Smoothing and local refinement techniques for improving tetrahedral mesh quality

J.M. Escobar, R. Montenegro *, G. Montero, E. Rodríguez,
J.M. González-Yuste

University Institute of Intelligent Systems and Numerical Applications in Engineering, University of Las Palmas de Gran Canaria, Campus Universitario de Tafira, 35017 Las Palmas de Gran Canaria, Spain.

Abstract

The improvement in the mesh quality without changing its connectivity is bounded. This bound is associated with the topology of the mesh and with the constraints imposed by the boundary of the domain. To solve this problem, we propose in this work to combine the tetrahedral mesh optimization technique introduced in [1,2] with the local mesh refinement algorithm presented in [3]. The main idea consists in increasing the node number, and thus, the degrees of freedom, in the neighbourhood of the regions where the elements have poor quality. Then, we refine all the elements whose quality are below to a certain threshold. Once it is done, we initiate another stage of optimization until the quality of the mesh reaches a limit.

Key words: Mesh smoothing, mesh untangling, mesh generation, adaptive refinement, nested meshes, 3-D finite element method.

1 Introduction

There are two basic ways to improve the quality of a pre-existing mesh. The first, usually named mesh optimization, consists in moving each node to a new position that improves the quality of the surrounding elements. This technique preserves the topology of the mesh, that is, it does not modify the connectivity of the nodes. The second one involves some changes in the node connections. For example, edge swapping is a well known technique included in this category. In this work we propose an hybrid method that combines both approaches.

* Corresponding author.

Preprint submitted to Elsevier Science *20 October 2003*

Firstly, we focus in tetrahedral mesh optimization. The quality improvement in mesh optimization may be obtained by an iterative process in which each node of the mesh is moved to a new position that minimises a certain objective function. The objective function is derived from some quality measure [4] of the *local submesh*, that is, the set of tetrahedra connected to the adjustable or *free node*. Among the many objective functions described in the literature [5] those having *barriers* are specially indicated to improve the quality of a *valid* mesh in which there are non *inverted* elements. In this case, the barrier avoids the possible appearance of inverted elements in the optimization process. Nevertheless, the existence of barriers prevents these objective functions from working properly when the mesh is tangled. For example, if the free node is out of the *feasible region* (the positions where the free node must be located to get a *valid* submesh) the barrier avoids reaching the appropriate minimun. It can even happen that the feasible region does not exist, for example, when the fixed boundary of the local submesh is tangled. In all these situations these objective functions are not well defined on all $\mathbb{R}^3$ and, therefore, they are not suitable to improve the quality of the mesh. To overcome this problem we can proceed as Freitag et al in [6,7], where an optimization method consisting of two stages is proposed. In the first one, the inverted elements are untangled by an algorithm that maximises their negative Jacobian determinants; in the second, the resulting mesh from the first stage is smoothed using a objective function with barrier, based on the element condition number.

In this paper we propose an alternative to these procedures, such that the untangling and smoothing are carried out in the same stage. In order to do this, we substitute the objective functions by modified versions that are defined and regular on all $\mathbb{R}^3$. With these modifications, the optimization process is also directly applicable to meshes with inverted elements, making a previous untangling procedure unnecessary [1,2]. This simultaneous procedure allows the number of iterations for reaching a prescribed quality to be reduced. Nevertheless, the improvement in the mesh quality without changing its connectivity is bounded. This bound is associated with the topology of the mesh and with the constraints imposed by the boundary of the domain. In practice, we observe that both average and minimum quality tends to become steady to its respective bounds as the number of iteration increases. As result, once a sufficient number of iterations has been done, the mesh quality will not improve significatively and the process must then automatically stop.

In this work we propose to combine the above optimization techniques with the mesh refinement algorithm based on 8-subtetrahedron subdivision [8–10] and presented in [3]. The main idea consists in increasing the node number, and thus, the degrees of freedom, in the neighbourhood of the regions where the elements have poor quality. Then, we refine all the elements whose quality are below to a certain threshold. Once it is done, we initiate another stage of optimization until the quality of the mesh reaches a limit. The overall process can be repeated several times until the required quality is obtained or no additional improvement is got.

A promising field of study would combine the 3-D refinement/derefinement of nested meshes with node movement, where the ideas presented here could be introduced. Good recent results have been obtained in [11] and [12] using these techniques, for determining the shape and size of the elements in anisotropic problems.

We summarize the optimization techniques in Section 2 and the refinement algorithm in Section 3. To illustrate the effectiveness of our approach, we present in Section 4 several applications where it can be seen the validity of the proposed strategies. Finally, conclusions are presented in Section 5.

2 Mesh Optimization with Improved Objective Functions

In finite element simulation the mesh quality is a crucial aspect for good numerical behaviour of the method. In a first stage, some automatic 3-D mesh generator constructs meshes with poor quality and, in special cases, for example when node movement is required, inverted elements may appear. So, it is necessary to develop a procedure that optimizes the pre-existing mesh. This process must be able to smooth and untangle the mesh.

The most usual techniques to improve the quality of a *valid* mesh, that is, one that does not have inverted elements, are based upon local smoothing. In short, these techniques consist of finding the new positions that the mesh nodes must hold, in such a way that they optimize an objective function. Such a function is based on a certain measurement of the quality of the *local submesh*, $N(v)$, formed by the set of tetrahedra connected to the *free node* v. As it is a local optimization process, we can not guarantee that the final mesh is globally optimum. Nevertheless, after repeating this process several times for all the nodes of the current mesh, quite satisfactory results can be achieved. Usually, objective functions are appropriate to improve the quality of a valid mesh, but they do not work properly when there are inverted elements. This is because they present singularities (barriers) when any tetrahedron of $N(v)$ changes the sign of its Jacobian determinant. To avoid this problem we can proceed as Freitag et al in [6,7], where an optimization method consisting of two stages is proposed. In the first one, the possible inverted elements are untangled by an algorithm that maximises their negative Jacobian determinants [7]; in the second, the resulting mesh from the first stage is smoothed using another objective function based on a quality metric of the tetrahedra of $N(v)$ [6]. One of these objective functions are presented in Section 2.1. After the untangling procedure, the mesh has a very poor quality because the technique has no motivation to create good-quality elements. As remarked in [6], it is not possible to apply a gradient-based algorithm to optimize the objective function because it is not continuous all over $\mathbb{R}^3$, making it necessary to use other non-standard approaches.

In Section 2.2 we propose an alternative to this procedure, such that the untangling and smoothing are carried out in the same stage. For this purpose, we use a suitable modification of the objective function such that it is regular all over $\mathbb{R}^3$. When a feasible region (subset of $\mathbb{R}^3$ where v could be placed, being $N(v)$ a valid submesh) exists, the minima of the original and modified objective functions are very close and, when this region does not exist, the minimum of the modified objective function is located in such a way that it tends to untangle $N(v)$. The latter occurs, for example, when the fixed boundary of $N(v)$ is tangled. With this approach, we can use any standard and efficient unconstrained optimization method to find the minimum of the modified objective function, see for example [13].

In this work we have applied the proposed modification to one objective function derived from an *algebraic mesh quality metric* studied in [4], but it would also be possible to apply it to other objective functions which have barriers like those presented in [5].

2.1 Objective Functions

Several *tetrahedron shape measures* [14] could be used to construct an objective function. Nevertheless those obtained by algebraic operations are specially indicated for our purpose because they can be computed very efficiently. The above mentioned algebraic mesh quality metric and the corresponding objective function are shown in this Section.

Let T be a tetrahedral element in the physical space whose vertices are given by $\mathbf{x}_k = (x_k, y_k, z_k)^T \in \mathbb{R}^3$, $k = 0,1,2,3$ and T_R be the reference tetrahedron with vertices $\mathbf{u}_0 = (0,0,0)^T$, $\mathbf{u}_1 = (1,0,0)^T$, $\mathbf{u}_2 = (0,1,0)^T$ and $\mathbf{u}_3 = (0,0,1)^T$. If we choose $\mathbf{x}_0$ as the translation vector, the affine map that takes T_R to T is $\mathbf{x} = A\mathbf{u} + \mathbf{x}_0$, where A is the Jacobian matrix of the affine map referenced to node $\mathbf{x}_0$, and expressed as $A = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0, \mathbf{x}_3 - \mathbf{x}_0)$.

Let now T_I be an equilateral tetrahedron with all its edges of length one and vertices located at $\mathbf{v}_0 = (0,0,0)^T$, $\mathbf{v}_1 = (1,0,0)^T$, $\mathbf{v}_2 = (1/2, \sqrt{3}/2, 0)^T$, $\mathbf{v}_3 = \left(1/2, \sqrt{3}/6, \sqrt{2}/\sqrt{3}\right)^T$. Let $\mathbf{v} = W\mathbf{u}$ be the linear map that takes T_R to T_I, being $W = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ its Jacobian matrix.

Therefore, the affine map that takes T_I to T is given by $\mathbf{x} = AW^{-1}\mathbf{v} + \mathbf{x}_0$, and its Jacobian matrix is $S = AW^{-1}$. This weighted matrix S is independent of the node chosen as reference; it is said to be *node invariant* [4]. We can use matrix norms, determinant or trace of S to construct algebraic quality measures of T. For example, the Frobenius norm of S, defined by $|S| = \sqrt{\operatorname{tr}(S^T S)}$, is specially indicated because it is easily computable. Thus, it is shown in [4] that $q = \frac{3\sigma^{\frac{2}{3}}}{|S|^2}$ is an algebraic quality measure of T, where $\sigma = \det(S)$. The maximum value of these quality measures is the unity and it corresponds to equilateral tetrahedron. Besides, any flat tetrahedron has quality measure zero. We can derive an optimization function from this quality measure. Thus, let $\mathbf{x} = (x, y, z)^T$ be the free node position of v, and let S_m be the weighted Jacobian matrix of the m-th tetrahedron of $N(v)$. We define the objective function of $\mathbf{x}$, associated to an m-th tetrahedron as

$$\eta_m = \frac{|S_m|^2}{3\sigma_m^{\frac{2}{3}}} \tag{1}$$

Then, the corresponding objective function for $N(v)$ can be constructed by using the p-norm of $(\eta_1, \eta_2, \ldots, \eta_M)$ as

$$|K_\eta|_p(\mathbf{x}) = \left[\sum_{m=1}^{M} \eta_m^p(\mathbf{x})\right]^{\frac{1}{p}} \tag{2}$$

where M is the number of tetrahedra in $N(v)$. The objective function $|K_\eta|_1$ was deduced and used in [15] for smoothing and adapting of 2-D meshes. The same function was introduced in [16], for both 2 and 3-D mesh smoothing, as a result of a force-directed method. Finally, this function, among others, is studied and compared in [5]. We note that the cited authors only use this objective function for smoothing valid meshes.

Although this optimization function is smooth in those points where $N(v)$ is a valid submesh, it becomes discontinuous when the volume of any tetrahedron of $N(v)$ goes to zero. It is due to the fact that η_m approaches infinity when σ_m tends to zero and its numerator is bounded below. In fact, it is possible to prove that $|S_m|$ reaches its minimum, with strictly positive value, when v is placed in the geometric centre of the fixed face of the m-th tetrahedron. The positions where v must be located to get $N(v)$ to be valid, i.e., the feasible region, is the interior of the polyhedral set P defined as $P = \bigcap\limits_{m=1}^{M} H_m$, where H_m are the half-spaces defined by $\sigma_m(\mathbf{x}) \geqslant 0$. This set can occasionally be empty, for example, when the fixed boundary of $N(v)$ is tangled. In this situation, function $|K_\eta|_p$ stops being useful as optimization function. On the other hand, when the feasible region exists, that is $int\ P \neq \emptyset$, the objective function tends to infinity as v approaches the boundary of P. Due to these singularities, a barrier is formed which avoids reaching the appropriate minimum by using gradient-based algorithms, when these start from a free node outside the feasible region. In other words, with these algorithms we can not optimize a tangled mesh $N(v)$ with the above objective function.

2.2 Modified Objective Functions

We propose a modification in the previous objective function (2), so that the barrier associated with its singularities will be eliminated and the new function will be smooth all over $\mathbb{R}^3$. An essential requirement is that the minima of the original and modified functions are nearly identical when $int\ P \neq \emptyset$. Our modification consists of substituting σ in (2) by the positive and increasing function

$$h(\sigma) = \frac{1}{2}(\sigma + \sqrt{\sigma^2 + 4\delta^2}) \tag{3}$$

being the parameter $\delta = h(0)$. We represent in Figure 1 the function $h(\sigma)$. Thus, the new objective function here proposed is given by

$$\left|K_\eta^*\right|_p(\mathbf{x}) = \left[\sum_{m=1}^{M} (\eta_m^*)^p(\mathbf{x})\right]^{\frac{1}{p}} \tag{4}$$

where

$$\eta_m^* = \frac{|S_m|^2}{3h^{\frac{2}{3}}(\sigma_m)} \tag{5}$$

is the modified objective function for the m-th tetrahedron.

The behaviour of $h(\sigma)$ in function of δ parameter is such that, $\lim_{\delta \to 0} h(\sigma) = \sigma$, $\forall \sigma \geq 0$ and $\lim_{\delta \to 0} h(\sigma) = 0$, $\forall \sigma \leq 0$. Thus, if $int\ P \neq \emptyset$, then $\forall \mathbf{x} \in int\ P$ we have $\sigma_m(\mathbf{x}) > 0$, for $m = 1, 2, \ldots, M$ and, as smaller values of δ are chosen, $h(\sigma_m)$ behaves very much as σ_m, so that, the original objective function and its corresponding modified version are very close in the feasible region. Particularly, in the feasible region, as $\delta \to 0$, function $\left|K_\eta^*\right|_p$ converges pointwise to $|K_\eta|_p$. Besides, by considering that $\forall \sigma > 0$, $\lim_{\delta \to 0} h'(\sigma) = 1$ and $\lim_{\delta \to 0} h^{(n)}(\sigma) = 0$, for $n \geq 2$, it is easy to prove that the derivatives of this objective function verify the same property of convergence. As a result of these considerations, it may be concluded that the positions of v that minimise original and modified objective functions are nearly identical when δ is *small*. Actually, the value of δ is selected in terms of point v under consideration, making it as small as possible and in such a way that the evaluation of the minimum of modified functions does not present any computational problem. Suppose that $int\ P = \emptyset$, then the original objective function,

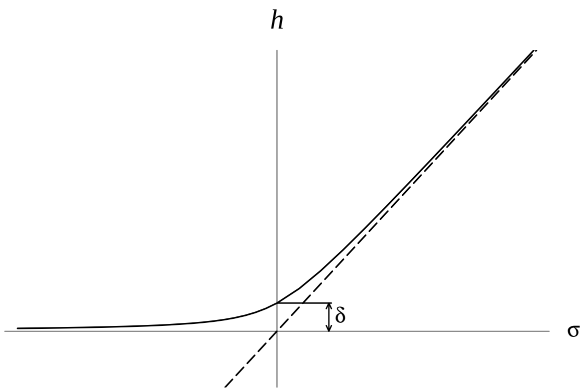


Fig. 1. Representation of function $h(\sigma)$.

$\left|K_\eta\right|_p$, is not suitable for our purpose because it is not correctly defined. Nevertheless, modified function is well defined and tends to solve the tangle. We can reason it from a qualitative point of view by considering that the dominant terms in $\left|K^*_\eta\right|_p$ are those associated to the tetrahedra with more negative values of σ and, therefore, the minimisation of these terms imply the increase of these values. It must be remarked that $h(\sigma)$ is an increasing function and $\left|K^*_\eta\right|_p$ tends to ∞ when the volume of any tetrahedron of $N(v)$ tends to $-\infty$, since $\lim\limits_{\sigma\to-\infty} h(\sigma) = 0$.

In conclusion, by using the modified objective function, we can untangle the mesh and, at the same time, improve its quality. More details about this mesh optimization procedure can be seen in reference [2].

3 Local Refinement Algorithm

We propose a local refinement algorithm [3] based on the 8-subtetrahedron subdivision developed in [10]. Consider an initial triangulation τ_1 of the domain given by a set of n_1 tetrahedra $t^1_1, t^1_2, ..., t^1_{n_1}$. Our goal is to build a sequence of m levels of nested meshes $T = \{\tau_1 < \tau_2 < ... < \tau_m\}$, such that the level τ_{j+1} is obtained from a local refinement of the previous level τ_j. The error indicator ϵ^j_i will be associated to the element $t^j_i \in \tau_j$. Once the error indicator ϵ^j_i is computed, such element must be refined if $\epsilon^j_i \geq \theta\epsilon^j_{\max}$, being $\theta \in [0, 1]$ the refinement parameter and $\epsilon^j_{\max}$ the maximal value of the error indicators of the elements of τ_j. From a constructive point of view, initially we shall obtain τ_2 from the initial mesh τ_1, attending to the following considerations:

a) *8-subtetrahedron subdivision*. A tetrahedron $t^1_i \in \tau_1$ is called of *type I* if $\epsilon^1_i \geq \gamma\epsilon^1_{\max}$. Later, this set of tetrahedra will be subdivided into 8 subtetrahedra as Figure 2(a) shows; 6 new nodes are introduced in the middle point of its edges and each one of its faces are subdivided into four subtriangles following the division proposed by Bank [17]. Thus, four subtetrahedra are determinated from the four vertices of t^1_i and the new edges. The other four subtetrahedra are obtained by joining the two nearest opposite vertices of the octohedron which result inside t^1_i. This simple strategy is that proposed in [10] or in [8], in opposite to others based on afin transformations to a reference tetrahedron, as that analysed in [9] which ensures the quality of the resulting tetrahedra. However, similar results were obtained by Bornemann et al. [8] with both strategies in their numerical experiments. On the other hand, for Lohner and Baum [10], this choice produces the lowest number of distorted tetrahedra in the refined mesh. Evidently, the best of the three existing options for the subdivision of the inner octohedron may be determined by analysing the quality of its four subtetrahedra, but this would augment the computational cost of the algorithm.

Once the *type I* tetrahedral subdivision is defined, we can find neighbouring tetrahedra which may have 6, 5, ..., 1 or 0 new nodes introduced in their edges that must be taken into account in order to ensure the mesh conformity. In the following we analyse each one of these cases. We must remark that in this process we only mark the edges of the tetrahedra of τ_1 in which a new node has been introduced. The corresponding tetrahedron is classified depending on the number of marked edges. In other words, until the conformity of τ_2 is not ensured by marking edges, this new mesh will not be defined.

b) *Tetrahedra with 6 new nodes*. Those tetrahedra that have marked their 6 edges for conformity reason, are included in the set of *type I* tetrahedra.

c) *Tetrahedra with 5 new nodes*. Those tetrahedra with 5 marked edges are also included in the set of *type I* tetrahedra. Previously, the edge without new node must be marked.

d) *Tetrahedra with 4 new nodes*. In this case, we mark the two free edges and it is considered as *type I*.

Proceeding as in (b), (c) and (d), we improve the mesh quality and simplify the algorithm considerably due to the global refinement defined in (a) by the error indicator. One may think that this procedure can augment the refined region, but we must take into account that only 1 or 2 new nodes are introduced from a total of 6. Note that this proportion is less or equal to that arising in the 2-D refinement with the 4-T Rivara algorithm, see for example References [18,19], in which the probability of finding a new node introduced in the longest edge of a triangle is $1/3$. This fact is accentuated in the proposed algorithm as its generalization in 3-D.

e) *Tetrahedra with 3 new nodes*. In this case, we must distinguish two situations:

e.1) If the 3 marked edges are not located on the same face, then we mark the others and the tetrahedron is introduced in the set of *type I* tetrahedra. Here, we can make the previous consideration too, if we compare this step with other algorithms based on the bisection by the longer edge.

In the following cases, we shall not mark any edge, i.e., any new node will not be introduced in a tetrahedron for conformity. We shall subdivide them creating subtetrahedra which will be called *transient subtetrahedra*.

e.2) If the 3 marked edges are located on the same face of the tetrahedron, then 4 transient subtetrahedra are created as Figure 2(b) shows. New edges are created by connecting the 3 new nodes one another and these with the vertex opposite to the face containing them. The tetrahedra of τ_1 with these characteristics will be inserted in the set of *type II* tetrahedra.

f) *Tetrahedra with 2 new nodes*. Also here, we shall distinguish two situations:

f.1) If the two marked edges are not located on the same face, then 4 transient subtetrahedra will be constructed from the edges conecting both new nodes and these with the vertices opposite to the two faces which contain each one of them. This tetrahedra are called *type III.a*; see Figure 2(c).

f.2) If the two marked edges are located on the same face, then 3 transient subtetrahedra are generated as Figure 2(d) shows. The face determinated by both marked edges is divided into 3 subtriangles, connecting the new node located in the longest edge with the vertex opposite and with the another new node, such that these three subtriangles and the vertex opposite to the face which contains them define three new subtetrahedra. We remark that from the two possible choices, the longest marked edge is fixed as reference in order to take advantage in some cases of the properties of the bisection by the longest edge. These tetrahedra are called *type III.b*.

g) *Tetrahedra with 1 new node*. Two transient subtetrahedra will be created as we can see in Figure 1(e). The new node is connected to the other two which are not located in the marked edge. This set of tetrahedra is called *type IV*.

h) *Tetrahedra without new node*. These tetrahedra of τ_1 are not divided and they will be inherit by the refined mesh τ_2. We call them *type V* tetrahedra; see Figure 2(f).

This classification process of the tetrahedra of τ_1 is carried out by marking their edges simply. The mesh conformity is ensured in a local level analysing the neighbourhood between the tetrahedra which contain a marked edge by an expansion process that starts in the *type I* tetrahedra of paragraph (a). Thus, when the run along this set of *type I* tetrahedra is over, the resulting mesh is conformal and locally refined.

Moreover, this is a low computational cost process, since the local expansion stops when we find tetrahedra whose edges have not to be marked. Implementations details in C++ are discussed in [3].

Generally, when we want to refine the level τ_j in which there already exist transient tetrahedra, we shall perform it in the same way as from τ_1 to τ_2, except for the following variation: if an edge of any transient tetrahedron must be marked, due to the error indicator or even to conformity reasons, then all the transient tetrahedra are eliminated from their parent (deleting process), all the parent edges are marked and this tetrahedron is introduced into the set of *type I* tetrahedra. We must remark that it will be only necessary to define a variable which determines if a tetrahedron is transient or not.

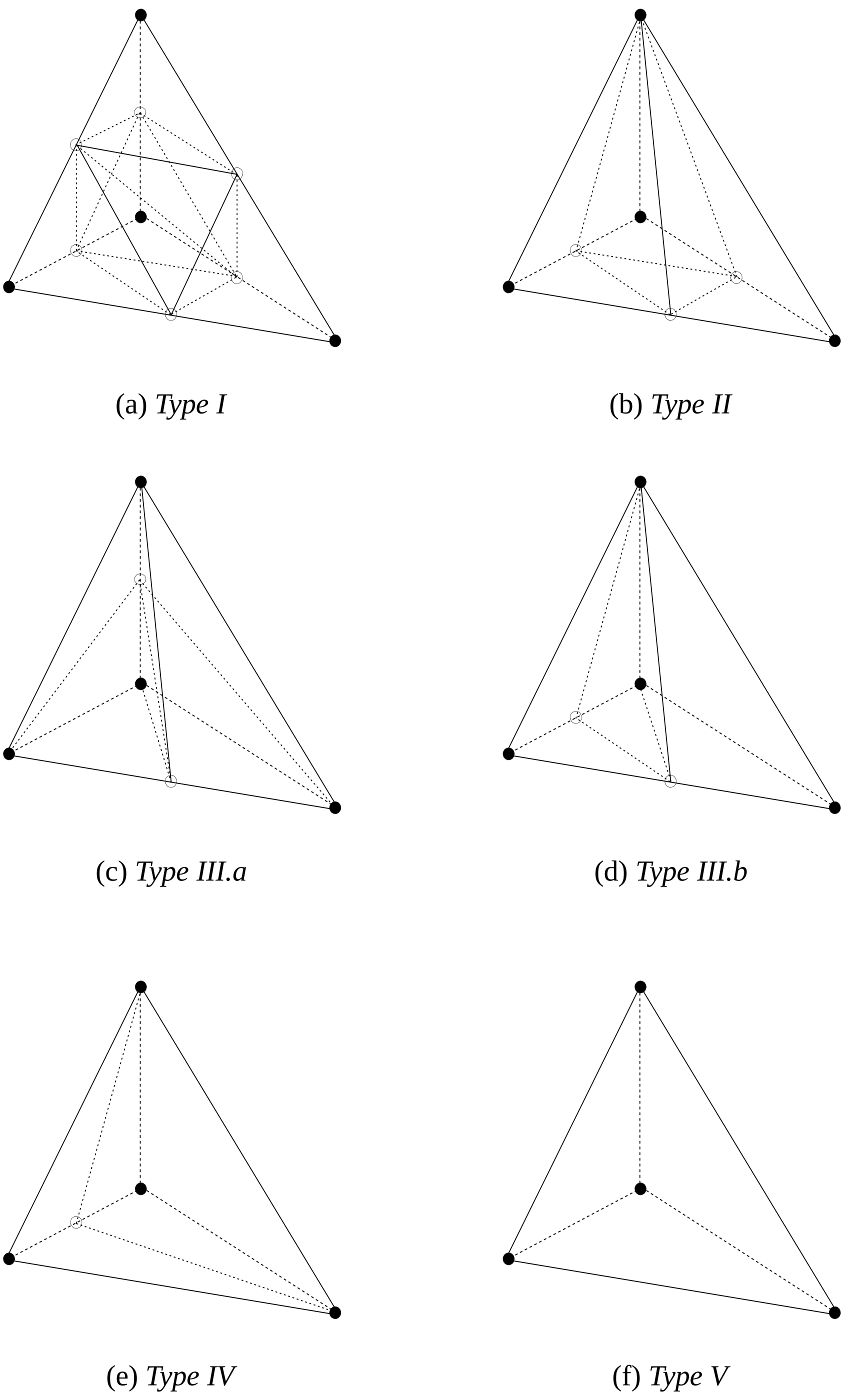


Fig. 2. Subdivision classification of a tetrahedron in function of the new nodes (white circles).

4 Numerical Experiments

In order to show the effectiveness of the refinement/smoothing combination, we consider the following test problem. We start from an initial mesh M_0 with 1364 nodes and 5387 tetrahedra. The mesh has been generated using our code studied in [1,20] and it contains 43 inverted tetrahedra. This mesh generator is based on 2-D refinement/derefinement techniques [19] and a version of the 3-D Delaunay triangulation [21]. Figure 3(a) shows a detail of the mesh with inverted and poor quality elements.

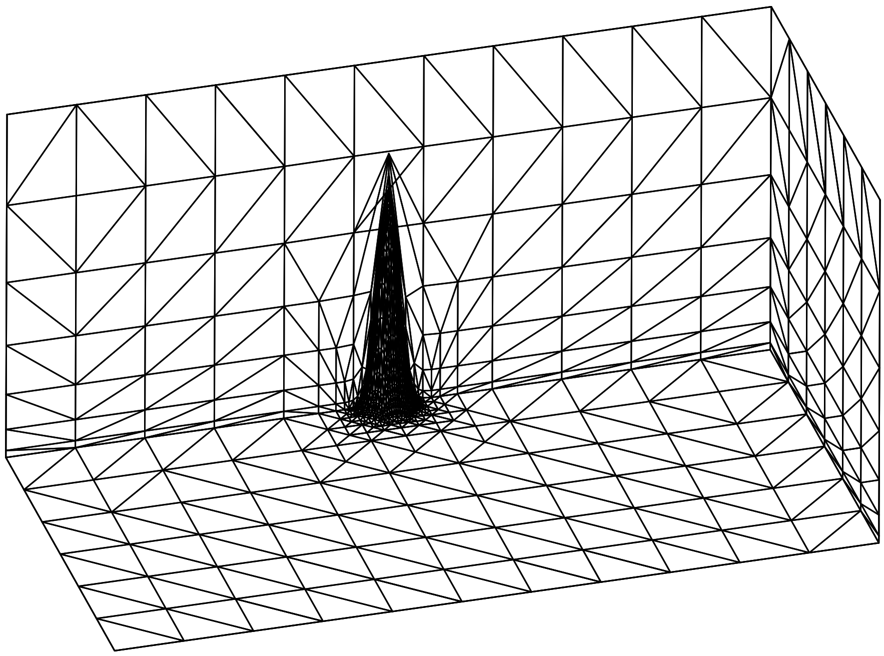
Figure 3(b) represents the mesh untangled and smoothed mesh M_0' resulting from applying a number of steps of the optimization process until the values of average and minimum quality tend to become steady to $q_{avg} = 0.6714$ y $q_{min} = 0.0925$, respectively. In this optimization process we have not allowed any node movement over the lower boundary of the domain. In the mesh M_0' we can observe elements with poor quality in the neighbourhood of the sharp surface. We remark that the quality of these elements can not be improved if we maintain the same topology of the mesh M_0 during the optimization process. So, we propose to proceed as follows.

Elements of M_0' with a quality measure near to $q_{min} = 0.0925$ are subdivided by 8-subtetrahedra and conformity of the mesh is assured. After this local refinement step, it yields the mesh M_1, with 1438 nodes and 5758 tetrahedra, see Figure 4(a). Obviously, the quality of this refined mesh is less than the one before applying the refinement process. In fact, we obtain $q_{avg} = 0.6432$ and $q_{min} = 0.0702$ for M_1.

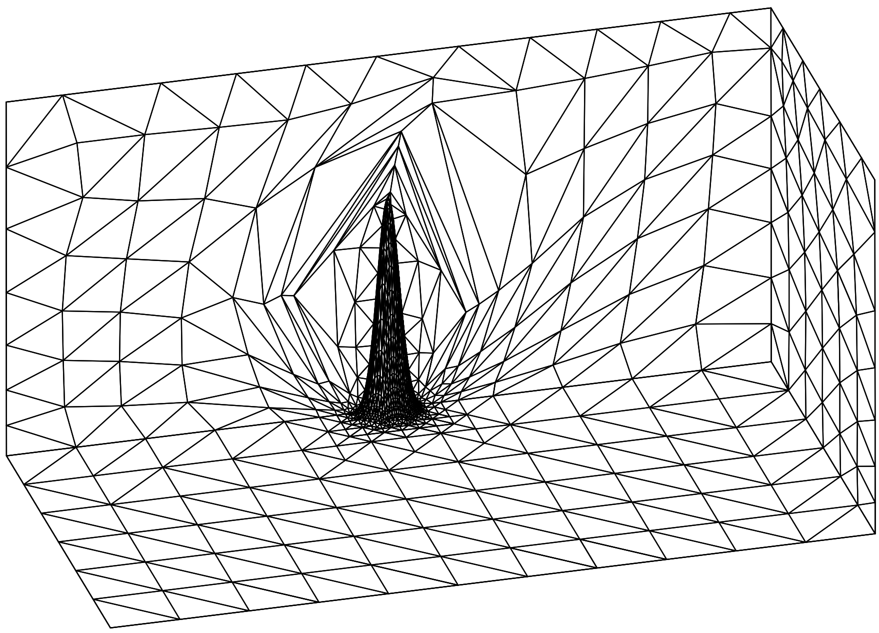
Nevertheless, due to the increasing of the node number in the neighbourhood of the regions where elements of M_0' have the worst quality, we can improve the value of minumum quality after applying the smoothing procedure over M_1. Then, we obtain the smoothed mesh M_1' in which $q_{avg} = 0.6499$ and $q_{min} = 0.1106$. Therefore, the value of q_{min} increases with respect to the corresponding value in M_0', but the value of q_{avg} decreases. Actually, in most cases it is more suitable to increase the minimum quality, that is to improve the quality of distorted elements. Besides, the relative increase obtained in q_{min} is superior than the relative decrease in q_{avg}. In Figure 4(b) it is shown a detail of the mesh M_1' in which an improvement of quality near the sharp surface can be observed.

If we now repeat the refinement/smoothing process starting from the mesh M_1', it results the mesh M_2 after refinement with 1475 nodes, 5925 tetrahedra, $q_{avg} = 0.6396$ and $q_{min} = 0.0924$. Once the smoothing procedure is applied over this mesh, we get the mesh M_2' with $q_{med} = 0.6464$ and $q_{min} = 0.1214$.

This last result implies that in the step from M_0 to M_2' the minimum quality of M_0 have been improved in a 31.2% with the introduction of a few new nodes. On the other hand, the average quality have only made worse in a 3.7%. The meshes M_2 y

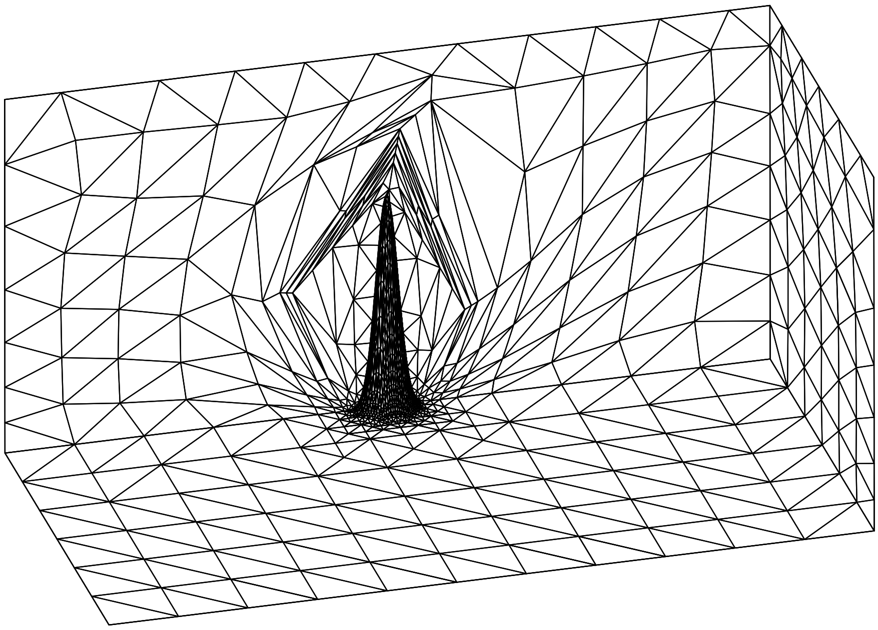


(a) M_0*: initial tangled mesh*

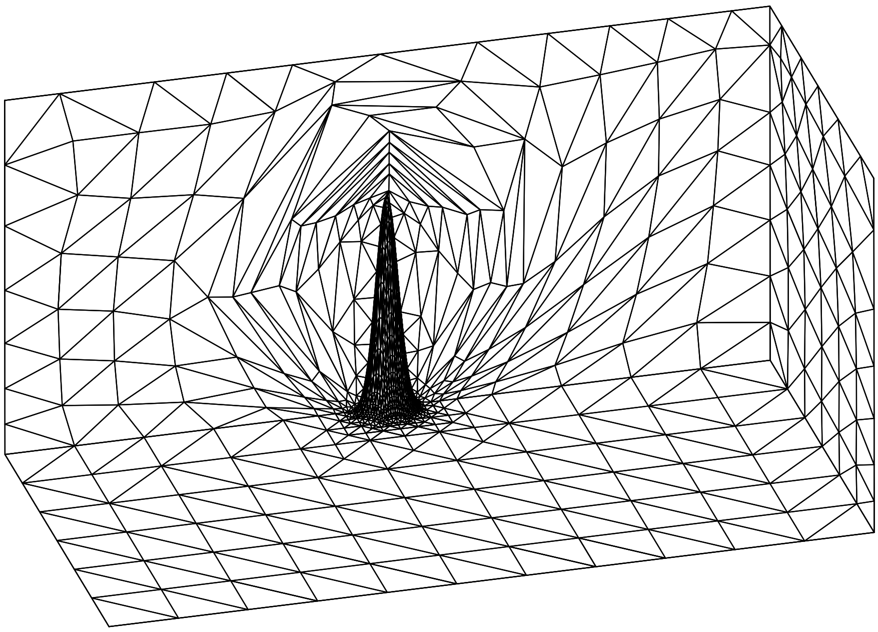


(b) M_0'*: untangled and smoothed mesh*

Fig. 3. *(a)* M_0*: initial mesh with* 43 *inverted tetrahedra and (b)* M_0'*: resulting untangled mesh after applying the optimization process over* M_0.

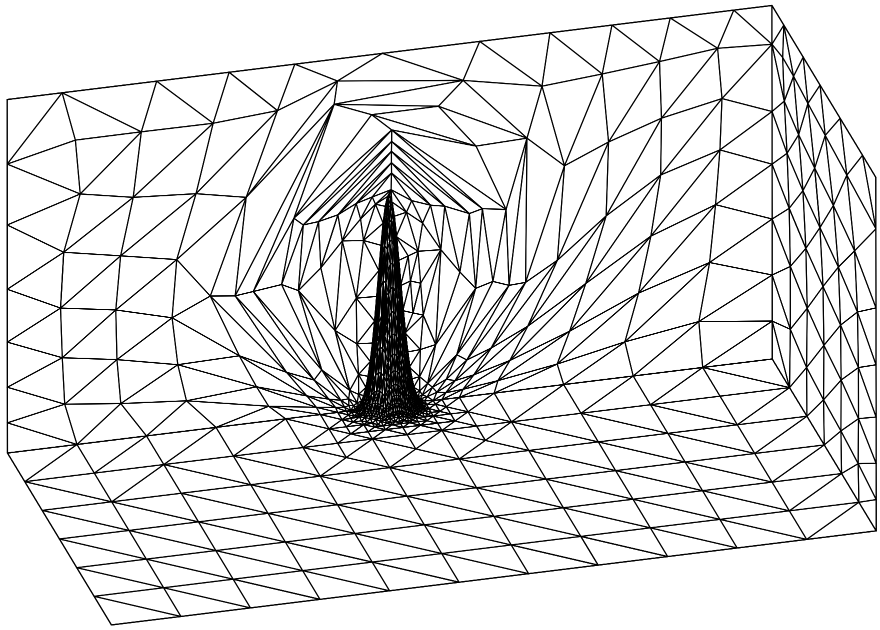


(a) M_1*: refined mesh*

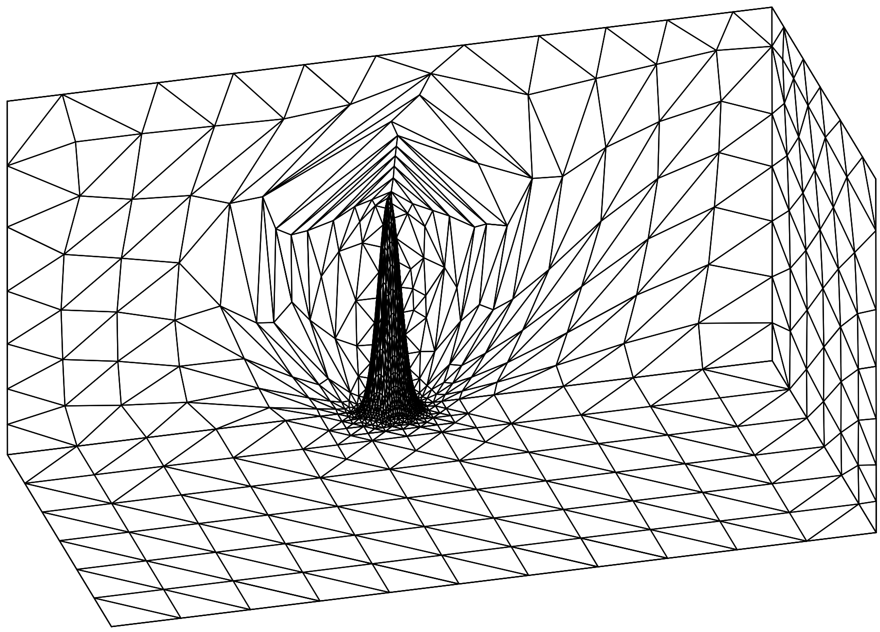


(b) M_1'*: smoothed mesh*

Fig. 4. *(a)* M_1*: resulting mesh after refining* M_0 *and (b)* M_1'*: mesh obtained after smoothing* M_1.



(a) M_2: *refined mesh*



(b) M_2': *smoothed mesh*

Fig. 5. *(a)* M_2*: resulting mesh after refining* M_1' *and (b)* M_2'*: mesh obtained after smoothing* M_2.

M_2' can been observed in Figures 5(a) and 5(b), respectively. Finally, let us remark several comments about the refinement/smoothing procedure:

a) We have used the following strategy to asure the conformity of the refined meshes. If an any transient tetrahedron must be generated, due to conformity reasons, with a quality measure less than the stablished threshold for refinement, then this transient tetrahedron and all his brothers are not created from their parent, all the parent edges are marked and this tetrahedron is introduced into the set of *type I* tetrahedra. Besides, each time that refinement algorithm is applied over a mesh, we considere all tetrahedra of this mesh as non-transient. We have obtained better results using these modifications of the refinement procedure presented in Section 3.

b) We have applied the refinement/smoothing procedure once the mesh have been untangled. We propose in future works to analyze the behaviour of this procedure when it is directly used over tangled meshes.

c) In the proposed test problem we have not permitted node movement over the lower boundary of the domain. Obviously, if we allow node recollocation over this surface, the final mesh quality would be better.

5 Conclusions

The combination of smoothing techniques and local refinement algorithms is useful to improve the minimum quality of the elements of tetrahedral meshes with very poor quality. Besides, as the proposed strategy refines a few elements in each refinement step, then the number of new nodes introduced in the initial mesh is much less than the total number. Obviously, we can repeat the refinement/smoothing combination until the required quality is obtained or no additional improvement is got. The bound of quality is associated with the topology of the initial mesh and with the constrains imposed by the boundary of the domain.

Acknowledgements

This work has been partially supported by the MCYT of Spanish Government and FEDER, grant contract REN2001-0925-C03-02/CLI.

References

[1] Montenegro R, Montero G, Escobar JM, Rodríguez E, González-Yuste JM. Tetrahedral Mesh Generation for Environmental Problems over Complex Terrains. Lecture Notes in Computer Science 2002;2329:335-344.

[2] Escobar JM, Rodríguez E, Montenegro R, Montero G, González-Yuste JM. Simultaneous Untangling and Smoothing of Tetrahedral Meshes. Comp. Methods Appl. Mech. Engrg. 2003;192:2775-2787.

[3] González-Yuste JM, Montenegro R, Escobar JM, Montero G, Rodríguez E. An Object Oriented Method for Tetrahedral Mesh Refinement. Proceedings of the Third International Conference on Engineering Computational Technology, Praga 2002; 1-18.

[4] Knupp PM. Algebraic Mesh Quality Metrics. SIAM. J. Sci. Comput. 2001;23:193-218.

[5] Knupp PM. Achieving Finite Element Mesh Quality Via Optimization of the Jacobian Matrix Norm and Associated Quantities. Part II-A Frame Work for Volume Mesh Optimization and the Condition Number of the Jacobian Matrix. Int. J. Numer. Methods Engrg. 2000;48:1165-1185.

[6] Freitag LA, Knupp PM. Tetrahedral Mesh Improvement Via Optimization of the Element Condition Number. Int. J. Numer. Methods Engrg. 2002;53:1377-1391.

[7] Freitag LA, Plassmann P. Local Optimization-based Simplicial Mesh Untangling and Improvement. Int. J. Numer. Meth. Eng., 49, 109-125, 2000.

[8] Bornemann F, Erdmann B, Kornhuber R, Adaptive Multilevel Methods in Three Space Dimensions. Int. J. Numer. Methods Engrg. 1993;36:3187-3203.

[9] Liu A, Joe B. Quality Local Refinement of Tetrahedral Meshes Based on 8-Subtetrahedron Subdivision. Mathematics of Comput. 1996;65(215):1183-1200.

[10] Lohner R, Baum JD. Adaptive h-Refinement on 3D Unstructured Grids for Transient Problems. Int. J. Num. Methods Fluids 1992;14:1407-1419.

[11] Pain CC, Umpleby AP, de Oliveira CRE, Goddard AJH. Tetrahedral Mesh Optimization for Steady-State and Transient Finite Element Calculations. Comput. Methods Appl. Mech. Engrg. 2001;190:3771-3796.

[12] Tam A, Ait-Ali-Yahia D, Robichaud MP, Moore A, Kozel V, Habashi WG. Anisotropic Mesh Adaptation for 3-D Flows on Structured and Unstructured Grids. Comput. Methods Appl. Mech. Engrg. 2000;189:1205-1230.

[13] Bazaraa MS, Sherali HD, Shetty CM. Nonlinear Programing: Theory and Algorithms. New York: John Wiley and Sons, Inc.; 1993.

[14] Dompierre J, Labbé P, Guibault F, Camarero R. Proposal of Benchmarks for 3D Unstructured Tetrahedral Mesh Optimization. Proceedings of the 7th International Meshing Roundtable, Sandia National Laboratories 1998; 459-478.

[15] Bank RE, Smith RK. Mesh Smoothing Using a Posteriori Error Estimates. SIAM J. Numer. Anal. 1997;34:979-997.

[16] Djidjev HN. Force-Directed Methods for Smoothing Unstructured Triangular and Tetrahedral Meshes. Tech. Report, Department of Computer Science, Univ. of Warwick, Coventry, UK; 2000. Available from http://www.andrew.cmu.edu/user.

[17] Bank RE, Sherman AH, Weiser A. Refinement algorithms and data structures for regular local mesh refinement. In Scientific Computing IMACS, Amsterdam: North-Holland 1983; 3-17.

[18] Rivara MC. A Grid Generator Based on 4-Triangles Conforming. Mesh-Refinement Algorithms. Int. J. Numer. Methods Engrg. 1987;24:1343-1354.

[19] Ferragut L, Montenegro R, Plaza A. Efficient Refinement/Derefinement Algorithm of Nested Meshes to Solve Evolution Problems. Comm. Numer. Methods Engrg. 1994;10:403-412.

[20] Montenegro R, Montero G, Escobar JM, Rodríguez E. Efficient Strategies for Adaptive 3-D Mesh Generation over Complex Orography. Neural, Parallel & Scientific Computation 2002;10(1):57-76.

[21] Escobar JM, Montenegro R. Several Aspects of Three-Dimensional Delaunay Triangulation. Adv. Engrg. Soft. 1996;27(1/2):27-39.