD community [1]. However, polyhedral meshes can be trivially generated starting from a tetrahedral mesh and creating the dual mesh, i.e. the Voronoi or Dirichlet mesh.

Dual meshes created from tetrahedral meshes have interesting property, such as being made of all trivalent polyhedra. A polyhedron is trivalent when exactly three edges and three faces share each of its nodes. However, dual meshes of tetrahedral meshes require modifications to fit the original mesh boundary and some topological changes to improve the quality, such as collapsing small faces. After these operations the dual mesh is not longer composed by trivalent polyhedra only, and may present a very large number of non trivalent polyhedra (see Figure 11 for an example of such polyhedra).

A restricted number of polygonal and polyhedral shapes have been historically used as cells both by finite element and by finite volume solvers, namely triangles and quadrilaterals in 2D and tetrahedra, prisms, pyramids and hexahedra in 3D.

Several researchers [2],[3],[4],[5],[6],[7],[8] have concentrated their efforts to provide methods to improve the quality of meshes that include such shapes. The inspiring work of P.Knupp [2][3] based on the properties of the metric tensor derived from the Jacobian matrix, can be considered as a comprehensive description of objective functions whose optimization leads to the improvement of the quality of general polygonal 2D meshes and 3D meshes made of general trivalent polyhedra. The applicability of such ideas in three dimensions is limited, at least in theory, to purely trivalent polyhedra. This limitation is not 'painful' for the historical meshes, while for general polyhedral meshes it may result in the inability of improving the quality of some polyhedra.

Remembering that meshes are useful because they allow the solution of discretized field equations, it is desirable that the quality of a cell reflected its ability to correctly participate in the solution of such equations. In many practical cases the user of a simulation code does not have any knowledge of the solution of his set of equations and it is impossible to minimize the solution error by modifying a priori the positions of the cell nodes. However, it seems reasonable that the position of the cell nodes be such that 'test' fields (linear, quadratic etc) are approximated with the smallest possible error.

2. INTERPOLATION TENSOR

Let us consider a node $\mathbf{p_0}$ in the mesh and a set of n nodes $\mathbf{p_k}$ $k=1,..,n$ surrounding it as in Figure 1. For simplicity let us suppose that they are 3-dimensional points, even though the following considerations hold in any dimensions.

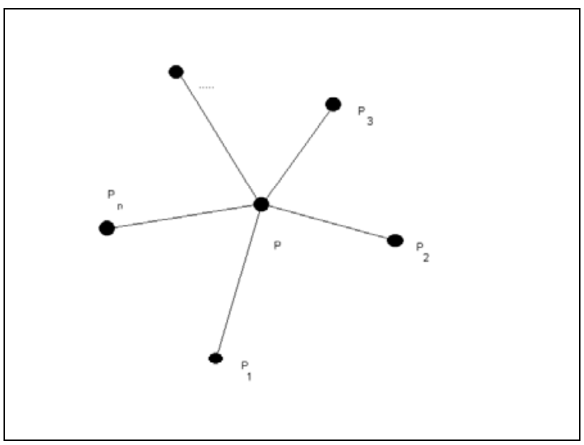

Figure 1. Generic arrangement of nodes

The nodes $\mathbf{p_k}$ may be its natural neighbors, the nodes connected by an edge, the other nodes of a cell or, more in general, the set of nodes within a specified volume or distance. The variation of a generic scalar field φ in the neighborhood of $\mathbf{p_0}$ can be expressed by the Taylor expansion (from now on Einstein's convention on repeated indices is used):

$$\Delta\varphi^{(k)} \approx \frac{\partial\varphi}{\partial x_i}\Delta x_i^{(k)} + \frac{1}{2}\frac{\partial^2\varphi}{\partial x_i \partial x_j}\Delta x_i^{(k)}\Delta x_j^{(k)} + \ldots \tag{1}$$

where k is the index of a point and:

$$\begin{aligned} & i = 1,2,3 \quad j = 1,2,3 \\ & x_1 = x, \ \ x_2 = y, \ \ x_3 = z \\ & \Delta\varphi^{(k)} = \varphi(\mathbf{p_k}) - \varphi(\mathbf{p_0}) \\ & \Delta x_i^{(k)} = x_i(\mathbf{p_k}) - x_i(\mathbf{p_0}) \end{aligned}$$

This methodology has also been called Generalized Finite Difference [9] and is the simplest definition used by meshless techniques. When the Taylor series is truncated at the second order terms, the least square method depicted above may be actually used to generate the stencils of first as well as second order discretized partial derivatives. Second order terms, thus, allow the solution of a large class of physical problems, including Navier-Stokes, heat-transfer equations etc.

For the development of the following methodology we limit the expansion to the first order terms. The reason is to reduce the amount of calculations needed to optimize a mesh, under the assumption that a linear approximation will suffice for the smoothing purposes. In reality there is no theoretical limitation and this same development could be carried out including an arbitrary number of terms of the Taylor expansion.

For the first order approximation, the error R of the Taylor expansion can be expressed as:

$$R_k = \Delta\varphi^{(k)} - \frac{\partial\varphi}{\partial x_i}\Delta x_i^{(k)} \tag{2}$$

The total error is: $\chi^2 = \sum_{k=1}^{n} R_k^2$ and applying the least square method to minimize it $\dfrac{\partial\chi^2}{\partial\left(\dfrac{\partial\varphi}{\partial x_i}\right)} = 0$ one obtains:

$$\mathbf{Ax} = \mathbf{A}\nabla\varphi = \mathbf{b} \tag{3}$$

For a three dimensional arrangement of points we have:

$$\mathbf{x} = \nabla\varphi = \begin{Bmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \\ \dfrac{\partial\varphi}{\partial z} \end{Bmatrix}; \quad \mathbf{b} = \begin{Bmatrix} \sum \Delta x \Delta\varphi \\ \sum \Delta y \Delta\varphi \\ \sum \Delta z \Delta\varphi \end{Bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} \sum \Delta x^2 & \sum \Delta x \Delta y & \sum \Delta x \Delta z \\ \sum \Delta x \Delta y & \sum \Delta y^2 & \sum \Delta y \Delta z \\ \sum \Delta x \Delta z & \sum \Delta y \Delta z & \sum \Delta z^2 \end{bmatrix} \quad (4)$$

The summations in (4) have been written without limits and indices on purpose, because they can be different when applied to a cell instead of a set of nodes, as it will be evident in section 3. The matrix **A** is a symmetric tensor of rank 2, as can be demonstrated applying the quotient rule, since for an arbitrary gradient vector **x**, **b** is always a vector. It is interesting to note that there is a relation between the tensor **A** and another important tensor that can be built using the arrangement of points. With simple manipulations of (4) we can state that:

$$\mathbf{A} = -\mathbf{T} + \sum_k d_k^2 \mathbf{I} \quad (5)$$

where **T** is the inertia tensor of the points $\mathbf{p_k}$, each one associated with a unit mass, as seen from point $\mathbf{p_0}$.

$$\mathbf{T} = \begin{bmatrix} \sum\left(\Delta y^2 + \Delta z^2\right) & -\sum \Delta x \Delta y & -\sum \Delta x \Delta z \\ -\sum \Delta x \Delta y & \sum\left(\Delta z^2 + \Delta x^2\right) & -\sum \Delta y \Delta z \\ -\sum \Delta x \Delta z & -\sum \Delta y \Delta z & \sum\left(\Delta x^2 + \Delta y^2\right) \end{bmatrix}$$

$$\sum_k d_k^2 = \sum_k (\Delta x_k^2 + \Delta y_k^2 + \Delta z_k^2)$$

The second term on the right hand side of (5) is a spherical tensor whose diagonal term is the sum of the distances squared of the nodes $\mathbf{p_k}$ from the node $\mathbf{p_0}$. **I** is the unit matrix.

The tensor **A** will be hereafter referred to as the **interpolation tensor.**

The solution of (3) is the least squares approximation of the gradient of the generic scalar field at $\mathbf{p_0}$. The value of the scalar field at an arbitrary point $\mathbf{p} = \mathbf{p}_0 + \mathbf{\Delta}$, $\mathbf{\Delta} = [\Delta_x, \Delta_y, \Delta_z]$, is readily found by:

$$\varphi(\mathbf{p}) = \varphi(\mathbf{p}_0) + \nabla\varphi \cdot \mathbf{\Delta} = \varphi(\mathbf{p}_0) + \mathbf{A}^{-1}\mathbf{b} \cdot \mathbf{\Delta}$$

In three dimensions the three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ of the interpolation tensor can be found by solving the characteristic equation:

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0$$

where I_1, I_2, I_3 are the three invariants of the tensor:

$$I_1 = a_{ii} = tr(\mathbf{A})$$

$$I_2 = \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} + \begin{vmatrix} a_{22} & a_{23} \\ a_{23} & a_{33} \end{vmatrix} + \begin{vmatrix} a_{33} & a_{13} \\ a_{13} & a_{11} \end{vmatrix}$$

$$I_3 = \varepsilon_{ijk} a_{1i} a_{2j} a_{3k} = Det(\mathbf{A}) \quad (6)$$

$tr(\mathbf{A})$ is the trace of the interpolation tensor, ε_{ijk} is the Levi-Civita symbol and $Det(\mathbf{A})$ is the determinant of the tensor. The invariants can also be expressed in terms of the eigenvalues. In fact they can be evaluated in the principal coordinate system of the tensor, where the tensor is diagonal with diagonal values exactly equal to the eigenvalues. So the equations (6) can be simplified into:

$$\begin{aligned} I_1 &= \lambda_1 + \lambda_2 + \lambda_3 \\ I_2 &= \lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1 \\ I_3 &= \lambda_1\lambda_2\lambda_3 \end{aligned} \quad (7)$$

A straightforward way to quantify the ability of the set of nodes $\mathbf{p_k}$ to correctly represent a linear field is to consider the condition number of the interpolation tensor **A**.

For a given set of points $\mathbf{p_k}$ surrounding $\mathbf{p_0}$ it seems reasonable to argue that the condition number of the interpolation tensor is a measure of the effect of perturbations either in the position of the nodes or in the scalar field at the nodes in the solution of the linear system in (3).

It ultimately gives a measure of the “goodness” of the linear interpolation on that particular arrangement of nodes.

3.CELL QUALITY

When applying the same concepts to a cell, it would be desirable to form the interpolation tensor in such a way that it would be independent of any particular interior point $\mathbf{p_0}$ and, instead, only dependent on the points of the cell.

This may be accomplished considering the so-called Moving Least Square method [9]. If the cell we are considering is formed by *n* nodes and *f* faces, there is a set *E* of *e=n+f-2* edges. The interpolation tensor can then be formed in the same way as in (4) with the difference that the summations are all evaluated over the set *E*.

Again, solving the 3x3 linear system of (3), the value of a scalar at any point **p** in the cell may be expressed as :

$$\varphi(\mathbf{p}) = \frac{1}{n}\vec{\nabla}\varphi \cdot \left(\sum_{k \subset E}^{n} (\mathbf{p} - \mathbf{p_k}) \right) = \\ = \frac{1}{n}\mathbf{A}^{-1}\mathbf{b} \cdot \left(\sum_{k \subset E}^{n} (\mathbf{p} - \mathbf{p_k}) \right)$$

(8)

The condition number of the interpolation tensor is here defined as $k = \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$, where $\|\mathbf{A}\| = \left[\sum_i \sum_j a_{ij}^2\right]^{1/2}$ is the Frobenius norm of the matrix A, as outlined in [2][3].

One could use a different matrix norm such as the spectral 2-norm. The latter has the disadvantage that the related condition number would be defined as the ratio between the maximum and the minimum eigenvalues of the matrix, whose formulation is not easily differentiable.

It is convenient to evaluate the condition number in the principal system of the interpolation tensor:

$$k = \left[\left(\sum_{i=1}^{d} \lambda_i^2 \right) \left(\sum_{i=1}^{d} \frac{1}{\lambda_i^2} \right) \right]^{1/2}$$

(9)

where d is the dimensionality of the cell, i.e. d=2 for surface or planar cells and d=3 for volume cell, while λ_i is the i-th eigenvalue of the tensor. Since the eigenvalues of a tensor are invariant under rotation, the condition number above defined is also invariant under rotations. It is intuitive to define the quality of a cell as:

$$q = \frac{d}{k}$$

(10)

The quality q is clearly invariant under rotation, translation, reflection and uniform scaling, so it meets the basic requirements for a cell shape measure that have been outlined for tetrahedral shape measures in [10]. This definition of cell quality has values between 0 and 1 where 1 stands for a perfect cell and 0 for a degenerated one.

The above definition holds in any dimension. Obviously, practical application of those ideas will be carried on in two and three dimensions. For examples, in three dimensions, degenerated means that the points of the cell are co-planar, co-linear or coincident. These conditions correspond respectively to one, two or three zero eigenvalues in the interpolation tensor, as can be seen by inspecting (9).

3.1 Three dimensional quality functions

In three dimensions, with some algebraic manipulations (see Appendix), the condition number can also be expressed in terms of the invariants of the interpolation tensor:

$$k = \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\| = \\ \left[\left(\lambda_1^2 + \lambda_2^2 + \lambda_3^2\right)\left(\frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} + \frac{1}{\lambda_3^2}\right) \right]^{1/2} = \\ \left[\left(\frac{I_1^2 - 2I_2}{I_3}\right)\left(\frac{I_2^2 - 2I_1 I_3}{I_3}\right) \right]^{1/2}$$

(11)

and the quality is given by:

$$q = \frac{3I_3}{\left[\left(I_1^2 - 2I_2\right)\left(I_2^2 - 2I_1 I_3\right)\right]^{1/2}}$$

(12)

It may also be interesting to apply this definition of cell quality to a pyramid, since the pyramid is the simplest non trivalent polyhedron. If we define a pyramid with a square base of side L and an apex of height $H = \frac{\sqrt{3}}{2}L$, as shown in Figure 2 then quality q=1, as it can be easily demonstrated applying (12).

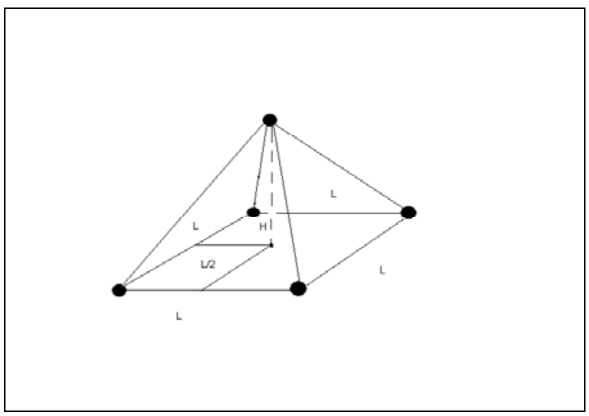

Figure 2. Perfect pyramid

3.2 Two dimensional quality functions

If I_3 vanishes, the points of the cell are either coplanar or collinear.

If one of the eigenvalues is zero (planar case) the two other eigenvalues can be found solving:

$$\lambda^2 - I_1\lambda + I_2 = 0 \tag{13}$$

This form of the characteristic equation is suitable to be used to define the condition number and the quality of two-dimensional (see Appendix for details of the algebraic manipulations) cells i.e. planar or surface meshes:

$$k = \left[\left(\lambda_1^2 + \lambda_2^2\right)\left(\frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2}\right)\right]^{1/2} = \left(\frac{I_1^2 - 2I_2}{I_2}\right) \tag{14}$$

$$q = \frac{2I_2}{I_1^2 - 2I_2} \tag{15}$$

Each regular polygon with an arbitrary number of equal length sides (i.e. an equilateral triangle, a square etc.) has quality q=1 as can be verified by applying (15).

3.3 Comparison between interpolation tensor and metric tensor

It is interesting to note that the interpolation tensor can be formed starting from the definition of a Jacobian matrix very similar to the one given in [2][3]. Remembering that:

$$\mathbf{e}_j = \lfloor \Delta x_j \quad \Delta y_j \quad \Delta z_j \rfloor$$

$$\mathbf{J} = \begin{bmatrix} \Delta x_1 & \Delta x_2 & \ldots & \Delta x_n \\ \Delta y_1 & \Delta y_2 & \ldots & \Delta y_n \\ \Delta z_1 & \Delta z_2 & \ldots & \Delta z_n \end{bmatrix}$$

then:

$$\mathbf{A} = \mathbf{J}\mathbf{J}^{\mathbf{T}}$$

The only difference is that **J** here can have an arbitrary number of columns, as many as the number n of point $\mathbf{p}_k$, i.e. in a regular conformal mesh the number of edges connected to the point $\mathbf{p}_0$. The metric tensor **G** described in [2][3], with the limitation of being formed with a square Jacobian, is simply:

$$\mathbf{G} = \mathbf{J}^{\mathbf{T}}\mathbf{J}$$

It is also worth noting that the eigenvalues of the metric tensor **G** and those of the interpolation tensor **A** are the same when **J** is a square matrix.

3.4 Geometric interpretation of the invariants

When the number of nodes $\mathbf{p}_k$ is equal to the dimensionality of the mesh, there is a simple geometrical interpretation of the invariants of the interpolation tensor. Let us focus our attention only on the nodes $\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_2$ shown in Figure 3. The invariants of **A** are:

$$I_1 = L_1^2 + L_2^2$$
$$I_2 = \left(\sum \Delta x^2\right)\left(\sum \Delta y^2\right) - \left(\sum \Delta x \Delta y\right)^2 = 4a^2$$

where L_1 and L_2 are the length of the edges going from $\mathbf{p}_0$ to $\mathbf{p}_1$ and to $\mathbf{p}_2$ and a is the area of triangle formed by $\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_2$.

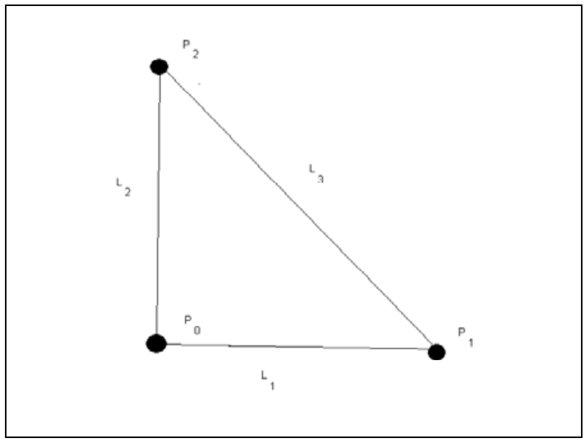

Figure 3. Bivalent node

Therefore the first invariant carries information related to the length of the edges, while the second is proportional to the square of the area of the triangle formed with points $\mathbf{p}_0$, $\mathbf{p}_1$ and $\mathbf{p}_2$. The condition number of **A**, in this case, would read as:

$$k = \frac{I_1^2}{I_2} - 2 = \frac{\left(L_1^2 + L_2^2\right)^2}{4a^2} - 2$$

that is the so-called Oddy's metric [11], which is widely used to improve triangular and quadrilateral meshes. If we apply the same formulation to the triangle of edges L_1, L_2, L_3 and area a the result is:

$$k = \frac{I_1^2}{I_2} - 2 = \frac{\left(L_1^2 + L_2^2 + L_3^2\right)^2}{12a^2} - 2$$

that substantially equals the square of the matrix-norm-based triangle quality measure k_2 in [12].

4.MESH QUALITY

It is not completely clear which is the best way to define the quality Q of a mesh, starting from the quality q_i (i=1,..,N) of the N component cells. One could define such quality in several ways such as the average $Q^{avg} = \frac{1}{N}\left(\sum_{i=1}^{N} q_i\right)$, the minimal $Q^{\min} = \min_{i=1,N} q_i$ etc. When a mesh is actually used to solve discretized partial differential equations, it may happen that even a single invalid (or extremely low quality) cell would prevent the solution from converging or, if the convergence is achieved, the solution would be locally affected by an intolerably high error.

However, an attempt to take into account these requirements can be made with the definition of a diagonal matrix whose non-zero terms are the qualities of the N cells that form the mesh:

$$\mathbf{K} = \begin{bmatrix} q_1 & 0 & \ldots & 0 \\ 0 & q_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & q_N \end{bmatrix} \tag{16}$$

Therefore, if we consider the condition number of K: $\tilde{k} = \|\mathbf{K}\|\|\mathbf{K}^{-1}\|$ we can define the quality of a mesh as :

$$Q^{mesh} = Q^{avg}\frac{N}{\tilde{k}} = \frac{\left(\sum_{i=1}^{N} q_i\right)}{\left[\left(\sum_{i=1}^{N} q_i^2\right)\left(\sum_{i=1}^{N}\frac{1}{q_i^2}\right)\right]^{1/2}} \tag{17}$$

This definition of global mesh quality has a range between 0 and 1, where 1 stands for a mesh made by perfect cells and 0 for a mesh where at least one cell is degenerated. Since it depends only on the cell qualities is invariant under rotation, translation, reflection and uniform scaling.

5.NUMERICAL METHOD

Without a-priori knowledge of which low quality cell could cause convergence problems, it is arguable that an optimization-based node-movement smoothing scheme should improve the quality of the worst cells. An improvement of the average quality is welcomed only if it does not cause a reduction of the quality of the worst cells.

Here the optimization will be carried out at node level on the sub-mesh surrounding it. The sub-mesh of a node is the small set of M cells that share that node. For simplicity, only the interior nodes of the mesh are moved, while the boundary nodes are kept frozen in their initial position. Even though this procedure is reasonable, it may prevent the smoother from improving those cells that have all or most of the nodes on boundary. Better result can certainly be achieved optimizing also the boundary nodes.

The first step of an optimization-based mesh improvement is to define a suitable objective function whose minimization would improve the mesh quality.

Many objective functions can be created using the invariants of the interpolation tensor, and a more comprehensive study is needed to assess their ability to improve the quality of the mesh.

Here the Frobenius norm of the interpolation tensor has been chosen because it is simple, differentiable and its minimization should make the eigenvalues equals [3], which in turn should make the cell quality equal to 1.
Thus the node objective function is simply (see Appendix for a derivation of the invariant form):

$$F = \sum_{i=1}^{M}\|\mathbf{A}_i\| \tag{18}$$

For the sub-mesh, the problem can be stated as an unconstrained minimization of a non-linear objective function that depends on the coordinates of the central point.

To efficiently find the minimum of F (or at least to ensure that a smaller value of F corresponds to the new position of the central node of the sub-mesh) the Hessian method is used.

It starts with imposing the necessary conditions for minimum $\nabla F = 0$ i.e. that the gradient of the objective function must vanish at the minimum. Consequently one can apply the Newton algorithm to solve the non-linear set of three equations and obtain:

$$\mathbf{H}(F)\boldsymbol{\delta} = -\nabla F \tag{19}$$

where $\mathbf{H}$ is the Hessian of F, δ the displacement to be imposed to the central point of the sub-mesh. The solution of
(19) represents the correct displacement to be imposed on the central point of the sub-mesh only if the initial position is sufficiently close to the minimum. Far from the minimum there is no guarantee that the Hessian is positive definite. Therefore, the proposed method performs a simple line minimization along the direction of δ and does not accept displacements that increase the sub-mesh objective function [8].

Several other methods, such as the conjugate gradient method etc., could be used to minimize the objective function but, in our knowledge [8], the Hessian method shows the best performance because it does not require many objective function evaluations, whose computation is generally the most expensive part of a minimization technique.

6.TEST CASES

The method outlined in the previous sections is suitable for improving the quality of planar, surface and volume meshes. One simple example of each class is presented in the following sections.

In all cases a total of 10 iterations of local optimization have been performed. For each of the test cases a table of results is shown that include the quality statistics before and after the optimization.

The computer used runs a Linux operating system and the processor is an Intel Pentium III Xeon at 700 Mhz. The surface and the planar optimization code evaluate the Hessian and the gradient of the objective function numerically, while the volume smoother uses the analytical formulation to reduce the cpu-time requirement. The following table describe the characteristic of each test case together with the cpu-time spent in smoothing.

Test Case	Number of cells	Number of nodes	Cpu-Time 10 iterations (sec.)
Planar	2,531	2,862	3.35
Surface	9,292	9,296	25.22
Polyhedral	33,143	178,224	120.33

Table 1. Test cases characteristics

6.1 Planar mesh

This test case is a multi-connected non-convex region with a clearly invalid initial 2D planar mesh as shown in Figure 4. Smoothing such a mesh can be a challenge, because the interpolation tensor method requires valid, convex, positive-area cells. In fact, directly optimizing the quality defined in (10) does not provide any direct control over the convexity of the cells. For this class of cases it is extremely useful to find an objective function that creates a "valid" mesh to be used as a starting mesh for the 'regular' smoother. After some experiment the best candidate was found in I_2. However, better convexity control may be achieved minimizing a different objective function such as:

$$F' = \sum_{i=1}^{M} \left\| \mathbf{A}'_i \right\|$$

where M is the number of cells in the sub-mesh and $\mathbf{A}'_i$ is the interpolation tensor of the central node and of the nodes the i-th cell in the sub-mesh that are connected by an edge to the central node. This objective function has a value equal to infinity when those nodes are coplanar creating a barrier. A more detailed study of this objective function will be the subject of additional work.

The smoothing procedure starts with 5 sweeps where the objective function is I_2 and then it switch to 5 more iterations with the interpolation tensor based objective function (18). The final result in Figure 5 is encouraging, since it shows a perfectly valid mesh with high quality.

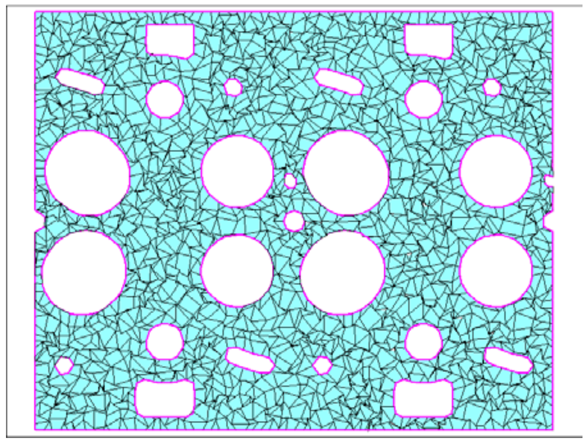

Figure 4. Invalid initial planar mesh

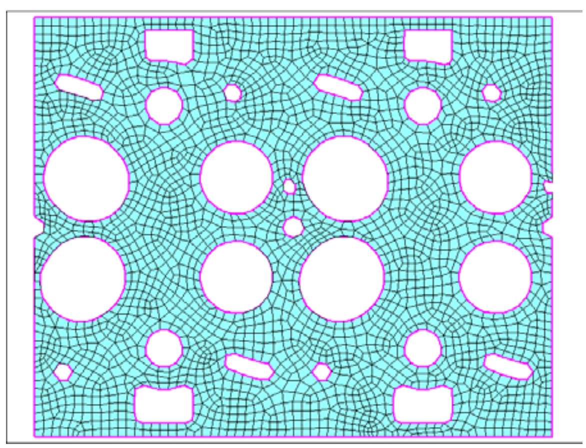

Figure 5. Smoothed planar mesh

Planar	Min Cell Quality	Avg Cell Quality	Max Cell Quality	Global Quality
Initial Mesh	9.80-9	5.30e-1	9.98e-01	5.02e-7
Smoothe d Mesh	2.97e-1	8.16e01	9.99e-01	7.98e-01

Table 2. Planar mesh results

6.2 Surface mesh

A mostly quad surface mesh is presented in Figure 6, with bad quality cells in the central part together with several triangles, pentagons and hexagons that have been created splitting and/or combining existing quads. The resulting surface, presented in Figure 7, shows that the worst cells have been improved as well as the non quadrilateral cells.

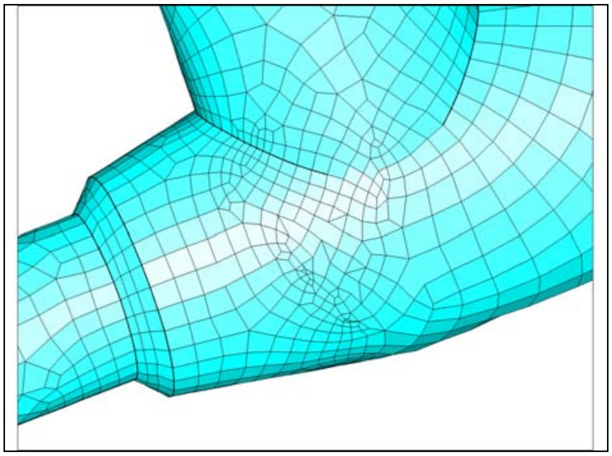

Figure 6. Initial surface mesh

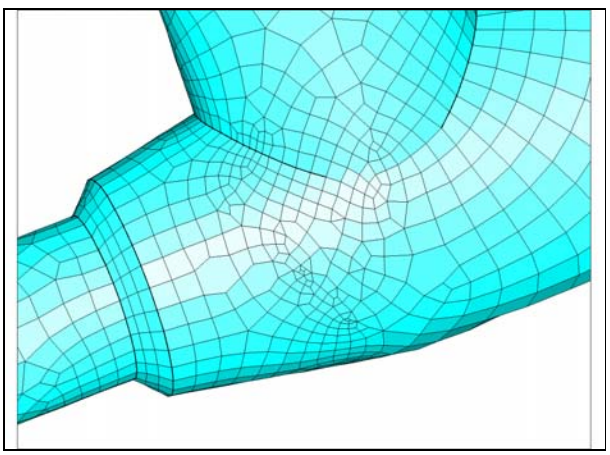

Figure 7. Smoothed surface mesh

.

Surface	Min Cell Quality	Avg Cell Quality	MaxCell Quality	Global Quality
Initial Mesh	2.10e-02	6.71e-01	9.99e-01	5.44e-01
Smoothed Mesh	3.37-01	9.19e-01	1.00e-00	9.01e-01

Table 3. Surface mesh results

The smoother modifies the position of the surface nodes and then projects them on the original surface mesh. Therefore no implicit constraint is imposed in the optimization.

6.3 Polyhedral mesh

A polyhedral mesh has been generated by taking the dual of a tetrahedral mesh and modifying it to fit the boundaries.
The small interior faces have been collapsed and the result is a polyhedral mesh with mostly trivalent polyhedra but many non trivalent polyhedra. In fact each time a quadrilateral face is collapsed a tetravalent node is created, each time a pentagonal face is collapsed a pentavalent node is created and so on. The concave polyhedra at the boundary have been split into two or more convex polyhedra. This operation too has the potential of creating non trivalent nodes. The boundary and the interior of the polyhedral mesh are shown in Fig. 8 and 9.

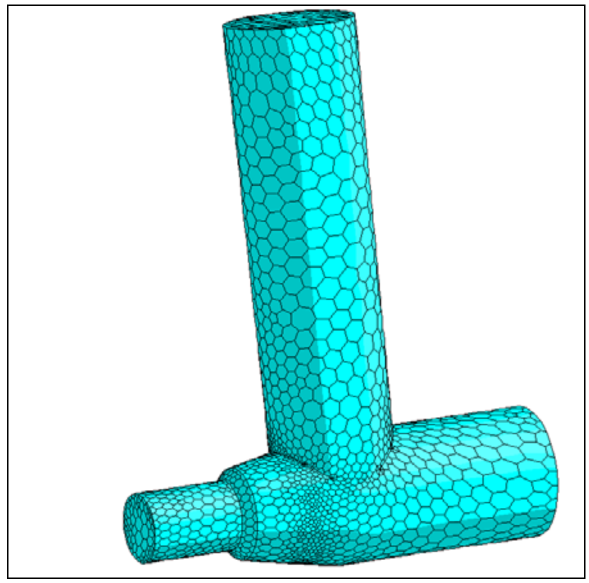

Figure 8. Boundary of polyhedral mesh.

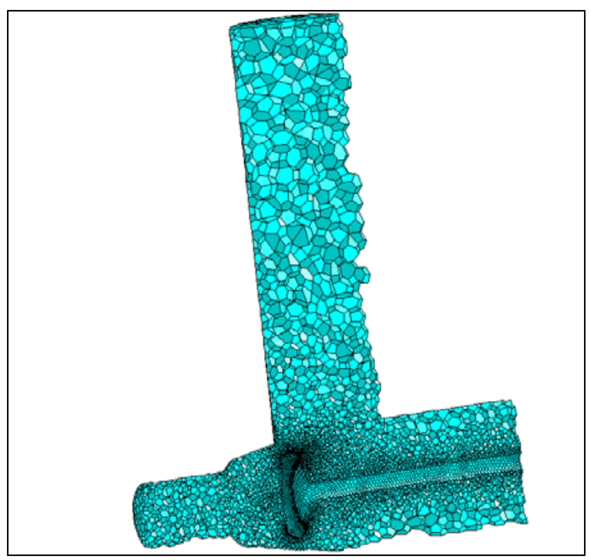

Figure 9. Interior of polyhedral mesh

A detail close to a non-convex region and one of the non trivalent polyhedra are shown in Fig. 10 and 11 respectively. The initial and final mesh statistics, together with their global quality, as defined in (12) are outlined in Table 4.

Polyhedral	Min Cell Quality	Avg Cell Quality	Max Cell Quality	Global Quality
Initial Mesh	1.83e-01	8.67E-1	9.99e-01	8.14e-01
Smoothed Mesh	2.18e-01	9.05e-01	9.99e-01	8.74e-01

Table 4. Polyhedral mesh results

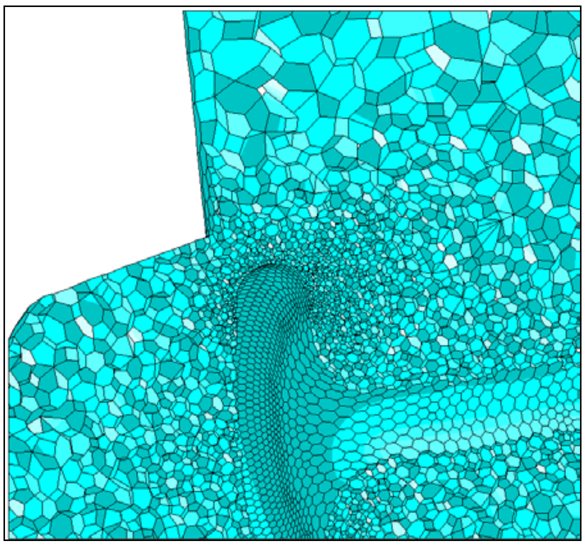

Figure 10. Detail of polyhedral mesh

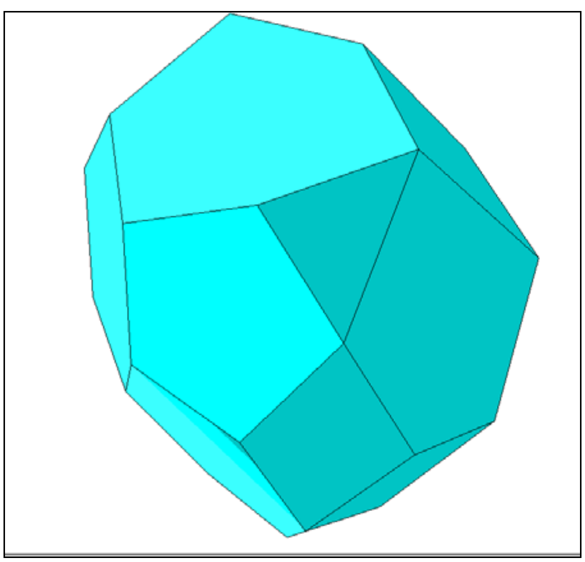

Figure 11. Non trivalent polyhedron

7.CONCLUSIONS

Definitions of both cell and mesh quality have been presented. A method for improving the quality of polygonal and polyhedral mesh based on the optimization of such quality has been described and tested in numerical experiments. This approach is similar to the one described in [2][3] but tries to address the limitation of trivalent polyhedra.
From the numerical experiments above described, it appears that this methodology can generate high quality mesh even starting from invalid initial meshes, at least in 2D, when an appropriate function (I_2) is used to achieve initial cell convexity throughout the mesh. Additional analysis I required to find the equivalent of I_2 in 3D. The natural candidate appears to be I_3.

More work is needed to describe the variety of objective functions that can be built combining the invariants of the interpolation tensor. It would also be useful, at least from a theoretical viewpoint, to carry the development of the interpolation tensor with higher order terms in the Taylor expansion. It may be worth noting that the smoothing technique here presented can also be used to optimize the position of the nodes for a meshless method.

APPENDIX

Three dimensional useful relations

$$\lambda_1^2+\lambda_2^2+\lambda_3^2=$$
$$=(\lambda_1+\lambda_2+\lambda_3)^2-2(\lambda_1\lambda_2+\lambda_2\lambda_3+\lambda_3\lambda_1)=$$
$$=I_1^2-2I_2$$

$$\frac{1}{\lambda_1^2}+\frac{1}{\lambda_2^2}+\frac{1}{\lambda_3^2}=$$
$$=\left(\frac{1}{\lambda_1}+\frac{1}{\lambda_2}+\frac{1}{\lambda_3}\right)^2-2\left(\frac{1}{\lambda_1\lambda_2}+\frac{1}{\lambda_2\lambda_3}+\frac{1}{\lambda_3\lambda_1}\right)=$$
$$=\left(\frac{I_2}{I_3}\right)^2-2\frac{I_1}{I_3}$$

$$k=\left[\left(\frac{I_1^2-2I_2}{I_3}\right)\left(\frac{I_2^2-2I_1I_3}{I_3}\right)\right]^{1/2}$$

Two dimensional useful relations

$$\lambda_1^2 + \lambda_2^2 = (\lambda_1 + \lambda_2)^2 - 2\lambda_1\lambda_2 = I_1^2 - 2I_2$$

$$\frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2} = \left(\frac{1}{\lambda_1} + \frac{1}{\lambda_2}\right)^2 - 2\frac{1}{\lambda_1\lambda_2} = \left(\frac{I_1}{I_2}\right)^2 - 2\frac{1}{I_2}$$

$$k = \left(\frac{I_1^2 - 2I_2}{I_2}\right) = \frac{I_1^2}{I_2} - 2$$

REFERENCES

[1] W.Oaks and S. Paoletti, "Polyhedral mesh generation", p57-66, Proceedings ot the 9th International Meshing RoundTable '2000, New Orleans, Louisiana, October 2-5 2000.

[2] P.Knupp, "Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities, Part I – A Framework for Surface Mesh Optimization & The Condition Number of the Jacobian Matrix", SAND 99-0709J.

[3] P.Knupp, "Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities, Part II – A Framework for Volume Mesh Optimization & The Condition Number of the Jacobian Matrix", SAND 99-0709J.

[4] P.Knupp, "Matrix Norm and the Condition Number: A General Framework to Improve Mesh Quality via Node-Movement", Proceedings of the 8th International Meshing RoundTable, South Lake Tahoe, 1999.

[5] L.Freitag, "On combining Laplacian and optimization-based smoothing techniques", in Trends in Unstructured Mesh Generation, volume AMD-vol.220, pp37-44, ASME Applied Mechanics Division, 1997.

[6] P.Zavattieri, "Optimization Strategies in Unstructured Mesh Generation", International Journal of Numerical Methods in Engineering, Vol.39, pp2055-2071,1996.

[7] J.Castillo, "A discrete variational method", SIAM Journal, 24(7), pp1069-1073,1986.

[8] S.Paoletti, Practical Optimization-Based Smoothing of Tetrahedral Meshes, Proceedings of the 7th International Conference on Numerical Grid Generation in Computational Field Simulation, Whistler, Brish Columbia, Canada, September 25-28, 2000.

[9] T.J.Liszka, C.A.M. Duarte, W.W. Tworzyidlo, "hp-Meshless Cloud Method", Comp. Meth. Appl. Mech. Eng., 139, pp. 263-288, 1996.

[10] J.Dompierre,P.Labbe',F.Guibault, R.Camarero, "Proposal of benchmark for 3D Unstructured Tetrahedral Mesh Optimization" Proc. Of 7th International Meshing RoundTable, Dearborn, Michigan, 26-28 October 1998.

[11] A. Oddy, J. Goldak, M. McDill, M. Bibby, "A distortion metric for isoparametric finite elements" Trans. CSME Np. 38-CSME-32, Accession no. 2161, 1988.

[12] P.P.Pebay, T.J.Baker,"A Comparison of Triangle quality Measures" . Proc. of the 10th International Meshing Roundtable, Newport Beach, California, U.S., October 7-10, 2001.