Tetrahedral Mesh Generation for Environmental Problems over Complex Terrains*

Rafael Montenegro, Gustavo Montero, José María Escobar, Eduardo Rodríguez, and José María González-Yuste**

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, gustavo}@dma.ulpgc.es, escobar@cic.teleco.ulpgc.es, barrera@dma.ulpgc.es, josem@sinf.ulpgc.es

Abstract. In the finite element simulation of environmental processes that occur in a three-dimensional domain defined over an irregular terrain, a mesh generator capable of adapting itself to the topographic characteristics is essential. The present study develops a code for generating a tetrahedral mesh from an "optimal" node distribution in the domain. The main ideas for the construction of the initial mesh combine the use of a refinement/derefinement algorithm for two-dimensional domains and a tetrahedral mesh generator algorithm based on Delaunay triangulation. Moreover, we propose a procedure to optimise the resulting mesh. A function to define the vertical distance between nodes distributed in the domain is also analysed. Finally, these techniques are applied to the construction of meshes adapted to the topography of the southern section of La Palma (Canary Islands).

1 Introduction

The problem in question presents certain difficulties due to the irregularity of the terrain surface. Here we construct a tetrahedral mesh that respects the orography of the terrain with a given precision. To do so, we only have digital terrain information. Furthermore, it is essential for the mesh to adapt to the geometrical terrain characteristics. In other words, node density must be high enough to fix the orography by using a linear piecewise interpolation. Our domain is limited in its lower part by the terrain and in its upper part by a horizontal plane placed at a height at which the magnitudes under study may be considered steady. The lateral walls are formed by four vertical planes. The generated mesh could be used for numerical simulation of natural processes, such as wind field adjustment [9], fire propagation [8] and atmospheric pollution. These phenomena have the main effect on the proximities of the terrain surface. Thus node density increases in these areas accordingly.

* Partially supported by MCYT, Spain. Grant contract: REN2001-0925-C03-02/CLI
** The authors acknowledge Dr. David Shea for editorial assistance

To construct the Delaunay triangulation, we must define a set of points within the domain and on its boundary. These nodes will be precisely the vertices of the tetrahedra that comprise the mesh. 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, we propose the construction of a regular triangulation of this upper boundary. Now, the refinement/derefinement algorithm [3, 11] is applied over this regular mesh to define an adaptive node distribution of the layer corresponding to the surface of the terrain. These process foundations are summarised in Sect. 2. Once the node distribution is defined on the terrain and the upper boundary, we begin to distribute the nodes located between both layers. A vertical spacing function, studied in Sect. 3, is involved in this process.

The node distribution in the domain will be the input to a three-dimensional mesh generator based on Delaunay triangulation [2]. To avoid conforming problems between mesh and orography, the tetrahedral mesh will be designed with the aid of an auxiliary parallelepiped. Section 4 is concerned with both the definition of the set of points in the real domain and its transformation to the auxiliary parallelepiped where the mesh is constructed. Next, 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. We should, then, apply a mesh optimisation to improve the quality of the elements in the resulting mesh. The details of the triangulation process are developed in Sect. 5; those related to the mesh optimisation process are presented in Sect. 6. Numerical experiments are shown in Sect. 7, and, finally, we offer some concluding remarks.

2 Adaptive Discretization of 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 to minimise the number of required nodes. First, we construct a sequence of nested meshes $T = \{\tau_1 < \tau_2 < ... < \tau_m\}$ from a regular triangulation τ_1 of the rectangular area under consideration. The τ_j level is obtained by previous level τ_{j-1} using the 4-T Rivara algorithm [12]. All triangles of the τ_{j-1} level are divided in four sub-triangles 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, new nodes, edges and elements named *proper* of level j appear in the τ_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 approximately the spatial step of the digitalisation. In this way we ensure that the mesh is capable of obtaining all the topographic information by an interpolation of the actual 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 [3, 11]. In this step we present 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 the mesh $\tau'_{m'}$ and its corresponding real height will be less than ε.

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

3 Vertical Spacing Function

As stated above, we are interested in generating a set of points with higher density in the area close to the terrain. Thus, every node is to be placed in accordance with the following function

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

so 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 $i = 0$ the height of the terrain is obtained, 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 these conditions the number of points defined over the vertical is $n + 1$ and (1) becomes

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

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

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

In (2) or (3), once the values of α and n are fixed, the points to insert are completely defined. Nevertheless, to maintain acceptable minimum quality of the generated mesh, 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 $d = z_1 - z_0$. Using (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) we obtain

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

Nevertheless, in practice, n will be approximated to the closest integer number. Conversely, if we fix the value of n and set α free, we get

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

In both cases, given one of the parameters, the other may 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 last two 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 $i = n-1$, (2) could be expressed as

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

and thus, by using (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 define the mesh, we may 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, we must verify 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 $0 \leq k < 1$. So, (8) yields

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

If we name $g(x) = 1 + x^k$, it can be demonstrated 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 ensure that (9) has a unique solution which can be obtained numerically, for example, by the fixed point method, as this converges 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 integer number, the imposed condition with distance D will not exactly hold, but approximately.

4 Determination of the Set of Points

The point generation will be carried out in three stages. In the first, we define a regular two-dimensional mesh τ_1 for the upper boundary of the domain with the required density of points. Second, the mesh τ_1 will be globally refined and subsequently derefined to obtain a two-dimensional mesh $\tau'_{m'}$ capable of fitting itself to the topography of the terrain. This last mesh defines the appropriate node distribution over the terrain surface. Next, we 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 (2) for $i = 1, 2, ..., n-1$.

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, if node P is proper of the initial mesh τ_1, nodes are generated from (2) 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, node P is proper of the finest level $\tau'_{m'}$, then any new node is generated.

This process has its 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. 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.

We set out a particular 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. First, the value of d for each point P is established as the average of the side lengths of the triangles that share P in the mesh $\tau'_{m'}$. 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 directly 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 (9). Finally, the vertical spacing function is determined when obtaining the value of α by (6). This strategy approximately respects 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.

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 which conforms with the domain boundary, i.e., a mesh that respects every established boundary.

Although Delaunay triangulation is suitable to generate finite element meshes with a high regularity degree for a given set of points, this does not occur in 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, we have two different sorts of techniques: *conforming Delaunay triangulation* [10] and *constrained Delaunay triangulation* [5]. 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 provide another solution, 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, so 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 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 (2). The triangulation of this set of points is done using a variant of Watson incremental algorithm [2] that effectively solves the problems derived from the round-off errors made when working with floating coma numbers.

Once the triangulation is built in the parallelepiped, the final mesh is obtained by re-establishing its original heights. This latter process can be understood as a compression of the global mesh defined in the parallelepiped, such that its lowest plane becomes the terrain surface. In this way, conformity is ensured.

Sometimes when re-establishing the position of each point to its real height, poor quality, or even *inverted* elements may occur. For inverted elements, their volume V_e, evaluated as the Jacobian determinant $|J_e|$ associated with the map from reference tetrahedron to the physical one e, becomes negative. For this reason, we need a procedure to untangle and smooth the resulting mesh, as analysed in Sect. 6.

We must also take into account the possibility of getting a high quality mesh by smoothing algorithms, based on movements of nodes around their initial positions, depends on the *topological quality* of the mesh. It is understood that this quality is high when every *node valence*, i.e., the number of nodes connected to it, approaches the valence corresponding to a regular mesh formed by *quasi-equilateral* tetrahedra.

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 accepted techniques for improving valid triangulation quality are based upon local smoothing. In short, these techniques locate the new positions that the mesh nodes must hold so that they optimise a certain objective function based upon a quality measurement of the tetrahedra connected to the adjustable or free node. The objective functions are generally useful for improving the quality of a valid mesh. They do not work properly, however, in the case of inverted elements, since they show singularity when the tetrahedra volumes change their sign. To avoid this problem we can proceed as in [4], where an optimisation method consisting of two stages is proposed. In the first, the possible inverted elements are untangled by an algorithm that maximises the negative Jacobian determinants corresponding to the inverted elements. In the second, the resulting mesh from the first stage is smoothed. We propose here an alternative to this procedure in which the untangling and smoothing are performed in the same stage. To do this, we shall use a modification of the objective function proposed in [1]. 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 function to minimise is given by

$$F(\mathbf{r}) = \sum_{e=1}^{s} f_e(\mathbf{r}) = \sum_{e=1}^{s} \frac{\sum_{i=1}^{6} (l_i^e)^2}{V_e^{2/3}} \ . \tag{11}$$

where f_e is the objective function associated to tetrahedron e, l_i^e $(i = 1, ..., 6)$ are the edge lengths of the tetrahedron e and V_e its volume. If $N(v)$ is a valid sub-mesh, then the minimisation of F originates positions of v for which the local mesh quality improves [1]. Nevertheless, F is not bounded when the volume of any tetrahedron of $N(v)$ is null. Moreover, we cannot use F if there are inverted tetrahedra. Thus, if $N(v)$ contains any inverted or zero volume elements, it will be impossible to find the relative minimum by conventional procedures, such as steepest descent, conjugate gradient, etc. To remedy this situation, 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 (11) by the increasing function

$$h(V_e) = \frac{1}{2}(V_e + \sqrt{V_e^2 + 4\delta^2}) \ . \tag{12}$$

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

$$\Phi(\mathbf{r}) = \sum_{e=1}^{s} \phi_e(\mathbf{r}) = \sum_{e=1}^{s} \frac{\sum_{i=1}^{6} (l_i^e)^2}{[h(V_e)]^{2/3}} \ . \tag{13}$$

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 as close as required, 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 much like f_e. As a result of these properties, we may conclude 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. 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 state that the steepest descent method has been the one used to calculate the minimum of the objective function.

7 Numerical Experiments

As a practical application of the mesh generator, we have considered a rectangular area in the south of La Palma island 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 used 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 an element size of about 2×2 km, four global refinements were made using Rivara 4-T algorithm [12]. Once the data were interpolated on this refined mesh, we applied the derefinement algorithm developed in [3, 11] with a derefinement parameter of $\varepsilon = 40$ m. Thus, the adapted mesh approximates the terrain surface with an error less than that value. The node distribution of τ_1 is the one considered on the upper boundary of the domain.

The result obtained is shown in Fig. 1, fixing as the only parameter distance $D = 1.5$ km. 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 steps of the optimisation process, although there is initial tangling that is nevertheless efficiently solved; see Fig. 2, where $q(e)$ is the quality measure proposed in [4] for tetrahedron e. In fact, to avoid inverted tetrahedra, the technique proposed in Sect. 6 has been efficiently applied. Moreover, the worst quality measure of the optimised mesh tetrahedra is about 0.2.

We note that the number of parameters necessary to define the resulting mesh is quite low, as well as the computational cost. In fact, the complexity of 2-D refinement/derefinement algorithm is linear [11]. Besides, in experimental results we have approximately obtained a linear complexity in function of the

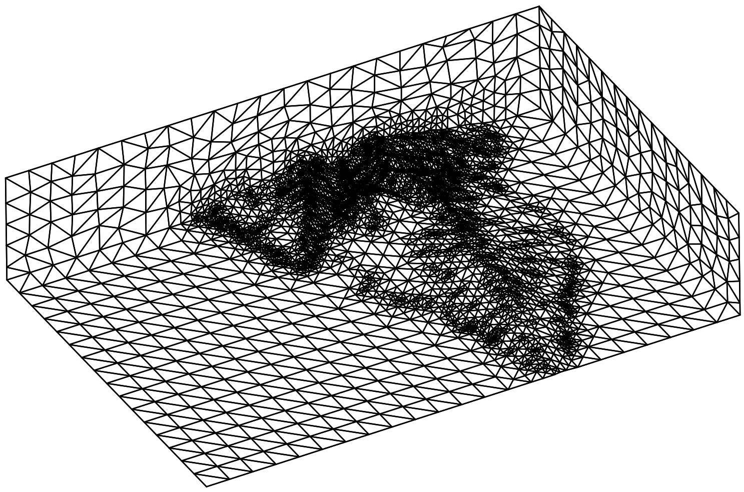

Fig. 1. Resulting mesh after five steps of the optimisation process

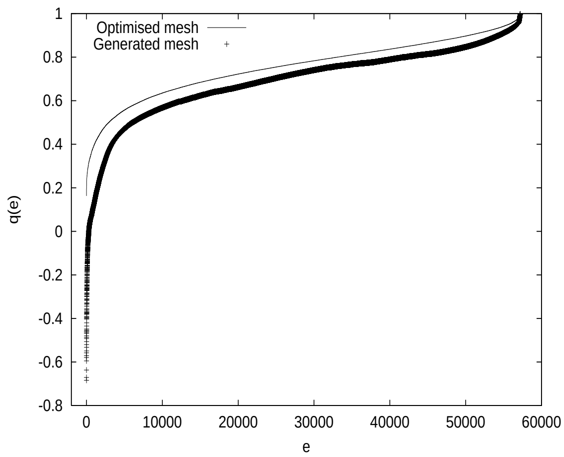

Fig. 2. Quality curves of the generated mesh and the resulting mesh after five steps of the optimisation process. Function $q(e)$ is a quality measure for tetrahedron e

number of points for our algorithm of 3-D Delaunay triangulation [2]. In the present application only a few seconds of CPU time on a Pentium III were necesary to construct the mesh before its optimisation. Finally, the complexity of each step of the mesh optimisation process is also linear. In practice we have found acceptable quality meshes appling a limited number of steps of this latter algorithm.

8 Conclusions

We have established the main aspects for generating a three-dimensional mesh capable of adapting to the topography of a rectangular area with minimal user intervention. In short, an efficient and adaptive point generation has been stated which is well distributed in the domain under study, because it preserves the topographic information of the terrain with a decreasing density as altitude increases. Points are generated using refinement/derefinement techniques in 2-D and the vertical spacing function here introduced. Next, with the aid of an auxiliary parallelepiped, a proceeding based on Delaunay triangulation has been set forth to generate the mesh automatically, assuring conformity with the terrain surface. Nevertheless, the obtained point distribution could also be useful in generating the three-dimensional mesh with other classical techniques, such as advancing front [6] and normal offsetting [7]. Finally, the procedure here proposed for optimising the generated mesh allows us to solve the tangling problems and mesh quality at the same time.

References

1. Djidjev, H.N.: 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
2. Escobar, J.M., Montenegro, R.: Several Aspects of Three-Dimensional Delaunay Triangulation. Adv. Eng. Soft. **27**(1/2) (1996) 27-39
3. Ferragut, L., Montenegro, R., Plaza, A.: Efficient Refinement/Derefinement Algorithm of Nested Meshes to Solve Evolution Problems. Comm. Num. Meth. Eng. **10** (1994) 403-412
4. Freitag, L.A., Knupp, P.M.: Tetrahedral Element Shape Optimization Via the Jacobian Determinant and Condition Number. In Proceedings of the Eighth International Meshing Roundtable. Sandia National Laboratories (1999) 247-258
5. George, P.L., Hecht, F., Saltel, E.: Automatic Mesh Generation with Specified Boundary. Comp. Meth. Appl. Mech. Eng. **92** (1991) 269-288
6. Jin, H., Tanner, R.I.: Generation of Unstructured Tetrahedral Meshes by Advancing Front Technique. Int. J. Num. Meth. Eng. **36** (1993) 1805-1823
7. Johnston, B.P., Sullivan Jr., J.M.: A Normal Offsetting Technique for Automatic Mesh Generation in Three Dimensions. Int. J. Num. Meth. Eng. **36** (1993) 1717-1734
8. Montenegro, R., Plaza, A., Ferragut, L., Asensio, I.: Application of a Nonlinear Evolution Model to Fire Propagation. Nonlinear Anal., Th., Meth. App. **30**(5) (1997) 2873-2882
9. Montero, G., Montenegro, R., Escobar, J.M.: A 3-D Diagnostic Model for Wind Field Adjustment. J. Wind Eng. Ind. Aer. **74-76** (1998) 249-261
10. Murphy, M., Mount, D.M., Gable, C.W.: A Point-Placement Strategy for Conforming Delaunay Tetrahedralization. In Symposium on Discrete Algorithms (2000) 67-74
11. Plaza, A., Montenegro, R., Ferragut, F.: An Improved Derefinement Algorithm of Nested Meshes. Adv. Eng. Soft. **27**(1/2) (1996) 51-57
12. Rivara, M.C.: A Grid Generator Based on 4-Triangles Conforming. Mesh-Refinement Algorithms. Int. J. Num. Meth. Eng. **24** (1987) 1343-1354