Efficient Strategies for Adaptive 3-D Mesh Generation over Complex Orography

R. Montenegro, G. Montero, J.M. Escobar and E. Rodríguez

University Institute of Intelligent Systems and Numerical Applications in Engineering, University of Las Palmas de Gran Canaria,
Edificio Instituto Polivalente, Campus Universitario de Tafira,
35017-Las Palmas de Gran Canaria, Spain.
rafa@dma.ulpgc.es, gustavo@dma.ulpgc.es,
escobar@cic.teleco.ulpgc.es, barrera@dma.ulpgc.es

Abstract
In the FEM simulation of processes that occur in a three-dimensional domain defined over an irregular terrain, it is fundamental to have a mesh generator being capable of adapting itself to the topographic characteristics. The objective of this work is to develop a code able to generate a tetrahedral mesh from an "optimal" node distribution in the domain. This last one is limited in its lower part by the terrain, and in its upper part by a horizontal plane placed in such a height that the magnitudes under study may be considered steady. The lateral walls are formed by four vertical planes. The main ideas for the construction of the initial mesh combine, on one hand, the use of a refinement/derefinement algorithm for two-dimensional domains; and, on the other hand, a tetrahedral mesh generator algorithm based on Delaunay triangulation. Moreover, it is proposed a procedure to optimise the resulting mesh. It is also analysed a function to define the vertical distance between nodes distributed in the domain, and several strategies for the appropriate generation of the corresponding set of points are proposed. Finally, in order to show the efficiency of these techniques, every strategy is applied to the construction of meshes adapted to the topography of a south part of La Palma (Canary Islands).

1. INTRODUCTION

The problem to solve has a certain difficulty due to the irregularity of the terrain surface. We intend to construct a tetrahedral mesh that respects the orography of the terrain with a specified precision. To do it, we only have a digital terrain information. Moreover, we need the mesh to adapt to the geometrical terrain characteristics; that is, the node density must be high enough to fix the orography characteristics by using a linear piecewise interpolation. The generated mesh could be used for numerical simulation of natural process, such as wind field adjustment (Winter et al., 1995; Montero et al., 1998), fire propagation (Montenegro et al., 1997), atmospheric pollution, etc. These phenomena have the main effect on the neighbouring of the terrain surface, so that it will be appropriate that the node density increases in these areas. On this mesh, that has only taken into account the geometrical characteristics of the domain, tetrahedral refinement/derefinement algorithms could be used to improve the numerical solution of the problem (Löhner & Baum, 1992; Liu & Joe, 1996). These algorithms will have special interest on time-dependent problems.

It is well known that to construct the Delaunay triangulation it will be necessary to define a set of points into the domain and on its boundary. These nodes will be precisely the vertices of the tetrahedra that make the mesh up. Point generation on our domain will be done over several layers, real or fictitious, defined from the terrain up to the upper boundary, i.e. the top of the domain. Specifically, it is proposed to construct a rectangular triangulation of this upper boundary. This two-dimensional mesh cant be obtained by applying a number of global refinement of a simple mesh defined in the input data. Another possibility is to construct a two-dimensional Delaunay triangulation of a regular node distribution. Consider the obtained mesh as the lowest level of the sequence that defines the node distribution in the remainder layers. Now, the refinement/derefinement algorithm is applied over this regular mesh (Ferragut et al.,1994; Plaza et al., 1996) to define the adaptive node distribution of the layer corresponding to the surface of the terrain. Firstly, it is constructed a function that interpolates the obtained heights from the topographic digitalisation of the rectangular area. Secondly, several global refinements are done on the mesh until a regular one, able to fit the orography, is obtained. The highest discretization degree is defined by the accuracy of the digitalisation. Then, it will be made a derefinement on these last mesh levels. To do so, it will be used as a derefinement parameter the greatest allowed heights error between the real terrain surface and the surface defined throughout the piecewise interpolation obtained with the resulting mesh. This process foundations are summarised in Section 2.

Once the node distribution is defined on the terrain and the upper boundary, we start to distribute the nodes located between both layers. This distribution can be made by different strategies, in which a vertical spacing function takes part that introduces a discretization degree that has to decrease with height, or at least keep steady. This function will be studied in Section 3.

The node distribution in the domain will be the input to a three-dimensional mesh generator based on Delaunay triangulation. To avoid conforming problems between mesh and orography, the tetrahedral mesh will be designed with the help of an auxiliary parallelepiped. Every terrain node is projected on its lower surface, and on the upper surface they are placed the nodes of the top layer at their real height.

This implies the use of a mapping, attending to the vertical spacing function, to place the remaining nodes inside the auxiliary parallelepiped. Thus, we can get sure that the maximum distance between two consecutive nodes on the same vertical of the real domain will always be less than or equal to the corresponding distance on the auxiliary parallelepiped.

Section 4 is about both the definition of the set of points in the real domain, and its transformation to the auxiliary parallelepiped where the mesh is constructed using a version of Delaunay triangulation method (Escobar & Montenegro, 1996). The final mesh quality, obtained by the inverse mapping to the real domain, depends on the node distribution defined in both domains, as the topology of the mesh obtained in the auxiliary parallelepiped remains unchanged. Four different strategies are proposed to establish the number of nodes on every vertical, and the main characteristics of each are analysed.

Once the set of points has been triangulated in the auxiliary parallelepiped, the points are placed, by the appropriate inverse transformation, in their real position, keeping the mesh topology. This process may give rise to mesh tangling that will have to be solved subsequently. It will be likewise advisable to apply a mesh optimisation to improve the elements quality in the resulting mesh. The details of the triangulation process are developed in Section 5; the ones of the mesh optimisation process are presented in Section 6.

Section 7 is about meshes generated by every strategy, stressing its advantages and disadvantages. Finally, we offer some concluding remarks.

2. DISCRETIZATION FITTED TO THE TERRAIN SURFACE

The three-dimensional mesh generation process starts by fixing the nodes placed on the terrain surface. Their distribution must be adapted to the orography in order to minimise the number of required nodes. First, a sequence of nested meshes $T = \{\tau_1 < \tau_2 < ... < \tau_m\}$ from a regular triangulation τ_1 of the rectangular area under consideration is constructed. The τ_j level is obtained by previous level τ_{j-1} by using the 4-T Rivara algorithm (Rivara, 1987). All triangles of the τ_{j-1} level are divided in four subtriangles by introducing a new node in the centres of each edge and connecting the node introduced on the longest side with the opposite vertex and with the other two introduced nodes. Thus, in the τ_j level appear new nodes, edges and elements named *proper* of j level. The number of levels m of the sequence is determined by the degree of discretization of the terrain digitalisation; in other words, the diameter of the triangulation must be about the spatial step of the digitalisation. In this way it gets assured that the mesh is capable of getting all the orographic information by an interpolation of the real heights on the mesh nodes. Finally, a new sequence $T' = \{\tau_1 < \tau'_2 < ... < \tau'_{m'}\}$, $m' \leq m$, is constructed by applying the derefinement algorithm, details may be seen in (Ferragut et al., 1994; Plaza et al., 1996). In this step it is introduced the derefinement parameter ε that fixes the precision with which we intend to approximate the terrain topography. The difference in absolute value between the resulting heights at any point of mesh $\tau'_{m'}$ and its corresponding real height will be less than ε. Moreover, the derefinement algorithm makes use of all the elements and edges genealogy defined in the sequence. The derefinement algorithm is summarised here:

INPUT: Sequence $T = \{\tau_1 < \tau_2 < ... < \tau_m\}$.

Loop in levels of T; For $j = m$ until 2, do:

1. For each proper node of τ_j the derefinement condition is evaluated and nodes and edges to be eliminated by the derefinement vectors are marked.

2. Conformity of the new mesh at level j is assured by minimising the derefinement area.

3.a. If any proper node of τ_j remains, then new nodal connections are defined for the new level j: τ_j^j. Genealogy vectors of τ_j^j and of $\tau_{j-1}$are modified accordingly.

3.b. Otherwise, the current level j is deleted from the structure vectors. Genealogy vectors of $\tau_{j-1}$are modified.

4. Changes in the mesh are inherited by the following meshes. The structure vectors are compressed.

5. A new sequence of nested meshes T^j is obtained. This sequence is the input in the next loop interaction. $T^j = \{\tau_1 < \tau_2 < ... < \tau_{j-1} < \tau_j^j < ... < \tau_{m_j}^j\}$.

OUTPUT: Derefined sequence $T' = T^2 = \{\tau_1 < \tau'_2 < ... < \tau'_{m'}\}$.

As a derefinement condition it is analysed the absolute difference between the node height under consideration and the interpolated value of the heights corresponding to both end nodes of the edge where that node was introduced during the refinement process. If that difference is less than the derefinement parameter ε, then the node could be deleted, although in some cases it should remain due to conformity reasons.

This resulting two-dimensional mesh $\tau'_{m'}$ may be modified when constructing Delaunay triangulation in the three-dimensional domain, as it is its node position the only thing we need and keep. We are also interested in keeping stored the level in which every node is proper, so as to proceed to the node generation inside the domain. This last will be used in the proposed vertical spacing strategies.

3. VERTICAL SPACING FUNCTION

As it has already been said, it is of interest to generate a set of points with the highest density in the area close to the terrain. Thus, every node is to be placed attending to a function

$$z_i = a\, i^\alpha + b \tag{1}$$

such that when the exponent $\alpha \geq 1$ increases, it provides a greater concentration of points near the terrain surface. The z_i height corresponds to the ith inserted point, in such a way that for $i = 0$ it is obtained the height of the terrain, and for $i = n$, the height of the last introduced point. This last height must coincide with the altitude h of the upper plane that bounds the domain. In this conditions the number of points defined over the vertical would be $n + 1$ and the function can be expressed

$$z_i = \frac{h - z_0}{n^\alpha} i^\alpha + z_0 \quad ; \quad i = 0, 1, 2, ..., n \tag{2}$$

It is sometimes appropriated to define the height of a point in terms of the previous one, thus avoiding the need of storing the value of z_0

$$z_i = z_{i-1} + \frac{h - z_{i-1}}{n^\alpha - (i-1)^\alpha} \left[i^\alpha - (i-1)^\alpha \right] \quad ; \quad i = 1, 2, ..., n \tag{3}$$

Generally speaking, in equations (2) or (3), once the value of α and n are fixed, the points to insert are completely defined. Nevertheless, in order to keep a minimum quality of the mesh to generate, the distance between the first inserted point ($i = 1$) and the surface of the terrain could be fixed. This will reduce to one, either α or n, the number of degrees of freedom. Consider the value of the distance d as a determined one, such that $d = z_1 - z_0$. Replacing this in equation (2),

$$d = z_1 - z_0 = \frac{h - z_0}{n^\alpha} \tag{4}$$

If we fix α and set free the value of n, from (4) it is obtained,

$$n = \left(\frac{h - z_0}{d} \right)^{1/\alpha} \tag{5}$$

Nevertheless, in practice, n will be approximated to the closest natural number. On the other hand, if we fix the value of n and set α free, it results,

$$\alpha = \frac{\log \frac{h - z_0}{d}}{\log n} \tag{6}$$

In both cases, given one of the parameters, the other can be calculated by expressions (5) or (6), respectively. In this way, the point distribution on the vertical respects the distance d between z_1 and z_0. Moreover, if the distance between the two last introduced points is fixed, that is, $D = z_n - z_{n-1}$, then the α and n parameters are perfectly defined. Let us assume that α is defined by (6). For $i = n - 1$, the equation (2) results

$$z_{n-1} = \frac{h - z_0}{n^\alpha} (n-1)^\alpha + z_0 \tag{7}$$

and thus, by using equation (6),

$$\frac{\log (n-1)}{\log n} = \frac{\log \frac{h - z_0 - D}{d}}{\log \frac{h - z_0}{d}} \tag{8}$$

From the characteristics which the mesh is defined with, it is possible to affirm *a priori* that $h - z_0 > D \geq d > 0$. Thus, the value of n will be bounded such that, $2 \leq n \leq \frac{h-z_0}{d}$, and the value of α cannot be less than 1. Moreover, to introduce at least one intermediate point between the terrain surface and the upper boundary of the domain, it must be satisfied that $d + D \leq h - z_0$.

If we call $k = \frac{\log \frac{h-z_0-D}{d}}{\log \frac{h-z_0}{d}}$, it can be easily proved that $0 \leq k < 1$. Thus, the equation (8) is transformed in

$$n = 1 + n^k \tag{9}$$

If we name $g(x) = 1 + x^k$, it can be proved that $g(x)$ is contractive in $\left[2, \frac{h-z_0}{d}\right]$ with Lipschitz constant $C = \frac{1}{2^{1-k}}$, and it is also bounded by

$$2 \leq g(x) \leq 1 + \left(\frac{h-z_0}{d}\right)^k \leq \frac{h-z_0}{d} \tag{10}$$

In view of the fixed point theorem, we can assure that equation (9) has a unique solution, and can be obtained numerically, for example, by the fixed point method, as this converge for any initial approximation chosen in the interval $\left[2, \frac{h-z_0}{d}\right]$. Nevertheless, the solution will not generally have integer values. Consequently, if its value is approximated to the closest natural number, the imposed condition with distance D will not exactly hold, but in an approximate way.

4. DETERMINATION OF THE SET OF POINTS

Whatever strategy, the point generation will be made in three stages. In the first one, it is defined a regular two-dimensional mesh τ_1 for the upper boundary of the domain with the required density of points. In second place, the mesh τ_1 will be globally refined and subsequently derefined to obtain a two-dimensional mesh $\tau'_{m'}$ capable to fit itself to the topography of the terrain. This last mesh defines the appropriate node distribution over the terrain surface. Then, we proceed to generate the set of points distributed between the upper boundary and the terrain surface. In order to do this, some points will be placed over the vertical of each node P of the terrain mesh $\tau'_{m'}$, attending to the vertical spacing function and to level j $(1 \leq j \leq m')$ where P is proper. The vertical spacing function will be determined by the strategy used to define the following parameters: the topographic height z_0 of P; the altitude h of the upper boundary; the maximum possible number of points $n+1$ in the vertical of P, including both P and the corresponding upper boundary point, if there is one; the degree of the spacing function α; the distance between the two first generated points $d = z_1 - z_0$; and the distance between the two last generated points $D = z_n - z_{n-1}$. Thus, the height of the ith point generated over the vertical of P is given by

$$z_i = \frac{h - z_0}{n^\alpha} i^\alpha + z_0 \quad ; \quad i = 1, 2, ..., n-1 \tag{11}$$

Regardless of the defined vertical spacing function, we shall use level j where P is proper to determine the definitive number of points generated over the vertical of P excluding the terrain and the upper boundary. We shall discriminate among the following cases:

1. If $j = 1$, that is to say, if node P is proper of the initial mesh τ_1, nodes are generated from equation (11) for $i = 1, 2, ..., n-1$.

2. If $2 \leq j \leq m' - 1$, we generate nodes for $i = 1, 2, ..., min(m' - j, n - 1)$.

3. If $j = m'$, that is to say, node P is proper of the finest level $\tau'_{m'}$, then any new node is generated.

This process has it justification, as mesh $\tau'_{m'}$ corresponds to the finest level of the sequence of nested meshes $T' = \{\tau_1 < \tau'_2 < ... < \tau'_{m'}\}$, obtained by the refinement/derefinement algorithm, and thus the number of introduced points decreases smoothly with altitude and they are also efficiently distributed in order to build the three-dimensional mesh in the domain.

4.1. Strategy 1: fixed number of layers and vertical spacing degree

In this case, the same value of α and n for every point P of $\tau'_{m'}$ is imposed, in order to generate points from equation (11). We can observe that parameter n lets us fix the number of layers $(n+1)$ that are to be generated on the domain, and on which points are going to be distributed. On the other hand, the value of α determines the concentration degree of layers over the terrain. In short, for $\alpha = 1$ the distance between two consecutive layers is constant over the vertical of each terrain surface point. Otherwise, if higher values of α are chosen, the layer concentration increases near the terrain. With this choice, the values of α and n are introduced as data, and thus, the set of points is completely defined. Consequently, freedom of fixing d and D is lost. This leads to the fact that those elements with any vertex on the terrain surface or on the upper boundary can have a poor quality. With this process, the regularity of the node distribution on the layers keeps increasing from one layer to the above one, being the most irregular one the corresponding to the terrain surface, and the most regular the corresponding to the upper boundary. To obtain this regularity, this strategy has been designed in such a way that, apart from eliminating the nodes of the finest level of every layer as we go to a higher layer, the own derefinement algorithm could be used on the mesh associated to a certain layer by a derefinement parameter ε variable with altitude. When transforming the coordinate corresponding to the altitude of the generated mesh points to the auxiliary parallelepiped, the surfaces that define every layer become horizontal planes.

4.2. Strategy 2: fixed number of layers and variable vertical spacing degree

In this strategy we also impose the same value of n for any point P of $\tau'_{m'}$, thus fixing the number of layers to be generated in the domain. Otherwise, the value of α is automatically determined by equation (6) in terms of the size of the elements closest to the terrain. Particularly, we have defined the value of d for each point P as the average of the side lengths of the triangles that share P in the mesh $\tau'_{m'}$. Once the spacing function for each point P is determined, with this strategy, the set of points respects the required distances between the terrain surface and the first generated layer. Thus, the possible poor quality elements could appear in the upper surface of the domain, as the value of D is imposed by the vertical spacing function. The resulting layers are less regular than the ones obtained with strategy 1. Nevertheless, if we have fixed a suitable number of layers, the elements tend to get smooth from a certain altitude that depends on the topography of the terrain. This strategy also allows to apply the algorithm of derefinement to any layer by a derefinement parameter ε variable with altitude. Finally, the transition between the set of points and the auxiliary parallelepiped is done as in strategy 1, in such a way that the surfaces that define each layer are transformed into horizontal planes. For this purpose, a fixed value of $\alpha = 1$ for any point P of $\tau'_{m'}$ has been used in the transformation. Thus, the distribution of the corresponding planes are uniform.

4.3. Strategy 3: variable number of layers and fixed vertical spacing degree

In this strategy it is imposed the same value of α for every point P of $\tau'_{m'}$. Nevertheless, the value of n is automatically determined in terms of the size of the elements closest to the terrain by equation (5), where the value of d for every point P has been determined as in strategy 2. Thus, this strategy has similar characteristics to those of the previous one with regard to the required distances between the surface terrain and the first layer. As with this strategy the number of obtained layers is different in terms of the considered point P, we can not define surfaces associated to each layer. Thus we will talk about the existence of virtual layers for each point. Finally, the transition between the set of points and the auxiliary parallelepiped is done as in strategy 1. As we have virtual layers, the transformation of the points is done according to the vertical spacing function associated to every point P of $\tau'_{m'}$.

4.4. Strategy 4: variable number of layers and vertical spacing degree

Finally, it is defined a strategy where values of α and n are automatically determined for every point P of $\tau'_{m'}$, according to the size of the elements closest to the terrain and to the upper boundary of the domain. We start by establishing the value of d as in strategies 1 and 2. A unique value of D is then fixed according to the desired distance between the last point that would be theoretically generated over the different verticals and the upper boundary. This distance is easily determined according to the size of the elements of the regular mesh τ_1. Once d and D are obtained, for every point P of $\tau'_{m'}$, their corresponding value of n is calculated by solving equation (9). Finally, the vertical spacing function is determined when obtaining the value of α by equation (6). This strategy has thus as a main characteristic that approximately holds both the required distances between the terrain surface and the first layer, and the imposed distance between the last virtual layer and the upper boundary. The transition between the set of points and the auxiliary parallelepiped is done as in the previous strategies.

5. THREE-DIMENSIONAL MESH GENERATION

Once the set of points has been defined, it will be necessary to build a three-dimensional mesh able to connect the points in an appropriate way and conforming with the domain boundary, i.e., a mesh that respects every established boundary. Although Delaunay triangulation is suitable to generate finite elements meshes with a high regularity degree for a given set of points, it is not so in which concerns to the problem of conformity with the boundary, as it generates a mesh of the convex hull of the set of points. It may be thus impossible to recover the domain boundary from the faces and edges generated by the triangulation. To avoid this, there exist two different sorts of techniques. Some, *conforming Delaunay triangulation* (Murphy & Mount, 2000), are based on the point setting following certain spacing criteria, so that the resulting triangulation is conforming with the boundary. The others, *constrained Delaunay triangulation* (George et al., 1991), are based on the swapping of edges and faces in the areas next to the non-respected boundary, so that it can be recovered. The first alternative is inadequate for our purpose, as we wish the resulting mesh to contain certain predetermined points. Moreover, given the terrain surface complexity, this strategy would imply a high computational cost. The second alternative could be appropriate, but it requires quite complex algorithms to recover the domain boundary.

To build the three-dimensional Delaunay triangulation of the domain points, we start by resetting them in an auxiliary parallelepiped, such that every point of the terrain surface is on the original coordinates x, y, but at an altitude equal to the minimum terrain height, z_{min}. In the upper plane of the parallelepiped we set the nodes of level τ_1, of the mesh sequence that defines the terrain surface, at an altitude h. Generally, the remaining points also keep their coordinates x, y, but their heights are obtained by replacing their corresponding z_0 by z_{min} in (11). The triangulation of this set of points is done using a variant of Watson incremental algorithm (Escobar & Montenegro, 1996) that solves in an effective way the problems derived from the round-off errors made at working with floating coma numbers. Once the triangulation is built in the parallelepiped, the final mesh is obtained re-establishing its original heights. This last process can be understood as a compression of the global mesh defined in the parallelepiped, such that its lowest plane becomes the terrain surface, so conformity is assured.

Sometimes it may occur that when reestablishing the position of the points to their real heights, poor quality, or even *inverted*, elements can be produced; that is, elements for which their volume V_e, evaluated as the Jacobian determinant $|J_e|$ associated to the map from reference tetrahedron to the physical one e, becomes negative. If densities of points are progressively decreasing with altitude, and if points are set over enough separated layers, the probability of poor quality or tangled elements decreases. Anyway, we need a procedure to untangle and smooth the resulting mesh, as it will be analysed in the following section. On the other hand, it must be taken into account that the possibility of getting a high quality mesh by smoothing algorithms, based on movements of nodes around their initial positions, depends, apart from the specific procedure, on the *topological quality* of the mesh. It is understood that this last one is high when every *node valence*, i.e., the number of nodes connected to it, gets close to the valence that a regular mesh formed by equilateral tetrahedra would have.

Delaunay triangulation is able to generate a high quality mesh, optimal in 2-D, for a given set of points. Thus, an appropriate approach when distributing the points will have as a consequence a high quality initial mesh. Our domain mesh keeps the topological quality of the triangulation obtained in the parallelepiped and an appropriate smoothing would thus lead to high quality meshes.

6. MESH OPTIMISATION

The most usual techniques to improve the quality of a valid triangulation, that is, that does not have inverted elements, are based upon local smoothing. In short, those techniques consist of finding the new positions that the mesh nodes must hold, in such a way that they optimise a certain objective function, based upon a certain measurement of the quality of the tetrahedra connected to the adjustable, or *free node*. As it is a local optimisation process, the final mesh cannot be obviously guaranteed to be globally optimum; nevertheless, after repeating this process several times, quite satisfactory results can be achieved. Usually, the objective functions are appropriate to improve the quality of a valid mesh, but they do not work properly when there exist inverted elements, due to the fact that they show singularity when the tetrahedra volumes change their sign. To avoid this problem we can proceed as (Freitag & Knupp, 1999), where an optimisation method consisting of two stages is proposed. In the first one, the possible inverted elements are untangled by an algorithm that maximises the negative Jacobian determinants corresponding to the inverted elements; in the second one, the resulting mesh from the first stage is smoothed. We propose here an alternative to this procedure, so the untangling and smoothing are made in the same stage. To do this, we shall use a modification of the objective function proposed in (Djidjev, 2000). Thus, let $N(v)$ be the set of the s tetrahedra attached to free node v, and $\mathbf{r} = (x, y, z)$ be its position vector. Hence, the objective function to minimise is given by

$$F(\mathbf{r}) = \sum_{e=1}^{s} f_e(\mathbf{r}) \tag{12}$$

where f_e is the objective function associated to tetrahedron e, given by

$$f_e(\mathbf{r}) = \frac{\sum_{i=1}^{6} (l_i^e)^2}{V_e^{2/3}} \tag{13}$$

and where $l_i^e (i = 1, ..., 6)$ are the edges lengths of the tetrahedron e and V_e its volume. If $N(v)$ is a valid submesh, then the minimisation of F originates positions of v for which the local mesh quality improves (Djidjev, 2000). Nevertheless, F is not bounded when the volume of any tetrahedron of $N(v)$ is null. Moreover, we cannot use F if there exist inverted tetrahedra. Then, if $N(v)$ contains any inverted or zero volume element, it will be impossible to find the relative minimum by conventional procedures, such as steepest descent, conjugate gradient, etc. To avoid this problem, we have modified function f_e in such a way that the new objective function is nearly identical to F in the minimum proximity, but being defined and regular in all $\mathbb{R}^3$. We substitute V_e in (13) by the increasing function

$$h(V_e) = \frac{1}{2}(V_e + \sqrt{V_e^2 + 4\delta^2}) \tag{14}$$

such that $\forall V_e \in \mathbb{R}$, $h(V_e) > 0$, being the parameter $\delta = h(0)$. This way, the new objective function here proposed is given by

$$\Phi(\mathbf{r}) = \sum_{e=1}^{s} \phi_e(\mathbf{r}) \tag{15}$$

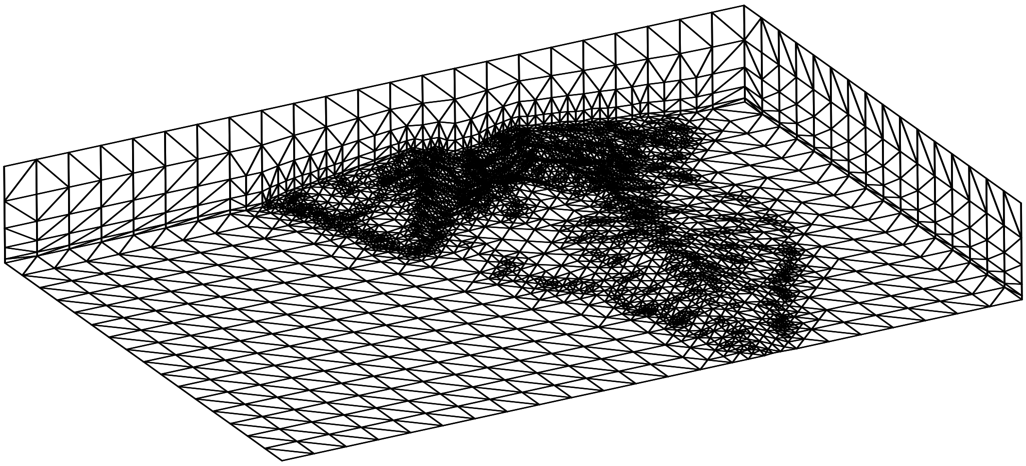
where

$$\phi_e(\mathbf{r}) = \frac{\sum_{i=1}^{6} (l_i^e)^2}{[h(V_e)]^{2/3}} \tag{16}$$

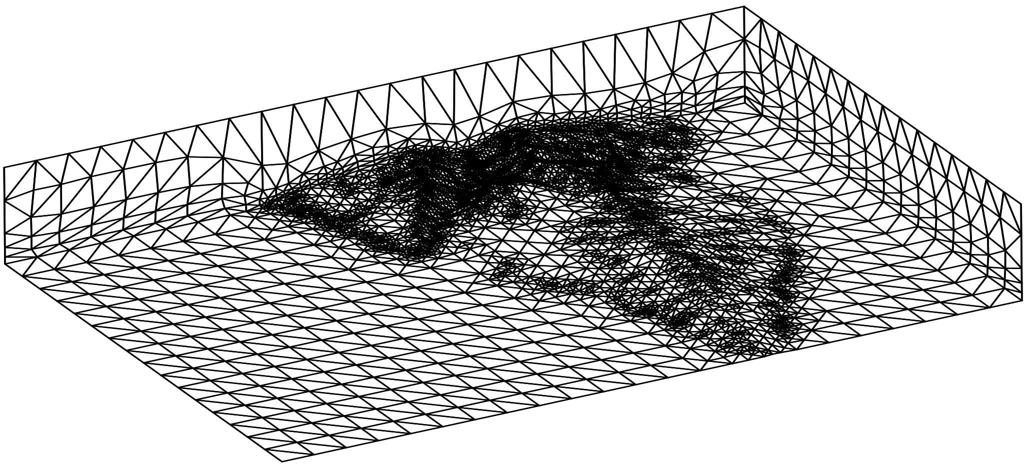
The asymptotic behaviour of $h(V_e)$, that is, $h(V_e) \approx V_e$ when $V_e \to +\infty$, will make function f_e and its corresponding modified version ϕ_e to be as close as wanted, for a value of δ small enough and positive values of V_e. On the other hand, when $V_e \to -\infty$, then $h(V_e) \to 0$. For the *most* inverted tetrahedra we shall then have a value of ϕ_e further from the minimum than for the *less* inverted ones. Moreover, with the proposed objective function Φ, the problems of F for tetrahedra with values close to zero are avoided; due to the introduction of parameter δ, the singularity of f_e disappears in ϕ_e. As smaller values of δ are chosen, function ϕ_e behaves very alike as f_e. As a result of these properties, it can be concluded that the positions of v that minimise objective functions F and Φ are nearly identical. Nevertheless, contrary to what happens to F, it is possible to find the minimum of Φ from any initial position of the free node. In particular, we can start from positions for which $N(v)$ is not a valid sub-mesh. Therefore, by using the modified objective function Φ, we can untangle the mesh and, at the same time, improve its quality. To save computational effort, F is used as objective function instead of Φ whenever $N(v)$ is an *enough* valid sub-mesh. We only turn to Φ when $N(v)$ has inverted elements or elements with nearly null quality measures. If that is the case, 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 Φ does not present any computational problem. Finally, we would like to state that the steepest descent method has been the one used to calculate the minima of the objective functions.

7. NUMERICAL EXPERIMENTS

As a practical application of the mesh generator we have taken under consideration a rectangular area in the south of Isla de La Palma (Canary Islands) of 45.6×31.2 km, where extreme heights vary from 0 to 2279 m. The upper boundary of the domain has been placed at $h = 9$ km. To define the topography we had a digitalisation of the area where heights were defined over a grid with a spacing step of 200 m in directions x and y. Starting from a uniform mesh τ_1 of the rectangular area with a size of elements about 2×2 km, four global refinements were made using Rivara 4-T algorithm (Rivara, 1987). Once the data were interpolated on this refined mesh, it was used the derefinement algorithm developed in (Ferragut et al., 1994; Plaza et al., 1996) with a derefinement parameter of $\varepsilon = 40$ m. Thus, the adapted mesh approximates the terrain surface with an error less than 40 m; see figures 1-4. The node distribution of the regular initial mesh τ_1, used before the global refinements, is the one considered on the upper boundary of the domain. Figure 1(a) represents the three-dimensional mesh generated by using strategy 1, with a total number of six layers, which means a choice of $n = 5$. To define the vertical spacing degree it was considered $\alpha = 2$. The number of generated tetrahedra is 52945, with 11578 nodes, and a maximum valence of 21. We can thus see that we get a poor quality mesh in very small elements close to the terrain; nevertheless, the quality of elements on the upper boundary is quite satisfactory. Figure 1(b) represents the mesh resulting from applying five sweeps of the optimisation process to the previous mesh. We would like to stress that in the optimisation process of meshes generated with the different strategies, the displacement of nodes on the terrain surface and the upper boundary has not been allowed. In figure 1(b) we can see a greater node density on the irregular terrain surface, and, on the contrary, a slight displacement of the rest of the nodes to the upper boundary. The result is a mesh of a better quality than the initially generated one (see figure 5(a)), but with a worse node distribution in the upper areas of the domain. We can also see that this strategy does not originate tangling with the proposed layer disposition of nodes, i.e., $q(e) > 0$ for every tetrahedron e, being $q(e)$ the quality measure proposed in (Freitag & Knupp, 1999). Therefore, the optimisation process just needs a mesh smoothing. Figure 2(a) shows the mesh obtained using strategy 2, and fixing a number of layers similar to the previous case.

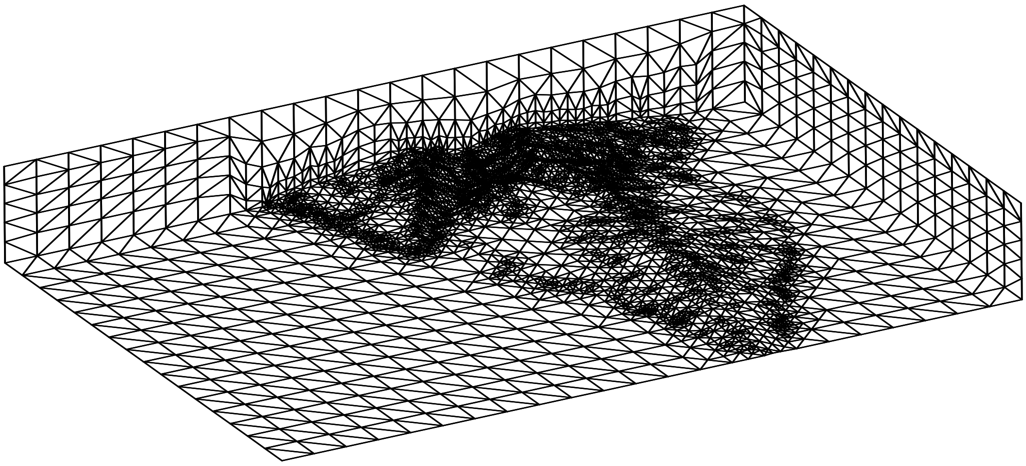


(a) *Generated mesh*

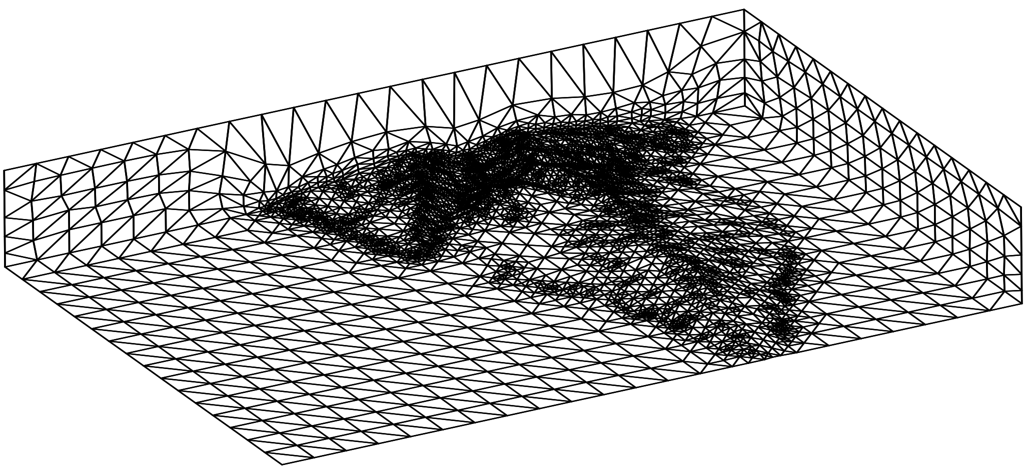


(b) *Optimised mesh*

Figure 1: *(a) mesh generated by using strategy 1 and (b) resulting mesh after five sweeps of the optimisation process.*

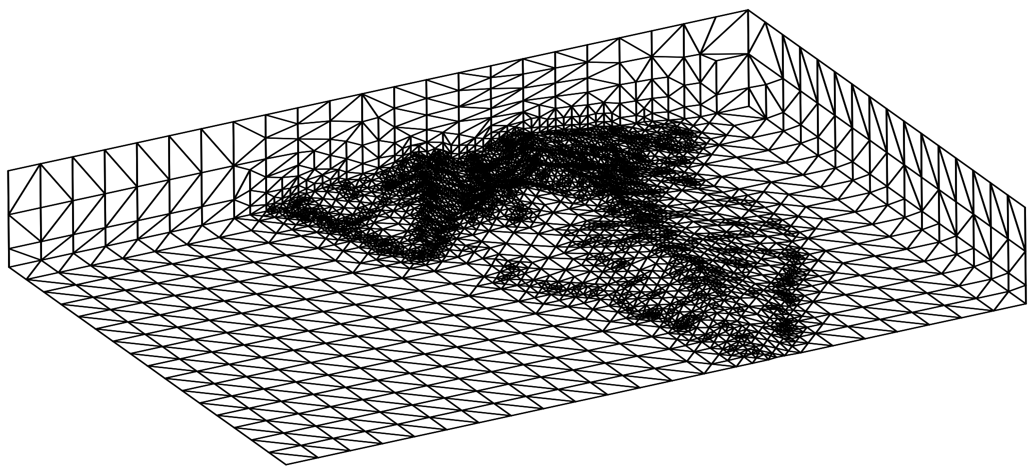


(a) *Generated mesh*

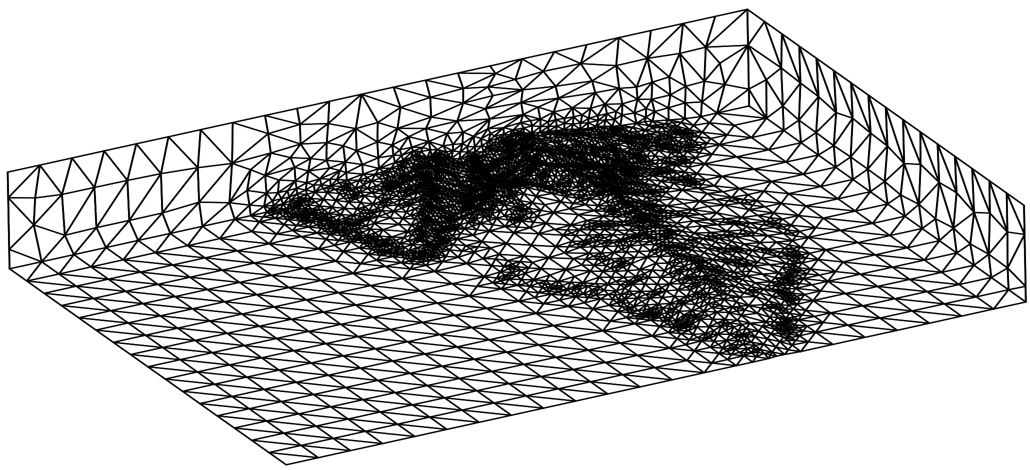


(b) *Optimised mesh*

Figure 2: *(a) mesh generated by using strategy 2 and (b) resulting mesh after five sweeps of the optimisation process.*

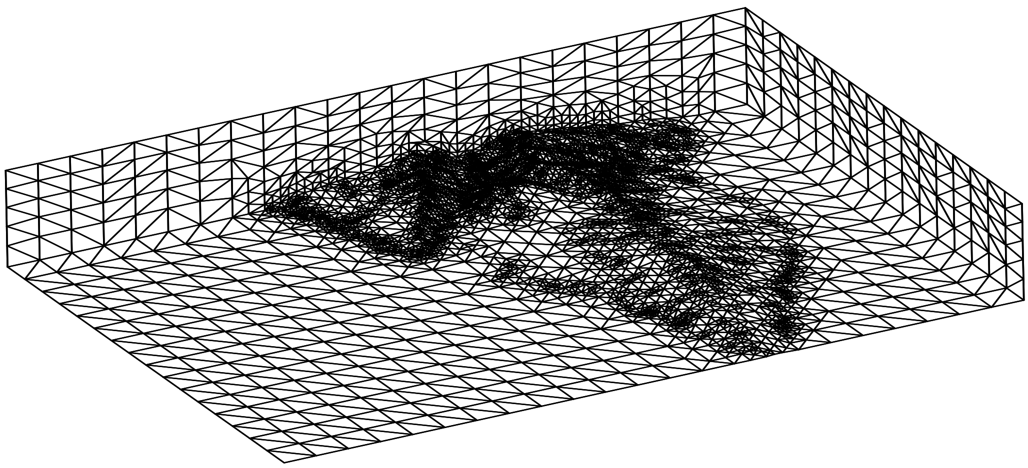


(a) *Generated mesh*

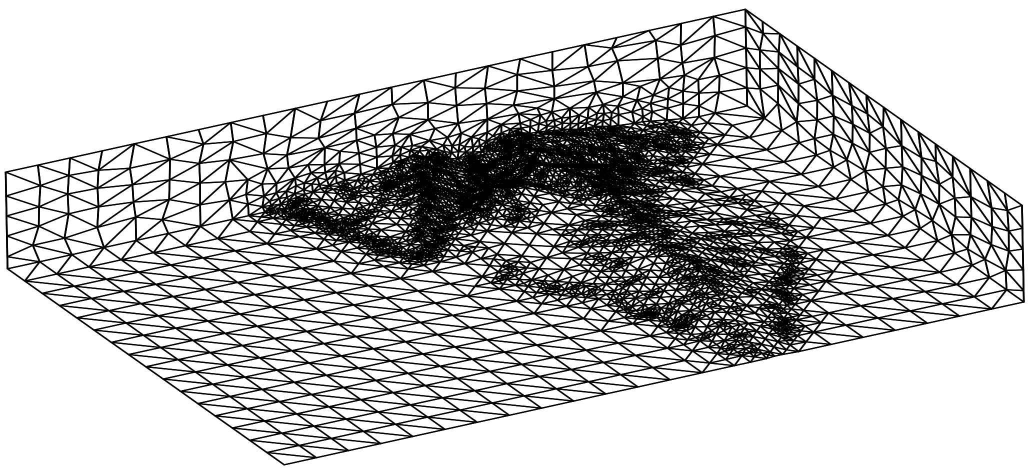


(b) *Optimised mesh*

Figure 3: *(a) mesh generated by using strategy 3 and (b) resulting mesh after five sweeps of the optimisation process.*

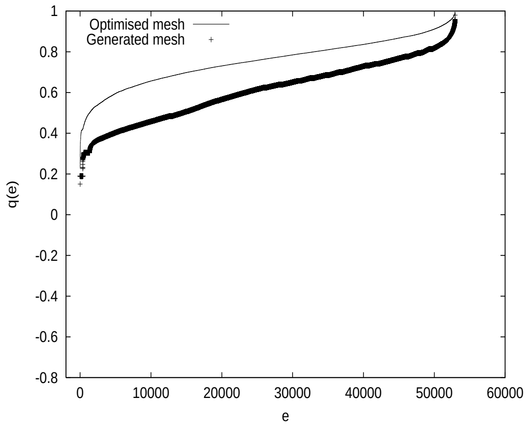


(a) *Generated mesh*

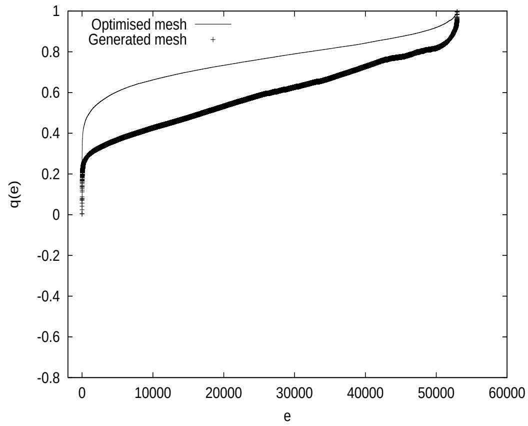


(b) *Optimised mesh*

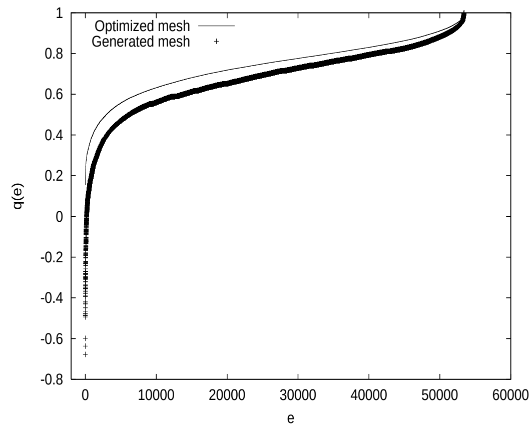
Figure 4: *(a) mesh generated by using strategy 4 and (b) resulting mesh after five steps of the optimisation process.*



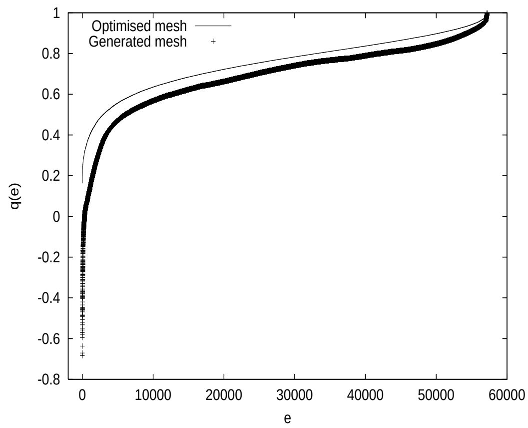
(a) *Strategy 1*



(b) *Strategy 2*



(c) *Strategy 3*



(d) *Strategy 4*

Figure 5: *Quality curves of meshes generated and optimised for the different strategies.*

The number of tetrahedra and nodes, and the maximum valence of this mesh, are the same than the ones in strategy 1. In this case, by letting the vertical spacing degree to adjust automatically, we get in the areas close to the terrain a better quality mesh than the one in the previous case. Moreover, the quality in the upper boundary is quite satisfactory. The result of the optimisation process on this mesh is similar to the previous case (see figure 2(b)), with a highest node density on the terrain, while in the rest there are hardly alterations due to the good initial quality of the mesh in these areas. Figure 5(b) shows the mesh quality curves corresponding to strategy 2 and its optimised one. Note that, as in the previous case, this strategy does not cause tangling. Figure 3(a) shows the mesh obtained using strategy 3, imposing the vertical spacing degree $\alpha = 1.5$, and letting the number of virtual layers over each point vertical to adjust automatically. The mesh thus obtained adapts itself to the existing orography, but not to the required tetrahedra size for areas near the upper boundary. The mesh has 53432 tetrahedra and 11173 nodes, with a maximum valence of 26. Figure 3(b) shows the mesh optimised after five process stages. Variations on mesh node densities are not as considerable as in the two previous cases. In fact, differences between the quality curves represented in figure 5(c) are not as sharp as in the preceding strategies, but there exists tangling that has been solved with an only sweep of the optimisation process. The result obtained with strategy 4 is shown in figure 4(a), fixing as the only parameter distance $D = 1.5\ km$. It can be seen that the distances between the lower and upper boundaries are kept. In this case, the mesh has 57193 tetrahedra and 11841 nodes, with a maximum valence of 26. The node distribution obtained with this strategy has such a quality that it is hardly modified after five sweeps of the optimisation process (see figure 4(b)), although there is initial tangling that is nevertheless efficiently solved (see figure 5(d)). With both first strategies we have obtained adapted meshes admissible for the good quality application of the finite elements method. On the other hand, in both last strategies, although the point distribution is more automatic, there exist some inverted tetrahedra in the resulting mesh, mainly in strategy 4. With this last strategy it is obtained a quasi-optimum point distribution in the real domain attending to the size of the elements originated on the terrain surface and the upper boundary. To avoid inverted tetrahedra, the technique proposed in the previous section has been efficiently applied. Moreover, the worst quality measure of the optimised meshes tetrahedra for the different strategies is about 0.2. We remark that the number of parameters necessary to define the resulting mesh is really low, as well as the computational cost.

8. CONCLUSIONS

We have established the main aspects to generate a three-dimensional mesh able to adapt to the topography of a rectangular area with a minimum intervention of users. In short, it has been stated a point generation, well distributed in the domain under study, able to get the topographic information of the terrain, and with a decreasing density as altitude increases in relation to the terrain. Points are generated using refinement/derefinement techniques in 2-D and the vertical spacing function here introduced. Four strategies have been proposed to define efficiently a set of points in a three-dimensional domain where the lower boundary corresponds to an irregular terrain. Next, with the help of an auxiliary parallelepiped, it has been set forth a proceeding based on Delaunay triangulation to generate the mesh automatically, assuring the conformity with the terrain surface. Nevertheless, the obtained point distribution could also be of interest to generate the three-dimensional mesh with other classical techniques, such as advancing front (Jin & Tanner, 1993) and normal offsetting (Johnston & Sullivan, 1993). Finally, the procedure here proposed for optimising the generated mesh let us solve at the same time the tangling problems and the mesh quality.

REFERENCES

1. Djidjev, H.N. (2000). Force-directed methods for smoothing unstructured triangular and tetrahedral meshes, *Tech. Report*, Dep. of Computer Science, Univ. of Warwick, Coventry, UK. Available from http://www.andrew.cmu.edu/user/sowen/topics/new.html.

2. Escobar,J.M. & Montenegro, R. (1996). Several aspects of three-dimensional Delaunay triangulation, *Advances in Engineering Software*, v. 27, 1/2, pp. 27-39.

3. Ferragut, L., Montenegro, R. & Plaza, A. (1994). Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems, *Comm. Num. Meth. Eng.*, v. 10, pp. 403-412.

4. Freitag, L.A. & Knupp, P.M. (1999). Tetrahedral element shape optimization via the Jacobian determinant and condition number, in *Proceedings of the Eighth International Meshing Roundtable*, Sandia National Laboratories, pp. 247-258.

5. George, P.L., Hecht, F. & Saltel, E. (1991). Automatic Mesh Generation with Specified Boundary, *Comp. Meth. Appl. Mech. Eng.*, v. 92, pp. 269-288.

6. Jin, H. & Tanner, R.I. (1993). Generation of unstructured tetrahedral meshes by advancing front technique, *Int. J. Num. Meth. Eng.*, v. 36, pp. 1805-1823.

7. Johnston, B.P. & Sullivan, J.M., Jr. (1993). A normal offsetting technique for automatic mesh generation in three dimensions, *Int. J. Num. Meth. Eng.*, v. 36, pp. 1717-1734.

8. Liu, A. & Joe, B. (1996). Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision, *Mathematics of Computations*, v. 65, 215, pp. 1183-1200.

9. Löhner, R. & Baum, J.D. (1992). Adaptive h-refinement on 3D unstructured grids for transient problems, *Int. J. Num. Meth. Fluids*, v. 14, pp. 1407-1419.

10. Montenegro, R., Plaza, A., Ferragut, L. & Asensio, I. (1997). Application of a nonlinear evolution model to fire propagation, *Nonlinear Analysis, Th., Meth. & App.*, v. 30, 5, pp. 2873-2882.

11. Montero, G., Montenegro, R. & Escobar, J.M. (1998). A 3-D diagnostic model for wind field adjustment, *J. of Wind Eng. and Ind. Aerodynamics*, v. 74-76, pp. 249-261.

12. Murphy, M., Mount, D.M. & Gable, C.W. (2000). A point-placement strategy for conforming Delaunay tetrahedralization", in *Symposium on Discrete Algorithms*, pp. 67-74.

13. Plaza, A., Montenegro, R. & Ferragut, L. (1996). An improved derefinement algorithm of nested meshes, *Advances in Engineering Software*, v.27, 1/2, pp. 51-57.

14. Rivara, M.C. (1987). A grid generator based on 4-triangles conforming. Mesh-refinement algorithms, *Int. J. Num. Meth. Eng.*, v. 24, pp. 1343-1354.

15. Winter, G., Montero, G., Ferragut, L. & Montenegro, R. (1995). Adaptive strategies using standard and mixed finite elements for wind field adjustment, *Solar Energy*, v. 54, 1, pp. 49-56.