UNIVERSITY INSTITUTE OF INTELLIGENT SYSTEMS
AND NUMERICAL APPLICATIONS IN ENGINEERING
Report GANA/DDA-02-001
Prepared using iusianirep.cls *[Version: 2002/01/01 v1.0]*

An Efficient Package for 3-D Wind Simulation[†]

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

Division of Advanced Numerical Algebra - Division of Discretization and Applications
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, josem@sinf.ulpgc.es

CONTENTS

[†]Partially supported by MCYT and FEDER, Spain. Grant contract: REN2001-0925-C03-02/CLI

Copyright © 2002 U.L.P.G.C.

1. OUR RESEARCH ON WIND SIMULATION

In the University of Las Palmas de Gran Canaria (ULPGC), a group of persons have been working hard in wind simulation from 1987. The model have been improved during this last sixteen years. Actually, a research group of the Division of Advanced Numerical Algebra and the Division of Discretization and Applications, belonging to the University Institute of Intelligent Systems and Numerical Applications in Engineering (IUSIANI) of the ULPGC, have developed an efficient package for the three-dimensional wind field simulation. The final code include most recent techniques in numerical methods, such that, adaptive finite element method, tetrahedral mesh refinement, automatic mesh generation, parameter estimation using genetic algorithms, visualization of results using AVS, etc. In this section we present the group and its experience in these years. Specifically, we enumerate the main cientific results in the topic of wind simulation, in which members of our group have participated. The techniques and the wind field model have been recognized by the international cientific community, as it can be proved by the articles published in prestigious journals, conference communications and research projects sponsored by different institutions.

1.1. Researchers

The research about 3-D wind simulation, presented in this report, have been developed in the Division of Advanced Numerical Algebra and the Division of Discretization and Applications, belonging to the University Institute of Intelligent Systems and Numerical Applications in Engineering (IUSIANI) of the ULPGC. In particular, the group is composed of the following researchers:

- Rafael Montenegro Armas. Director of the Division of Discretization and Applications of IUSIANI. Professor of Applied Mathematics of the ULPGC. Doctor of Industrial Engineering.
- Gustavo Montero García. Director of the Division of Advanced Numerical Algebra of IUSIANI. Professor of Applied Mathematics of the ULPGC. Doctor of Industrial Engineering.
- José María Escobar Sánchez. Member of the Division of Discretization and Applications of IUSIANI. Associated Professor of Signal Theory. Licenciate of Physics and Doctor of Mathematics.
- Eduardo Rodríguez Barrera. Member of IUSIANI. Licenciate of Computer Science.
- José María González Yuste. Member of IUSIANI. Licenciate of Computer Science.

1.2. Research Projects

- Numerical Modelling of Wind Field Velocities in Wind Farms. Sponsored by *Consejería de Industria y Energía del Gobierno de Canarias*, Spain; (B.O.C.: 05.18.1987).
- Numerical Modelling of Wind Field Velocities in Wind Farms Using Adaptive Finite Element Methods and Applications. Sponsored by *Consejería de Industria y Energía del Gobierno de Canarias*, Spain; (B.O.C.: 01.06.1988).

Copyright © 2002 U.L.P.G.C.

- Wind Field Adjustment Using Mixed Finite Elements. Applications to Wind Farms. Sponsored by *Consejería de Industria y Energía del Gobierno de Canarias*, Spain; (B.O.C.: 10.27.1989).
- Construction of an Adaptive Model for the Evolution Air Pollution Problem. Sponsored by *Consejería de Industria y Energía del Gobierno de Canarias*, Spain; (B.O.C.: 10.15.1990).
- Numerical Simulation of Pollutions. Application to Power Stations. Sponsored by *Consejería de Industria y Energía del Gobierno de Canarias*, Spain; (B.O.C.: 10.15.1990).
- Construction of Wind Maps with 3-D Numerical Simulation for Wind Farms. Sponsored by *Consejería de Industria y Energía del Gobierno de Canarias*, Spain; (B.O.C.: 03.23.1992).
- Applications of Genetic Algorithms in the Optimisation of Fluid Dynamic Problems. Sponsored by *Dassault Aviation* (1994-1995).
- Numerical Modelling of Atmospheric Pollutant Transport. Sponsored by *Plan Nacional de I+D+I, Ministerio de Ciencia y Tecnología*, Spain, and FEDER; REN2001-0925-C03-02/CLI (2002-2004).

1.3. Journal Articles

- L. Ferragut, R. Montenegro, A. Plaza. Efficient Refinement/Derefinement Algorithm of Nested Meshes to solve Evolution Problems. *Communications in Numerical Methods in Engineering*; v. 10, pp. 403-412, John Wiley & Sons Ltd. (1994).
- G. Winter, G. Montero, L. Ferragut, R. Montenegro. Adaptive Strategies Using Standard and Mixed Finite Elements for Wind Field Adjustment. *Solar Energy*; v. 54, 1, pp. 49-56, Pergamon Press (1995).
- A. Plaza, R. Montenegro, L. Ferragut. An Inproved Derefinement Algorithm of Nested Meshes. *Advances in Engineering Software*, v. 27, 1/2, pp. 51-57, Elsevier Ltd. (1996).
- J.M. Escobar, R. Montenegro. Several Aspects of Three-Dimensional Delaunay Triangulation. *Advances in Engineering Software*, v. 27, 1/2, pp. 27-39, Elsevier Ltd. (1996).
- R. Montenegro, A. Plaza, L. Ferragut, I. Asensio. Application of a Nonlinear Evolution Model to Fire Propagation. *Nonlinear Analysis, Theory, Methods & Applications*, v. 30, 5, pp. 2873-2882, Elsevier Ltd. (1997).
- G. Montero. Solving Optimal Control Problems by Genetic Algorithms. *Nonlinear Analysis, Theory, Methods & Applications*, v. 30, 5, pp. 2891-2909, Elsevier Ltd. (1997).
- G. Montero, R. Montenegro, J.M. Escobar. A 3-D Diagnostic Model for Wind Field Adjustment. *Journal of Wind Engineering and Industrial Aerodynamics*, v. 74-76, pp. 249-261, Elsevier Ltd. (1998).
- R. Montenegro, J.M. Escobar, A. Plaza, L. Ferragut, G.F. Carey. Some Aspects of Unstructured Simplex Mesh Generation and Refinement. *Computational Methods and Neural Networks*, pp. 129-166, Dynamic Publ., Inc., Atlanta, USA (2000).

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

- G. Montero, N. Sanín. 3-D Modelling of Wind Field Adjustment Using Finite Differences in a Terrain Conformal Coordinate System. *Journal of Wind Engineering and Industrial Aerodynamics*, v. 89, pp. 471-488, Elsevier Ltd. (2001).
- R. Montenegro, G. Montero, J.M. Escobar, E. Rodríguez, J.M. González-Yuste. Tetrahedral Mesh Generation for Environmental Problems over Complex Terrains. To appear in *Lecture Notes in Computer Science*, Springer Verlag (2002).
- E. Rodríguez, G. Montero, R. Montenegro, J.M. Escobar, J.M. González-Yuste. Parameter Estimation in a Three-dimensional Wind Field Model Using Genetic Algorithms. To appear in *Lecture Notes in Computer Science*, Springer Verlag (2002).
- R. Montenegro, G. Montero, J.M. Escobar, E. Rodríguez. Efficient Strategies for Adaptive 3-D Mesh Generation over Complex Orography. To appear in *Neural, Parallel & Scientific Computations*, Dynamic Publishers (2002).

1.4. Conference Communications

- R. Montenegro, G. Montero, L. Ferragut, G. Winter. Wind Field Adjustment by Adaptive Finite Element Method. Proceedings of the SIAN-3. Madrid, Spain, May 22-24, 1990.
- L. Ferragut, R. Montenegro, G. Montero, G. Winter. Wind Field Adjustment by Mixed Finite Elements. Proceedings of the First Congress on Numerical Methods in Engineering, pp. 90-95. Las Palmas de Gran Canaria, Spain, June 11-15, 1990.
- L. Ferragut, G. Montero, G. Winter, R. Montenegro. Wind Field Adjustment: Resolution by Adaptive Mixed Finite Element and Multigrid Algorithm. Applications. Proceedings of European Community Wind Energy Conference and Exhibition, pp. 140-144. Sponsored by Commission of the European Communities. Madrid, Spain, September 10-14, 1990.
- A. Plaza, L. Ferragut, R. Montenegro. Derefinement Algorithm of Nested Meshes. Proceedings of the 12 th World Computer Congress "Information Processing 92 (IFIP'92)", Volume I: Algorithms, Software, Architecture, pp. 409-415, Elsevier Science Publishers B.V. (North-Holland). Madrid, Spain, September 7-11, 1992.
- A. Plaza, R. Montenegro, L. Ferragut. An Adaptive Refinement/Derefinement Algorithm of Structured Grids for Solving Time-Dependent Problems. Numerical Methods in Engineering '92, Proceedings of the First European Conference on Numerical Methods in Engineering, pp. 225-232, Elsevier Science Publishers B.V. Brussels, Belgium, September 7-11, 1992.
- R. Montenegro, L. Ferragut, A. Plaza. Applications of an Adaptive Finite Element Method. Invited lecture in Proceedings of the Fourth International Colloquium on Differential Equations, pp. 189-198, VSP International Science Publishers, The Netherlands, 1994. Plovdiv, Bulgaria, August 18-23, 1993.
- J.M. Escobar, R. Montenegro. Introduction to 3-D Mesh Generation. Proceedings of the XIII CEDYA/III CMA, pp. 181-186. Madrid, Spain, September 13-15, 1993.
- G. Montero, G. Winter, P. Cuesta. Modelling of Beach Sand Transport Phenomena by Wind Flow. Proceedings of the European Congress on Fluidization, pp. 187-193. Las Palmas de Gran Canaria, Spain, February 16-19, 1994.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

- J.M. Escobar, R. Montenegro. Construction of 3-D Meshes Using The Triangulation of Delaunay. Invited lecture in Proceedings of the Third International Colloquium on Numerical Analysis, pp. 51-60, VSP International Science Publishers (The Netherlands). Plovdiv, Bulgaria, August 13-17, 1994.
- A. Plaza, R. Montenegro, L. Ferragut. An Approach to Automatic Finite Element Mesh Generation by Managing Binary Images. Invited lecture in Proceedings of the Third International Colloquium on Numerical Analysis, pp. 135-142, VSP International Science Publishers (The Netherlands). Plovdiv, Bulgaria, August 13-17, 1994.
- G. Winter, G. Montero, P. Cuesta, M. Galán. Mesh Generation Using Genetic Algorithms. Proceedings of Advances in Structures Technology, pp. 225-231, Civil-Comp Ltd. Athens, Greece, August 1994.
- G. Winter, G. Montero, P. Cuesta, M. Galán. Mesh Generation and Adaptive Remeshing by Genetic Algorithms on Transonic Flow Simulation. Proceedings of Computational Fluid Dynamics'94, pp. 281-287, John Wiley & Sons Ltd. Stuttgart, Germany, September 1994.
- R. Montenegro, A. Plaza, J.M. Escobar, L. Ferragut. Aspects about Mesh Generation for Finite Element Method. Invited lecture in Proceedings de la First International Conference on Neural, Parallel and Scientific Computations, pp. 348-353, Dynamic Publishers Inc. Atlanta, U.S.A., May 28-31, 1995.
- G. Montero, R. Montenegro, J.M. Escobar. A 3-D Diadnostic Model for Wind Field Adjustment. Proceedings of the Second European & African Congress on Wind Engineering, 2EACWE, pp. 325-332. Genova, Italy, June 22-26, 1997.
- R. Montenegro, J.M. Escobar, G. Montero. A 3-D Model for Wind Fields. Proceedings of XV CEDYA/V CMA, v. 2, pp. 959-964. Vigo, Spain, September 22-26, 1997.
- N. Sanín, J.M. Santana, G. Montero. Construction of a 3-D Model for Wind Field Adjustment Using Finite Differences. Proceedings of XVI CEDYA/VI CMA. Las Palmas de Gran Canaria, Spain, September 21-24, 1999.
- N. Sanín, G. Montero. A 3-D High-order Accurate Time-stepping Scheme for Air Pollution Modelling. Proceedings of the MS'2000 International Conference on Modelling and Simulation. Las Palmas de Gran Canaria, September 25-27, 2000.
- J.M. Escobar, R. Montenegro, G. Montero, E. Rodríguez. Efficient 3-D Adaptive Mesh Generation for the Simulation of Problems Defined over Complex Orography. Part I: Basis. Proceedings of the XVII CEDYA/VII CMA, published in CD-ROM, 8 pages. Summary book, pp. 697-698. Salamanca, Spain, September 24-28, 2001.
- J.M. Escobar, R. Montenegro, G. Montero, E. Rodríguez. Efficient 3-D Adaptive Mesh Generation for the Simulation of Problems Defined over Complex Orography. Part II: Strategies and Applications. Proceedings of the XVII CEDYA/VII CMA, published in CD-ROM, 8 pages. Summary book, pp. 699-700. Salamanca, Spain, September 24-28, 2001.
- J.M. Escobar, J.M. González, R. Montenegro, G. Montero. A Local Refinement for Tetrahedral Meshes. Proceedings of the XVII CEDYA/VII CMA, published in CD-ROM, 8 pages. Summary book, pp. 701-702. Salamanca, Spain, September 24-28, 2001.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

- E. Rodríguez, G. Montero, R. Montenegro, J.M. Escobar. Parameter Estimation for a 3-D Wind Field Adjustment Model Using Genetic Algorithm. Proceedings of the XVII CEDYA/VII CMA, published in CD-ROM, 8 pages. Summary book, pp. 783-784. Salamanca, Spain, September 24-28, 2001.
- R. Montenegro, G. Montero, J.M. Escobar, E. Rodríguez, J.M. González-Yuste. Tetrahedral Mesh Generation for Environmental Problems over Complezx Terrains. To present in The 2002 International Conference on Computational Science (ICCS 2002). Amsterdam, The Netherlands, April 21-24, 2002.
- E. Rodríguez, G. Montero, R. Montenegro, J.M. Escobar, J.M. González-Yuste. Parameter Estimation in a Three-dimensional Wind Field Model Using Genetic Algorithms. To present in The 2002 International Conference on Computational Science (ICCS 2002). Amsterdam, The Netherlands, April 21-24, 2002.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

2. THE 3-D WIND SIMULATION PACKAGE

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.

We present a 3-D mass consistent model for wind field adjustment using the finite element method. This model generates a field of velocities for an incompressible fluid which adjusts to an initial one obtained from experimental measurements and physical considerations. The first step for constructing the initial wind field is to carry out a horizontal interpolation at the heigh of the measuremment stations over the terrain. From the values, vertical wind profiles are constructed taking into account the atmospheric stability, the roughness of the terrain, the geostrophic wind and the atmospheric stratification. Once the initial wind field is obtained, we formulate the problem of Incompressible Fluid Dynamic with *no-flow-through* boundary condition on the terrain. Next, a least square problem is formulated. The technique of Lagrange multipliers leads to an elliptic problem. Finally, we solve it using the finite element method which has proved to be an efficient tool for solving this kind of problems.

However, some regions of the studied domain may need more accuracy of the numerical solution due to the irregulatity of the terrain as well as strong variations of the solution. In order to improve the obtained solution we propose an adaptive refinement process of the three-dimensional mesh. First, we compute error indicators in each element of the mesh to be refined. This indicators will determine which elements must be refined. Our refinement technique, based on the 8-tetrahedral subdivision, allows a higher discretization of the affected regions without an excessive propagation in the mesh. The process may be repeated for any refined mesh until the error indicators satisfy a fixed tolerence.

Since mass consistent models are diagnostic models for constructing wind velocity fields from a few experimental measurements and, in general, these models are defined by the physical laws of an incompressible fluid, by the empirical design of the wind profiles and by the values of velocities measured at the stations, the existence of many parameters in the model is evident. Some of them are clearly bounded and defined, while others are still under discussion and interpretation. There are many methods for the resolution of inverse problems involving parameter estimation and they have been largely studied in the literature. Among them, we have chosen a robust and flexible tool: genetic algorithms, which allow to solve linear and non-linear multiparameter optimisation problems.

2.1. Efficient Mesh Generation over Complex Terrain

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

Copyright © 2002 U.L.P.G.C.

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 [32], fire propagation [30] and atmospheric pollution. These phenomena have the main effect on the proximities of the terrain surface. Thus node density increases in these areas accordingly.

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 [13, 37] 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. 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 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 [10]. To avoid conforming problems between mesh and orography, the tetrahedral mesh will be designed with the aid of an auxiliary parallelepiped. 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. Also, the details of the triangulation and mesh optimisation processes are presented.

2.1.1. 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 [41]. 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 [13, 37]. In this step we present the derefinement

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

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.

2.1.2. 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

$$z_i = a\, i^{\alpha} + b \ . \tag{2.1}$$

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 (2.1) becomes

$$z_i = \frac{h - z_0}{n^{\alpha}}\, i^{\alpha} + z_0 \quad ; \quad i = 0, 1, 2, ..., n \ . \tag{2.2}$$

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

$$z_i = z_{i-1} + \frac{h - z_{i-1}}{n^{\alpha} - (i-1)^{\alpha}}\, \left[i^{\alpha} - (i-1)^{\alpha} \right] \quad ; \quad i = 1, 2, ..., n \ . \tag{2.3}$$

In (2.2) or (2.3), 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 (2.2),

$$d = z_1 - z_0 = \frac{h - z_0}{n^{\alpha}} \ . \tag{2.4}$$

If we fix α and set free the value of n, from (2.4) we obtain

$$n = \left(\frac{h - z_0}{d} \right)^{1/\alpha} \ . \tag{2.5}$$

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

$$\alpha = \frac{\log \frac{h - z_0}{d}}{\log\, n} \ . \tag{2.6}$$

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

In both cases, given one of the parameters, the other may be calculated by expressions (2.5) or (2.6), 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 (2.6). For $i = n-1$, (2.2) could be expressed as

$$z_{n-1} = \frac{h - z_0}{n^\alpha} \, (n-1)^\alpha + z_0 \ . \tag{2.7}$$

and thus, by using (2.6),

$$\frac{\log\,(n-1)}{\log\,n} = \frac{\log \frac{h-z_0-D}{d}}{\log \frac{h-z_0}{d}} \ . \tag{2.8}$$

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, (2.8) yields

$$n = 1 + n^k \ . \tag{2.9}$$

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

$$2 \leq g(x) \leq 1 + \left(\frac{h - z_0}{d}\right)^k \leq \frac{h - z_0}{d} \ . \tag{2.10}$$

In view of the fixed point theorem, we can ensure that (2.9) 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.

2.1.3. 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

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

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 (2.2) 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 (2.2) 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 (2.9). Finally, the vertical spacing function is determined when obtaining the value of α by (2.6). 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.

2.1.4. 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* [35] and *constrained Delaunay triangulation* [17]. 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

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

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 (2.2). The triangulation of this set of points is done using a variant of Watson incremental algorithm [10] 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.

2.1.5. 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 [14], 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 [9]. 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

$$F(\mathbf{r}) = \sum_{e=1}^{s} f_e(\mathbf{r}) = \sum_{e=1}^{s} \frac{\sum_{i=1}^{6} (l_i^e)^2}{V_e^{2/3}} \ . \tag{2.11}$$

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

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

the minimisation of F originates positions of v for which the local mesh quality improves [9]. 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 (2.11) by the increasing function

$$h(V_e) = \frac{1}{2}(V_e + \sqrt{V_e^2 + 4\delta^2}) \ . \tag{2.12}$$

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

$$\Phi(\mathbf{r}) = \sum_{e=1}^{s} \phi_e(\mathbf{r}) = \sum_{e=1}^{s} \frac{\sum_{i=1}^{6} (l_i^e)^2}{[h(V_e)]^{2/3}} \ . \tag{2.13}$$

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.

2.1.6. 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 [41]. Once the data were interpolated on this refined mesh, we applied the derefinement algorithm developed in [13, 37] with a derefinement parameter of $\varepsilon = 40$ m. Thus, the adapted mesh approximates the terrain surface of Fig. 1 with an error less than that value. The node distribution of τ_1 is the one considered on the upper boundary of the domain.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

The result obtained is shown in Fig. 2, 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. 3, where $q(e)$ is the quality measure proposed in [14] 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 [37]. Besides, in experimental results we have approximately obtained a linear complexity in function of the number of points for our algorithm of 3-D Delaunay triangulation [10]. 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.

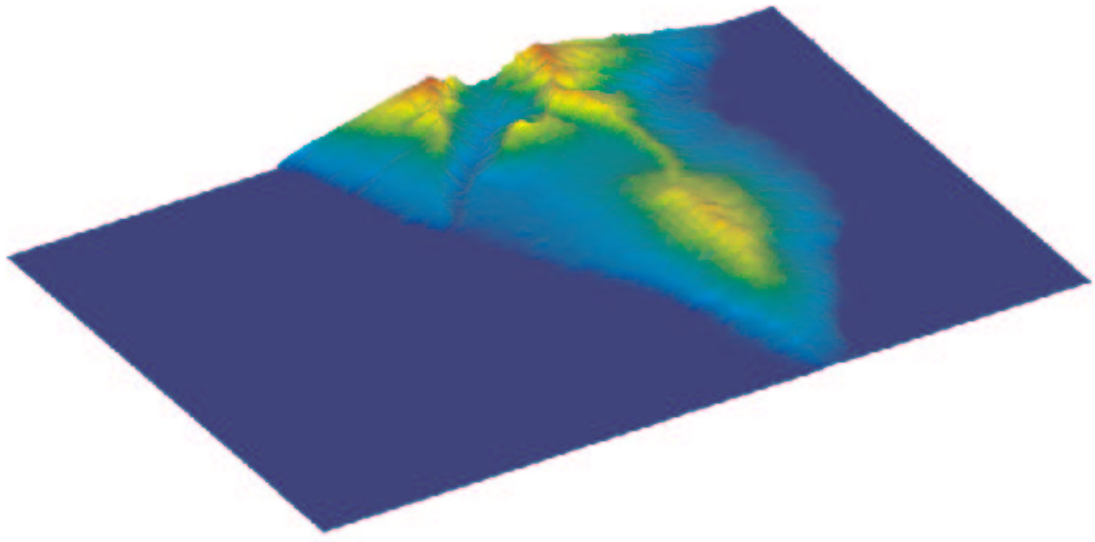


Figure 1. Terrain surface approximated by a refined/derefined mesh

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

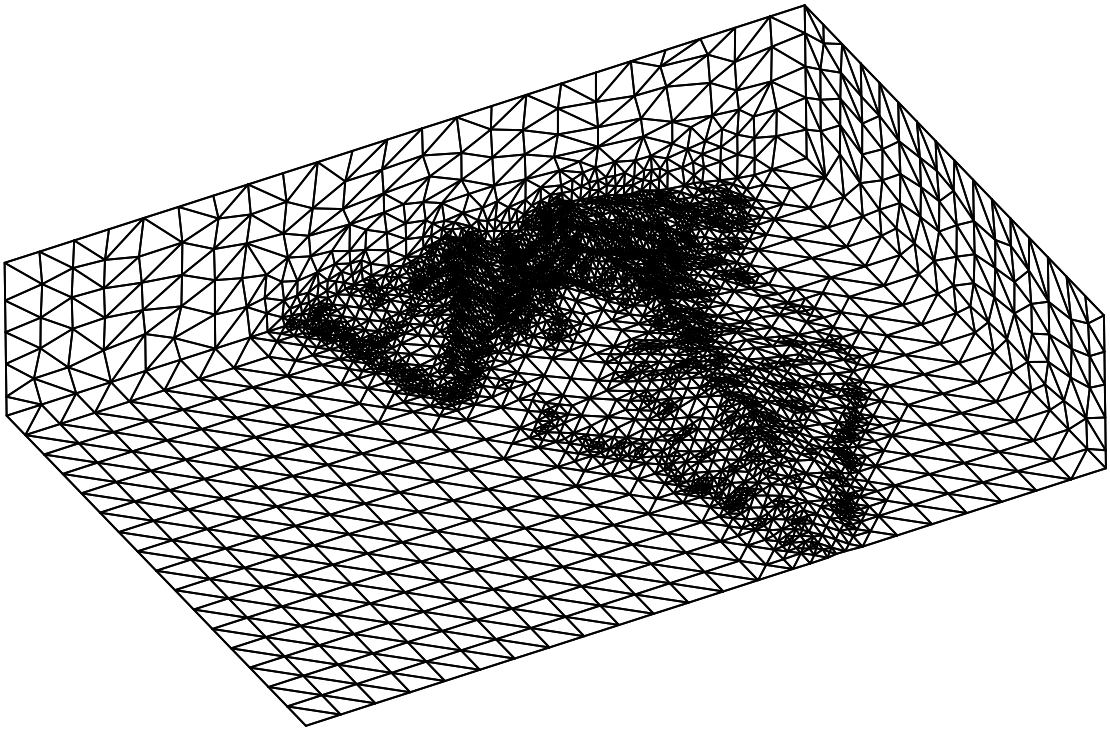


Figure 2. Resulting mesh after five steps of the optimisation process

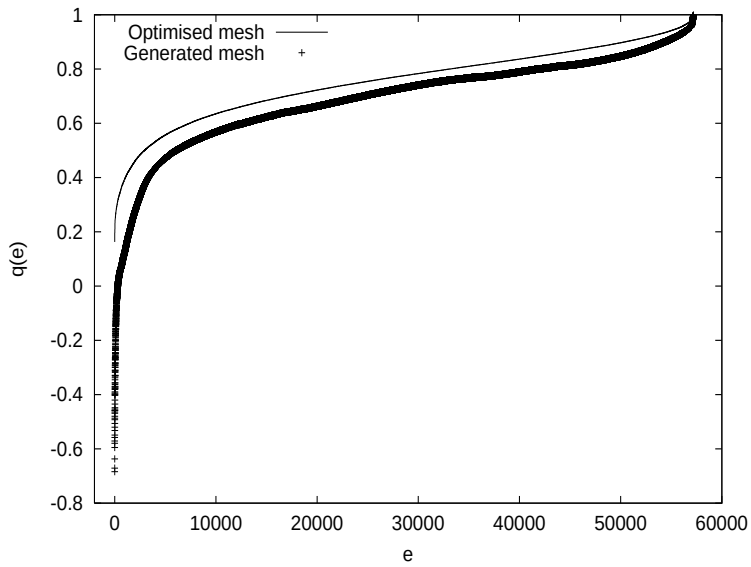


Figure 3. Quality curves of the generated mesh and the resulting mesh after five steps of the optimisation process. Function $q(e)$ is a quality measure for tetrahedron e

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

A second experiment has been carried out in order to show the behaviour of the mesh generator when the surface of the terrain has a hole under the sea level. We have consider only a half of the domain of $10000 \times 5000 \times 4000\ m^3$. In Fig. 4, the mesh on the surface and two lateral walls are detailed. The complete tetrahedral mesh is represented in Figs. 5 and 6 from two point of view.

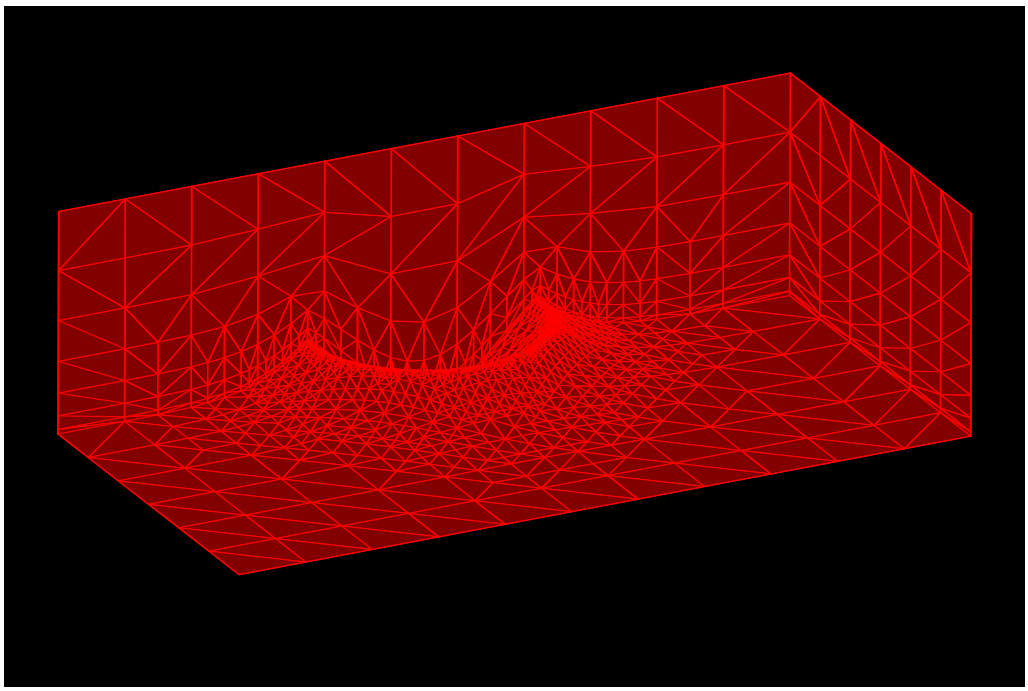


Figure 4. Mesh obtained for the second numerical experiment

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

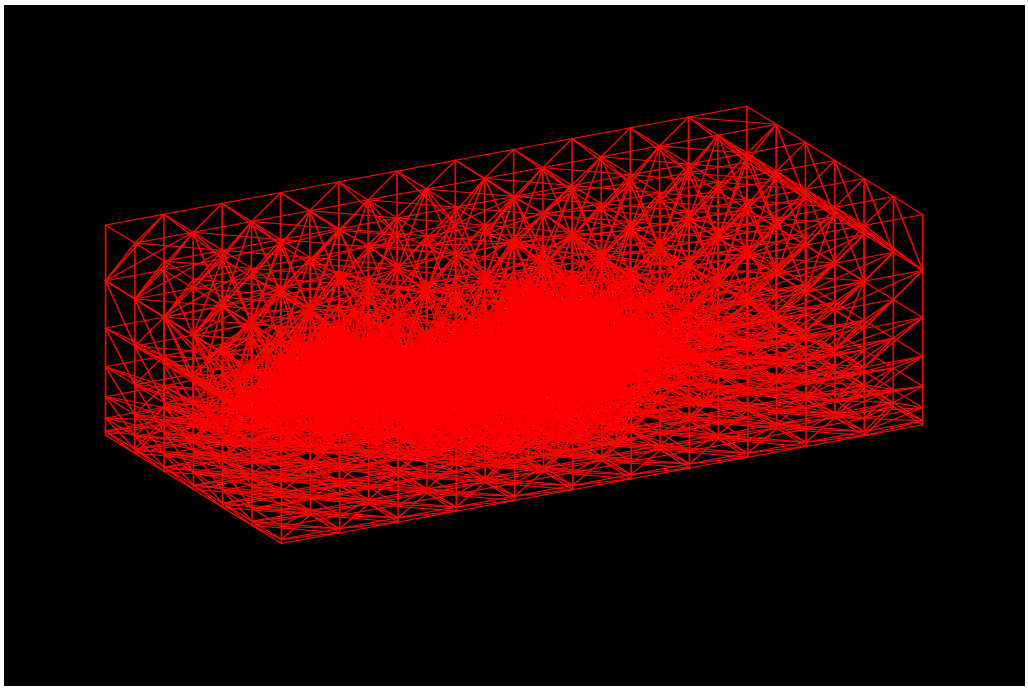


Figure 5. The complete image of the mesh. First point of view

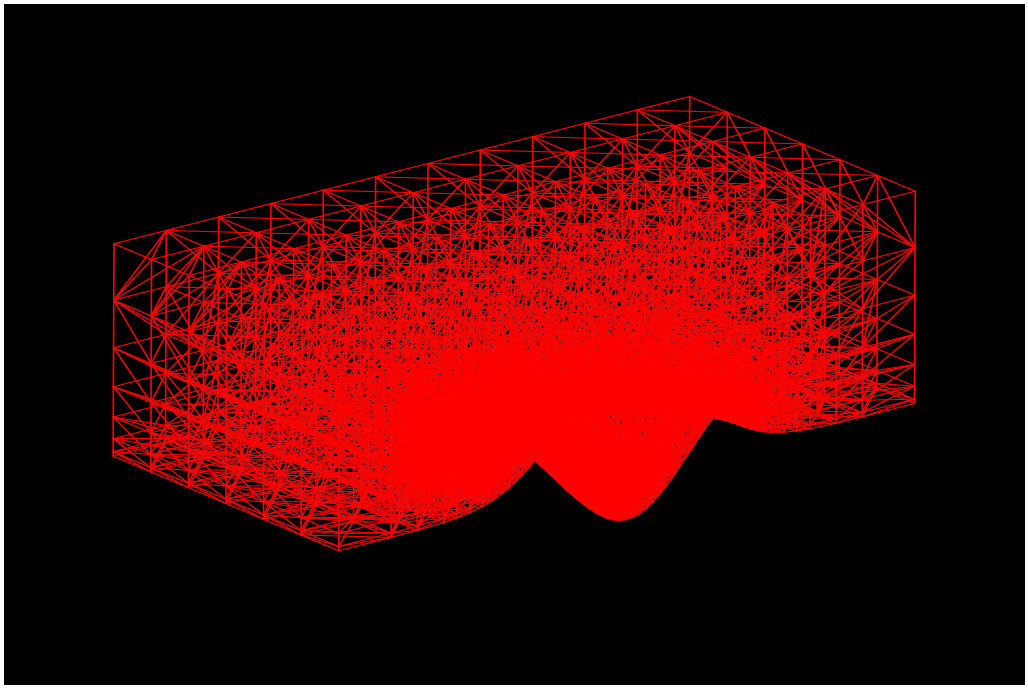


Figure 6. The complete image of the mesh. Second point of view

Copyright © 2002 U.L.P.G.C.
Prepared using iusianirep.cls

2.2. Wind Model

2.2.1. Mass Consistent Model in 3-D. This model [32] is based on the continuity equation for an incompressible flow with constant air density in the domain Ω and *no-flow-through* conditions on Γ_b

$$\vec{\nabla} \cdot \vec{u} = 0 \quad \text{in } \Omega \, . \tag{2.14}$$

$$\vec{n} \cdot \vec{u} = 0 \quad \text{on } \Gamma_b \, . \tag{2.15}$$

We formulate a least-square problem in Ω with $\vec{u}(\widetilde{u}, \widetilde{v}, \widetilde{w})$ to be adjusted

$$E(\vec{u}) = \int_\Omega \left[\alpha_1^2 \left((\widetilde{u} - u_0)^2 + (\widetilde{v} - v_0)^2 \right) + \alpha_2^2 \left(\widetilde{w} - w_0 \right)^2 \right] \, d\Omega \, . \tag{2.16}$$

where the interpolated wind $\vec{v}_0 = (u_0, v_0, w_0)$ is obtained from experimental measurements, and α_1, α_2 are the Gauss precision moduli. This problem is equivalent to find a saddle point $(\vec{v}, \phi)$ of the Lagrangian (see [50])

$$E\left(\vec{v}\right) = \min_{\vec{u} \in K} \left[E\left(\vec{u}\right) + \int_\Omega \phi \vec{\nabla} \cdot \vec{u} \, d\Omega \right] \, . \tag{2.17}$$

being $\vec{v} = (u, v, w)$, ϕ the Lagrange multiplier and K the set of admissible functions. The Lagrange multipliers technique is used to minimise the problem (2.17), whose minimum comes to form the Euler-Lagrange equations

$$u = u_0 + T_h \frac{\partial \phi}{\partial x}, \quad v = v_0 + T_h \frac{\partial \phi}{\partial y}, \quad w = w_0 + T_v \frac{\partial \phi}{\partial z} \, . \tag{2.18}$$

where $T = (T_h, T_h, T_v)$ is the diagonal transmissivity tensor, with $T_h = \frac{1}{2\alpha_1^2}$ and $T_v = \frac{1}{2\alpha_2^2}$. Since α_1 and α_2 are constant in Ω, the variational approach results in an elliptic problem substituting (2.18) in (2.14)

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{T_v}{T_h} \frac{\partial^2 \phi}{\partial z^2} = -\frac{1}{T_h} \left(\frac{\partial u_0}{\partial x} + \frac{\partial v_0}{\partial y} + \frac{\partial w_0}{\partial z} \right) \quad \text{in } \Omega \, . \tag{2.19}$$

The boundary conditions result as follows (Dirichlet condition for open or *flow-through* boundaries and Neumann condition for terrain and top)

$$\phi = 0 \quad \text{on } \Gamma_a \, . \tag{2.20}$$

$$\vec{n} \cdot T \, \vec{\nabla} \mu = -\vec{n} \cdot \vec{v}_0 \quad \text{on } \Gamma_b \, . \tag{2.21}$$

The classical formulation of the problem given by (2.19)-(2.21), is discretized using a tetrahedral mesh of finite elements (see [29]) that leads to a set of 4×4 elemental matrices and 4×1 elemental vectors, which are assembled into a global linear system of equations. A preconditioned conjugate gradient method is used for solving this symmetric linear system.

2.2.2. Horizontal Interpolation. The wind speeds measured at station height z_m are interpolated in function of the distance and the height difference between each point and

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

the station [32]

$$\vec{v}_0(z_m) = \varepsilon \frac{\sum_{n=1}^{N} \frac{\vec{v}_n}{d_n^2}}{\sum_{n=1}^{N} \frac{1}{d_n^2}} + (1-\varepsilon) \frac{\sum_{n=1}^{N} \frac{\vec{v}_n}{|\Delta h_n|}}{\sum_{n=1}^{N} \frac{1}{|\Delta h_n|}} \ . \tag{2.22}$$

where $\vec{v}_n$ is the velocity observed at station n, N is the number of stations considered in the interpolation, d_n is the horizontal distance from station n to the point where we are computing the wind velocity, $|\Delta h_n|$ is the height difference between station n and the studied point, and ε is a weighting parameter ($0 \leq \varepsilon \leq 1$), which allows to give more importance to one of these interpolation criteria.

2.2.3. Vertical Profile of Wind. We have considered a logarithmic profile in the surface layer, which takes into account the previous horizontal interpolation, as well as the effect of roughness and the air stability (neutral, stable or unstable atmosphere, according to the Pasquill stability class) on the wind intensity and direction. Above the surface layer, a linear interpolation is carried out using the geostrophic wind. The logarithmic profile is given by

$$\vec{v}_0(z) = \frac{\vec{v}^*}{k} \left(\log \frac{z}{z_0} - \Phi_m \right) \qquad z_0 < z \leq z_{sl} \ . \tag{2.23}$$

where $\vec{v}^*$ is the friction velocity, k is von Karman constant, z_0 is the roughness length and z_{sl} is the height of the surface layer. Function Φ_m depends on the air stability

$$\begin{aligned} \Phi_m &= 0 && \text{(neutral)} \ . \\ \Phi_m &= -5\frac{z}{L} && \text{(stable)} \ . \\ \Phi_m &= \log \left[\left(\frac{\theta^2+1}{2} \right) \left(\frac{\theta+1}{2} \right)^2 \right] - 2\arctan\theta + \frac{\pi}{2} && \text{(unstable)} \ . \end{aligned} \tag{2.24}$$

where $\theta = (1-16\frac{z}{L})^{1/4}$ and $\frac{1}{L} = az_0^b$, with a, b, depending on the Pasquill stability class. L is the so called Monin-Obukhov length. The friction velocity is obtained at each point from the interpolated measurements at the height of the stations (*horizontal interpolation*)

$$\vec{v}^* = \frac{k \ \vec{v}_0(z_m)}{\log \frac{z_m}{z_0} - \Phi_m} \ . \tag{2.25}$$

The height of the planetary boundary layer z_{pbl} above the ground is chosen such that the wind intensity and direction are constant at that height

$$z_{pbl} = \frac{\gamma \ |\vec{v}^*|}{f} \ . \tag{2.26}$$

where $f = 2\omega \sin\varphi$ is the Coriolis parameter (ω is the earth rotation and φ the latitude), and γ is a parameter depending on the atmospheric stability. The mixing height h is considered to be equal to z_{pbl} in neutral and unstable conditions. In stable conditions, Zilitinkevich suggested (see [5])

$$h = \gamma' \sqrt{\frac{|\vec{v}^*| \ L}{f}} \ . \tag{2.27}$$

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

where γ' is a constant of proportionality. The height of surface layer is $z_{sl} = \frac{h}{10}$. From z_{sl} to z_{pbl}, a linear interpolation with geostrophic wind $\vec{v}_g$ is carried out

$$\vec{v}_0(z) = \rho(z)\,\vec{v}_0(z_{sl}) + [1-\rho(z)]\vec{v}_g \qquad z_{sl} < z \le z_{pbl}\ . \tag{2.28}$$

$$\rho(z) = 1 - \left(\frac{z-z_{sl}}{z_{pbl}-z_{sl}}\right)^2 \left(3 - 2\frac{z-z_{sl}}{z_{pbl}-z_{sl}}\right)\ . \tag{2.29}$$

Finally, this model assumes $\vec{v}_0(z) = \vec{v}_g$ if $z > z_{pbl}$ and $\vec{v}_0(z) = 0$ if $z \le z_0$.

2.2.4. Finite element formulation. For the discretization by the finite element method of the problem given in (2.19),(2.20) and (2.21) we have used a tetrahedral mesh (see [11, 12] which leads to a set of elemental matrices of order 4×4,

$$\begin{aligned}\{\mathbf{A}^e\}_{ij} = \int_{\Omega^e} \{&(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial x})(\frac{\partial\hat{\psi}_j}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\hat{\psi}_j}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\hat{\psi}_j}{\partial\varphi}\frac{\partial\varphi}{\partial x})+\\ &+(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial y})(\frac{\partial\hat{\psi}_j}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\hat{\psi}_j}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\hat{\psi}_j}{\partial\varphi}\frac{\partial\varphi}{\partial y})+\\ &+\alpha^2(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial z})(\frac{\partial\hat{\psi}_j}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\hat{\psi}_j}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\hat{\psi}_j}{\partial\varphi}\frac{\partial\varphi}{\partial z})\}\cdot|\mathbf{J}|\ \ d\xi\,d\eta\,d\varphi\end{aligned} \tag{2.30}$$

and 4×1 elemental vectors,

$$\begin{aligned}\{\mathbf{b}^e\}_i = \int_{\Omega^e} -2\alpha_1^2\,\{&u_0(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial x} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial x})+\\ &+v_0(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial y} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial y})+\\ &+w_0\,(\frac{\partial\hat{\psi}_i}{\partial\xi}\frac{\partial\xi}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\eta}\frac{\partial\eta}{\partial z} + \frac{\partial\hat{\psi}_i}{\partial\varphi}\frac{\partial\varphi}{\partial z})\}\cdot|\mathbf{J}|\ \ d\xi\,d\eta\,d\varphi\end{aligned} \tag{2.31}$$

These matrices and vectors are assembled into a linear system of equations which is solved using a Preconditioned Conjugate Gradient algorithm.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

2.3. Adaptive Mesh Refinement

Nowadays, the most of the codes which use the finite element methods consider adaptive techniques. In the generation of adaptive meshes, the local refinement of the domain may be due to the geometry or to the numerical solution. It is carried out computing error estimators of the numerical solution or at least suitable error indicators for determining the elements to be refined or derefired in a mesh. Here we propose an error indicator which takes into account the gradient of the solution in each element.

At this point, one could choose between structured or non-structured meshes. Evidently, the use of non-structured meshes allows more flexibility for complex geometries using an optimal number of nodes. In this case, advanced front [27] and Delaunay triangulation [17, 10] are classical choices. Once the geometry of the domain is discretized, the mesh must be adapted taking into account the singularities of the numerical solution. This process implies the addition (refinement) or elimination (derefinement) of nodes in the current mesh. For this reason, a new mesh must be defined. The changes may affect the mesh locally or globally, depending on the selected triangulation method. Several strategies of refinement have been developed for 2-D triangulations and they have been generalised in 3-D. If we choose a refinement which locally affect the current mesh, we have to decide between nested or non-nested meshes. The choice is not so clear in this case. The use of nested meshes has several important advantages. We can obtain secuences of nested meshes in a minimal computing time. Moreover, the application of the multigrid method for solving the linear system of equations associated to the problem is straightforward. On the other hand, the smoothness and the degeneration of the mesh, and the preservation of surfaces defined on the domain are controllable automatically from the characteristics of the initial mesh. Thus, if the domain a complex geometry, a good choice is to obtain the initial mesh using a non-structured mesh generator and then, to apply a refinement/derefinement technique of nested meshes based on a suitable error indicator.

Taking into account these ideas, adaptive techniques in 2-D had been developed in the past which obtained good results in several steady and non-steady problems (see, i.e., [13, 37, 30, 50]). In these works, a version of Rivara 4-T local refinement algorithm [41] was used. In 3-D the problem is substantially different. Among the refinement algorithms developed in 3-D, we can consider those based on the bisection of tetrahedra [1, 42, 38] and those which used the 8-subtetrahedron subdivision [4, 25, 26]. In fact, the algorithm developed in [38] may be understood as the generalization in 3-D of the 4-T Rivara algorithm. This last one is also based on the bisection of the triangle by its bigger edge. The disadvantage of this method is the high number of possible cases in which a tetrahedron may be divided, considering the different possibilities of the 4-T subdivision on its four faces, during the process of mesh conformity. However, the algoritms proposed in [4, 25, 26], which genereralise the subdivision in 4 subtriangles of Bank et al [2] in 3-D, are simpler due to a lower number of possible subdivisions of a tetrahedron. These considerations have led us to implement a version of the 8-subtetrahedron subdivision algorithm.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

2.3.1. Refinement algorithm We propose a refinement algorithm based on the 8-subtetrahedron subdivision developed in [26]. Consider an initial triangulation τ_1 of the domain given by a set of n_1 tetrahedra t_1^1, t_2^1, ..., $t_{n_1}^1$. Our goal is to build a sequence of m leves 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_i^j associated to the element $t_i^j \in \tau_j$ which has been used is gradient type and it is defined as follows,

$$\epsilon_i^j = d_i^p \left| \vec{\nabla} \phi_h \right| \tag{2.32}$$

where the parameter α is generally assumed to be 1 or 2, and d_i, the length of the longest edge of tetrahedron t_i^j. Note that if $p = 1$, then ϵ_i^j represents the maximal variation of ϕ_h in the element t_i^j. Once the error indicator ϵ_i^j is computed, such element must be refined if $\epsilon_i^j \geq \theta \epsilon_{\max}^j$, being $\theta \in [0, 1]$ the refinement parameter and $\epsilon_{\max}^j$, 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 $t_i^1 \in \tau_1$ is called of *type I* if $\epsilon_i^1 \geq \gamma \epsilon_{\max}^1$. Later, this set of tetrahedra will be subdivided into 8 subtetrahedra as Figure 7(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 [2]. Thus, four subtetrahedra are determinated from the four vertices of t_i^1 and the new edges. The other four subtetrahedra are obtained by joining the two nearest opposite vertices of the octohedron which result inside t_i^1. This simple strategy is that proposed in [26] or in [4], in opposite to others based on afin transformations to a reference tetrahedron, as that analysed in [25] which ensures the quality of the resulting tetrahedra. However, similar results were obtained by Bornemann et al. [4] with both strategies in their numerical experiments. On the other hand, for Löhner and Baum [26], our 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 edgewithout 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.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

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, 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 7(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 in this case, 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 those with the vertices opposite to the two faces which contain each one. This tetrahedra are called *type III.a*; see Figure 7(c).

f.2. If the two marked edges are located on the same face, then 3 transient subtetrahedra are generated as Figure 7(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.*

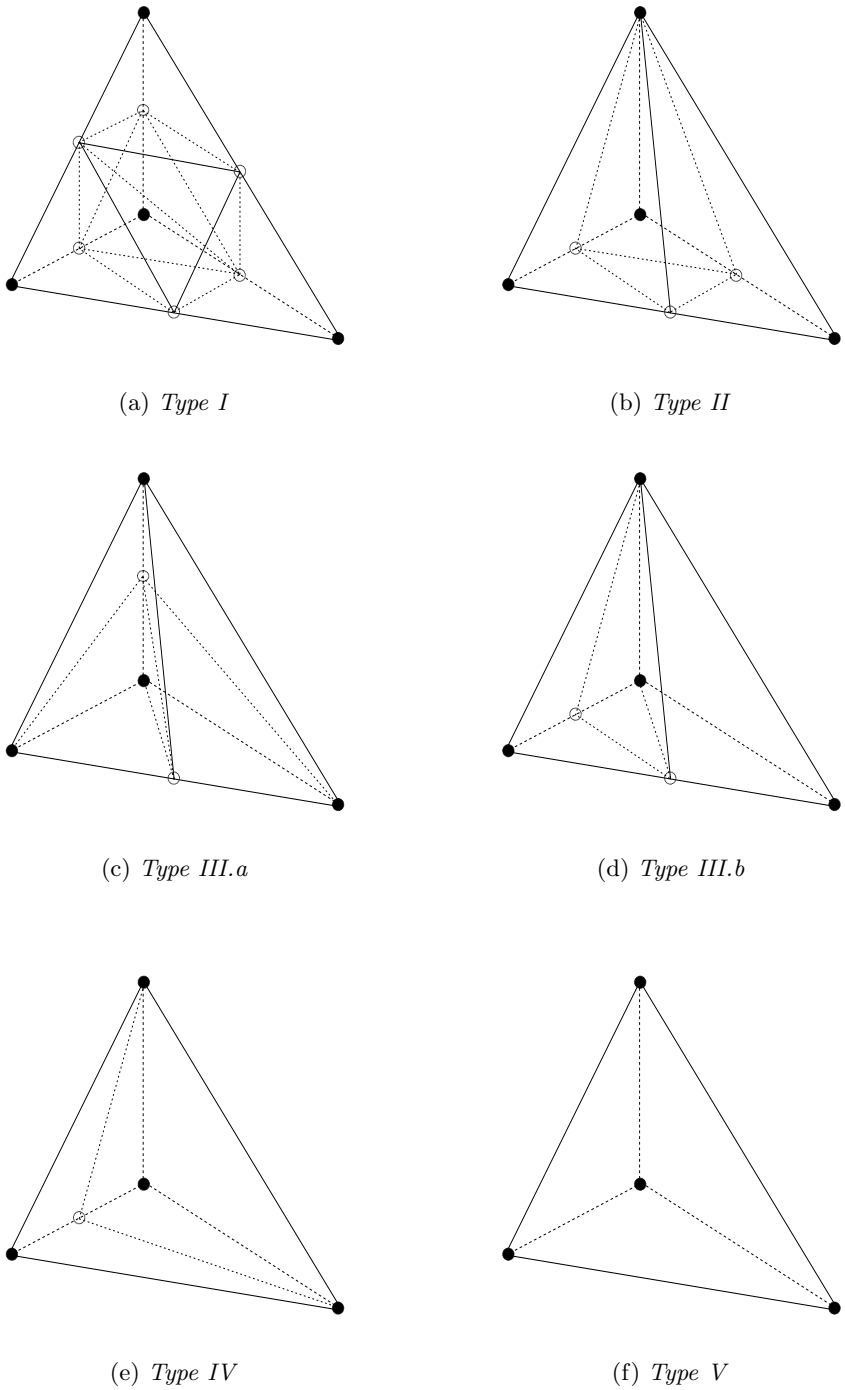
g) *Tetrahedra with 1 new node.* Two transient subtetrahedra will be created as we can see in Figure 1(e). The new node is connected to the other two which are not located in the marked edge. This set of tetrahedra is called *type IV.*

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 7(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 process a low computational cost, since the local expansion stops when we find tetrahedra whose edges have not to be marked.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls



(a) *Type I* (b) *Type II*

(c) *Type III.a* (d) *Type III.b*

(e) *Type IV* (f) *Type V*

Figure 7. Subdivision classification of a tetrahedron in function of the new nodes (white circles).

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

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 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.

2.3.2. Numerical Experiments The studied case corresponds to a test problem of wind field adjustment located in a square region of $10000 \times 10000\ m^2$, where the topography of the terrain is given by the function,

$$z = z_{max} \exp\left[-\left(\left(\frac{x - x_c}{s_x}\right)^2 + \left(\frac{y - y_c}{s_y}\right)^2\right)\right] \tag{2.33}$$

being in our example $z_{max} = 1500$, $x_c = y_c = 5000$, $s_x = 1000$ and $s_y = 800$. The location of the stations and the wind measures are shown in Table I. We have considered the height of 10 m over the terrain for the stations and null vertical component of wind velocities.

Stations m	x_m in m	y_m in m	u_m in m/s	v_m in m/s
1	0.0	0.0	0.0	5.0
2	5000.0	0.0	0.0	5.0
3	10000.0	0.0	0.0	5.0
4	0.0	5000.0	0.0	5.0
5	5000.0	5000.0	0.0	5.0
6	10000.0	5000.0	0.0	5.0
7	0.0	10000.0	0.0	5.0
8	5000.0	10000.0	0.0	5.0
9	10000.0	10000.0	0.0	5.0

Table I. Location of the stations and measures of wind velocities.

Other values of the parameters of the model in this numerical experiment are $\varepsilon = 0.5$, $\gamma = 0.3$, $\gamma' = 0.4$, $\alpha = 0.1$, $k = 0.4$, $\phi = 28.6^\circ$, $u_g = 0.0\ m/s$, $v_g = 10.0\ m/s$, as well as a softly stable atmosphere. Figure 8 represents the initial mesh obtained using the package developed in [12]. From this result, using the error indicator (2.32) with $p = 1$, two adaptive refinement steps were carried out with $\theta = 0.8$. The refined meshes obtained in this process are shown in Figures 9 y 10. We cam observe more concentration of nodes in region of the mountain where the wind experiments higher variations in speed or direction. Figures 11, 12 and 13 show the wind velocities obtained after the last refinement over horizontal planes at 10 m (height of the 8 station at the bottom), at 500 m and at 1500 m (heigh of the top of the mountain). As we could foresee for the value of α considered here, the wind tends to border the mountain horizontally.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

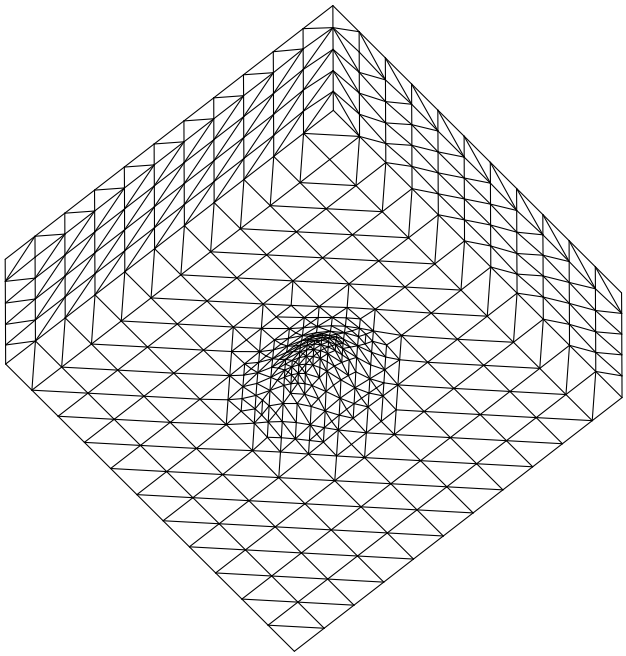


Figure 8. Initial mesh.

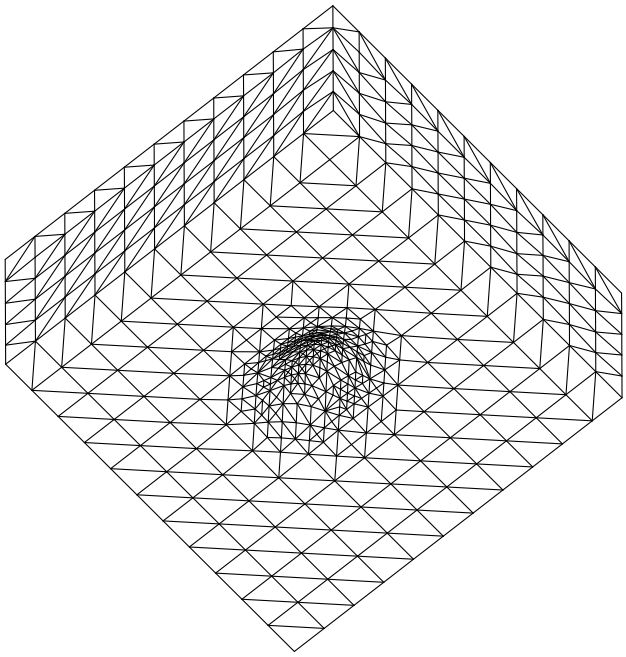


Figure 9. Mesh corresponding to the first refinement step.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

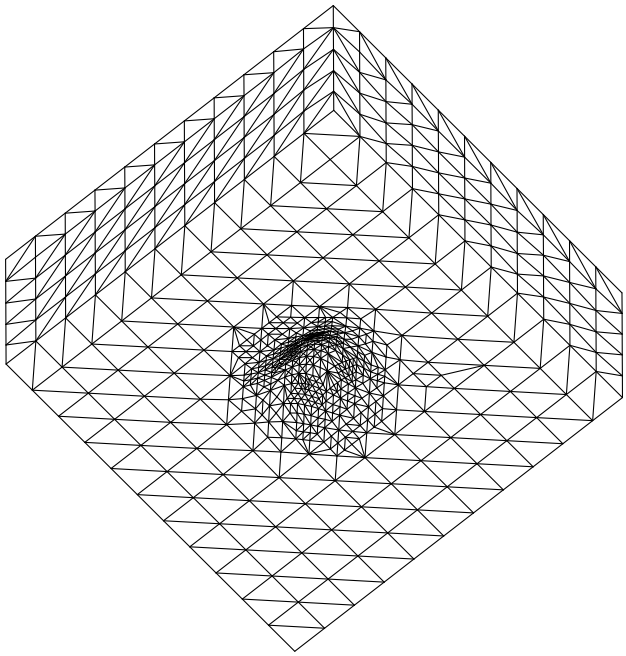


Figure 10. Mesh corresponding to the second refinement step.

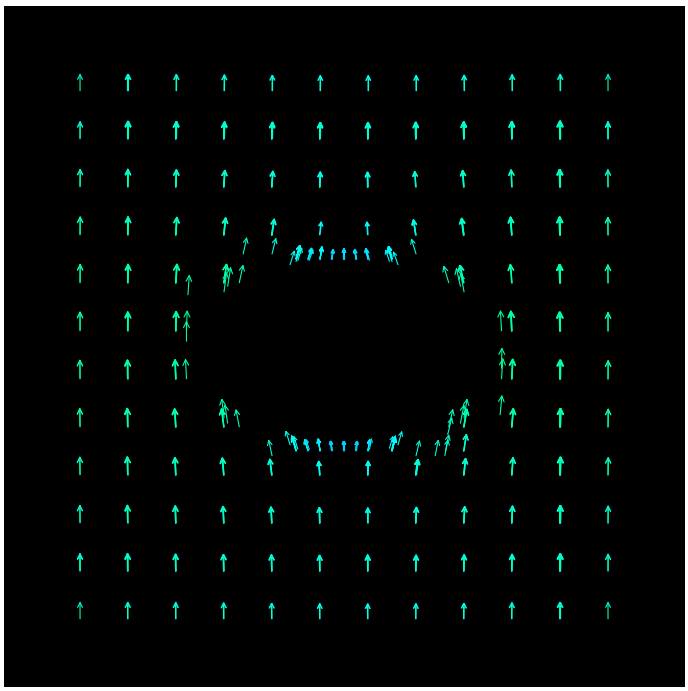


Figure 11. Wind velocities at 10 m.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

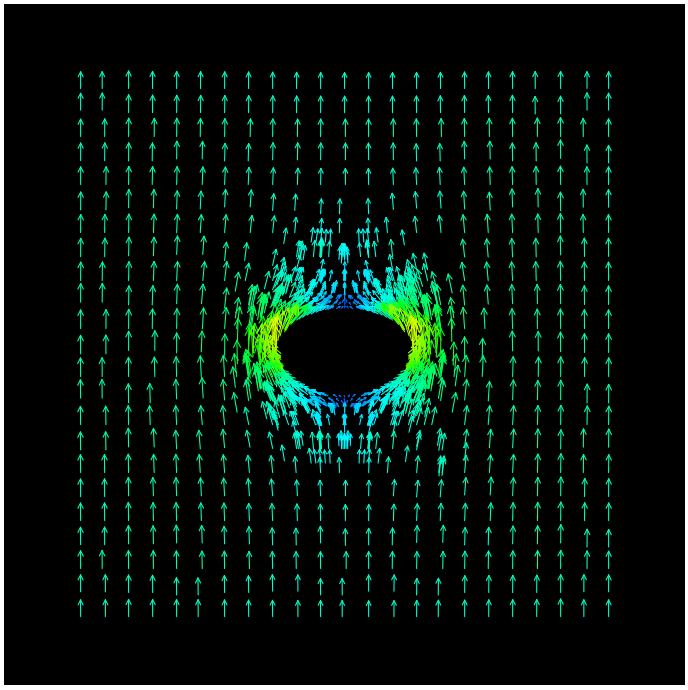


Figure 12. Wind velocities at 500 m.

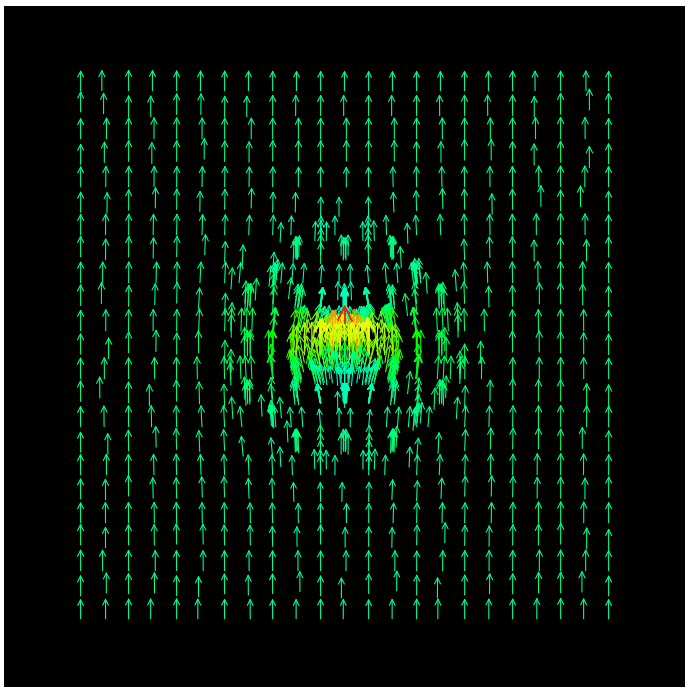


Figure 13. Wind velocities at 1500 m.

Copyright © 2002 U.L.P.G.C.
Prepared using iusianirep.cls

2.4. Parameter Estimation using Genetic Algorithms

The efficiency of a mass consistent model for wind field adjustment depends on several parameters that arise in various stages of the process. First, those involved in the construction of the initial wind field using horizontal interpolation and vertical extrapolation of the wind measures registered at meteorological stations. On the other hand, the Gauss precision moduli which allow from a strictly horizontal wind adjustment to a pure vertical one. In general, the values of all of these parameters are based on empirical laws. The main goal of this work is the estimation of these parameters using genetic algorithms, such that the wind velocities observed at the measurement station are regenerated as much as possible by the model.

2.4.1. Discussion on the Parameters to Be Estimated. First, we will consider the so called stability parameter

$$\alpha = \frac{\alpha_1}{\alpha_2} = \sqrt{\frac{T_v}{T_h}} \ . \tag{2.34}$$

since the minimum of the functional given by (2.16) is the same if we divide it by α_2^2. On the other hand, for $\alpha >> 1$ flow adjustment in the vertical direction predominates, while for $\alpha << 1$ flow adjustment occurs primarily in the horizontal plane. Thus, the selection of α allows the air to go over a terrain barrier or around it, respectively [39]. Moreover, the behaviour of mass consistent models in many numerical experiments has shown that they are very sensitive to the values chosen for α. Therefore, we shall give particular attention to this problem. In the past, many authors have studied the parametrisation of stability, since the difficulty in determining the correct values of α have limited the possible wide use of mass-consistent models in complex terrain. Sherman [44], Kitada et al. [21] and Davis et al. [6], proposed to take $\alpha = 10^{-2}$, i.e., proportional to the magnitude of w/u. Other authors such as Ross et al. [43] and Moussiopoulos et al. [34] related α to the Froude number. Geai [16], Lalas et al. [23] and Tombrou et al. [47], make the α parameter vary in the vertical direction. Finally, Barnard et al. [3] proposed a procedure to obtain α for each single wind field simulation. The main idea is to use N observed wind speeds to obtain the wind field and to keep the rest, N_r, as a reference. Then, several simulations are performed with different values of α. The value which gives the best agreement with the reference observations is taken to be the final magnitude of the stability parameter. Since this method provides values of α that are only reliable for each particular case, it cannot provide an *a priori* value suitable for other simulations. Here, we follow a version of the method proposed in [3], using genetic algorithms as optimisation technique which lead to an automatic selection of α.

The second parameter to be estimate is the weighting coefficient ε ($0 \leq \varepsilon \leq 1$) of (2.22). Note that $\varepsilon \to 1$ signifies more importance of the *horizontal distance* from each point to the measurement stations, while $\varepsilon \to 0$ signifies more importance of the *height difference* between each point and the measurement stations [32]. In general, the second approach has been used for complex terrains. On the other hand, the first approach has been widely used for problems with regular topography or in 2-D horizontal analysis. In realistic applications, the possibility of existing zones with complex orography and others with regular one, suggests that an intermediate value of ε should be more suitable.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

The next parameter to discuss is γ, given in (2.26) and related to the height of the planetary boundary layer. There exist different versions of where to search this parameter. Panofsky et al. [36] proposed the interval [0.15,0.25]. On the other hand, Ratto [40] directly suggested $\gamma = 0.3$ in the *WINDS* code, while γ is located in [0.3,0.4] by de Baas [8]. Therefore, in our simulations, the search space for γ must include all these possibilities.

Finally, we are interested in obtaining suitable values of the parameter γ' involved in the computation of the mixing height for stable atmosphere, see (2.27). Garratt [15] proposed $\gamma' = 0.4$. Also in the *WINDS* code one may find bounds of γ' around 0.4. Thus, the value of γ' will be searched in the surroundings of 0.4.

2.4.2. Genetic Algorithms. Genetic algorithms (GAs) are optimisation tools based on the natural evolution mechanism. They produce successive trials that have an increasing probability to obtain a global optimum. This work is based on the model developed by Levine [24]. The most important aspects of GAs are the construction of an initial population, the evaluation of each individual in the fitness function, the selection of the parents of the next generation, the crossover of those parents to create the children, and the mutation to increase diversity.

Two population replacements are commonly used. The first, the generational replacement, replaces the entire population each generation [18]. The second, known as steady-state, only replaces a few individuals each generation [46, 48, 49]. Stopping criteria are iteration limit exceeded, population too similar, and no change in the best solution found in a given number of iterations. Initial population is randomly generated.

The selection phase allocates an intermediate population on the basis of the evaluation of the fitness function. We have considered four selection schemes [24]: proportional selection (P), stochastic universal selection (SU), binary tournament selection (BT) and probabilistic binary tournament selection (PBT).

The crossover operator takes bits from each parent and combines them to create a child. One-point (OP) and uniform (U) crossover operators are used here. The first one selects randomly the place where each of the parents strings are broken in two substrings. Children will be the union of first substring of one parent and the second of the other. Uniform crossover depends on the probability of exchange between two bits of the parents [45].

The mutation operator is better used after crossover [7]. It allows to reach individuals on the search space that could not be evaluated otherwise. When part of a chromosome has been randomly selected to be mutated, the corresponding genes belonging to that part are changed. This happens with probability p. This work deals with four mutation operators. Three of them are of the form $\nu \leftarrow \nu \pm p \times \nu$, where ν is the existing allele value, and p may be a constant value (C), chosen uniformly from the interval $(0, \beta)$ with $\beta \leq 1$ (U), or selected from a Gaussian distribution (G). The fourth operator (R) simply replaces ν with a value selected uniformly random from the initialisation range of that gene.

The fitness function plays the role of the environment. It evaluates each string of a population. This is a measure, relative to the rest of the population, of how well that string satisfies a problem-specific metric. The values are mapped to a nonnegative and monotonically increasing fitness value. In the numerical experiments with this wind model,

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

we look for suitable values of α, ε, γ and γ'. For this purpose, the average relative error of the wind velocities given by the model with respect to the measures at the reference stations is minimised

$$F(\alpha, \varepsilon, \gamma, \gamma') = \frac{\displaystyle\sum_{n=1}^{N_r} \frac{|\vec{v}_n - \vec{v}(x_n, y_n, z_n)|}{|\vec{v}_n|}}{N_r} \ . \tag{2.35}$$

where $\vec{v}(x_n, y_n, z_n)$ is the wind velocity obtained by the model at the location of station n, and N_r is the number of reference stations.

2.4.3. Numerical Experiments. We consider the same wind field problem related to the southern area of La Palma Island (Canary Islands) which was defined in [33]. A $45600 \times 31200 \times 9000\ m^3$ domain with real data of the topography is discretized using the code developed in [29]. The maximum height in this zone of the island is 2279 m. The mesh contains 11578 nodes and 52945 tetrahedra; see one of the initial meshes in Fig. 14 and the used mesh in Fig. 15. The wind measurements were taken in four stations: MBI, MBII, MBIII and LPA. From the three cases studied in [33], we have selected case I with softly unstable conditions and case III with softly stable conditions in order to test the procedure for different stability conditions of the atmosphere. Due to the small number of available data, we have used the observed wind speeds of stations MBI, MBII and LPA to obtain the interpolated wind field (2.22), i.e., $N = 3$, and the measurement of MBIII is considered as reference station in the fitness function (2.35), i.e., $N_r = 1$.

In the first application (case I), the parameter γ' is not involved in the modelling due to the unstable condition of the atmosphere, i.e., $h = z_{pbl}$. Thus, only α, ε and γ will be estimated in this case. The experiment has been divided in two stages. First, we fix $\gamma = 0.3$ and estimate $\alpha \in [10^{-3}, 10]$ and $\varepsilon \in [0, 1]$. The second column of Tab. 1 (*Stage 1*) shows the values obtained for α and ε, which suggest a nearly vertical wind adjustment and remark the complexity of the terrain respectively. Note that we obtain with the model an error at station MBIII about 10.7%. The strategy of GAs (*BT, U, R*) corresponds to the most efficient selection, crossover and mutation operators after several tests with different combinations. In the second stage, α, ε and $\gamma \in [0.15, 0.5]$ are estimated. The results are also shown in the third column of Tab. 1. We observe that α is near the maximum value of the space of search, ε remains quite small and γ is reduced, such that the error at station MBIII is 10.7%. We remark that in this experiment the worst evaluation of the fitness function, corresponding to values of the parameters in the search space, yields an error of 72.19%. Therefore, the knowledge of the studied parameters is essential for the efficiency of the numerical model.

For the second experiment (case III) we have followed a similar procedure. Now, $\gamma' \in [0.15, 0.5]$ must be also considered. First, a problem with two unknown parameters (α, ε) is solved. The second column of Tab. 2 (*Stage 1*) shows the values obtained for α, ε. Next, four problems arising from fixing one of the parameters each time, respectively, are studied (*Stages 2-5*). Finally, the four parameters are estimated at the same time in *Stage 6*. The atmospheric stable conditions reduce the vertical adjustment predominance arising in the previous experiment with unstable conditions, as well as augment the importance of the horizontal distance in the interpolation of the observed wind speeds. The minimum

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

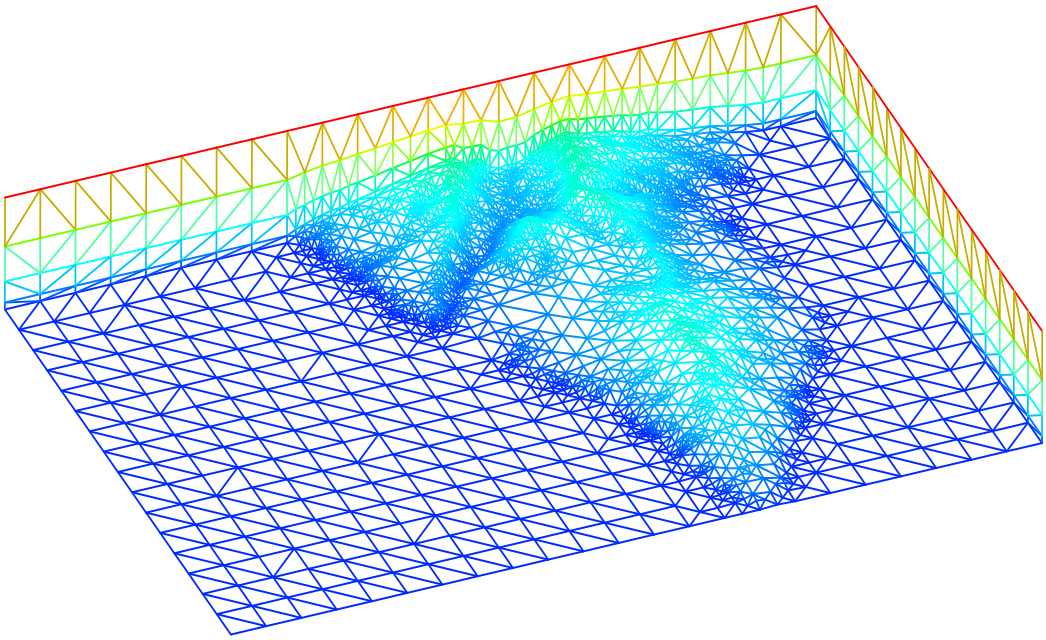


Figure 14. Detail of one of the initial finite element mesh constructed for the numerical experiment. We only plot the triangulation of the terrain and two vertical wall in order to hold clarity

error obtained at station MBIII was about 22.2%, while the error related to the worst evaluation was 118.04%.

In both experiments, the number of individuals of the initial population was 100, except for stage 6 in case III where it was 150. Iterations and CPU timings on a 933 MHz Pentium III are shown in Tab. 1 and Tab. 2 for each stage. Evidently, the computational cost would be considerably reduced using a massive parallel machine, where the GAs become competitive with other optimisation methods.

Finally, as example, Fig. 2 shows the wind field obtained by the model, in the second experiment, at a height of 200 m using the values of the parameters corresponding to *Stage 6*. Here, the measures of the four stations have been taken into account for determining the interpolated wind field.

Copyright © 2002 U.L.P.G.C.
Prepared using iusianirep.cls

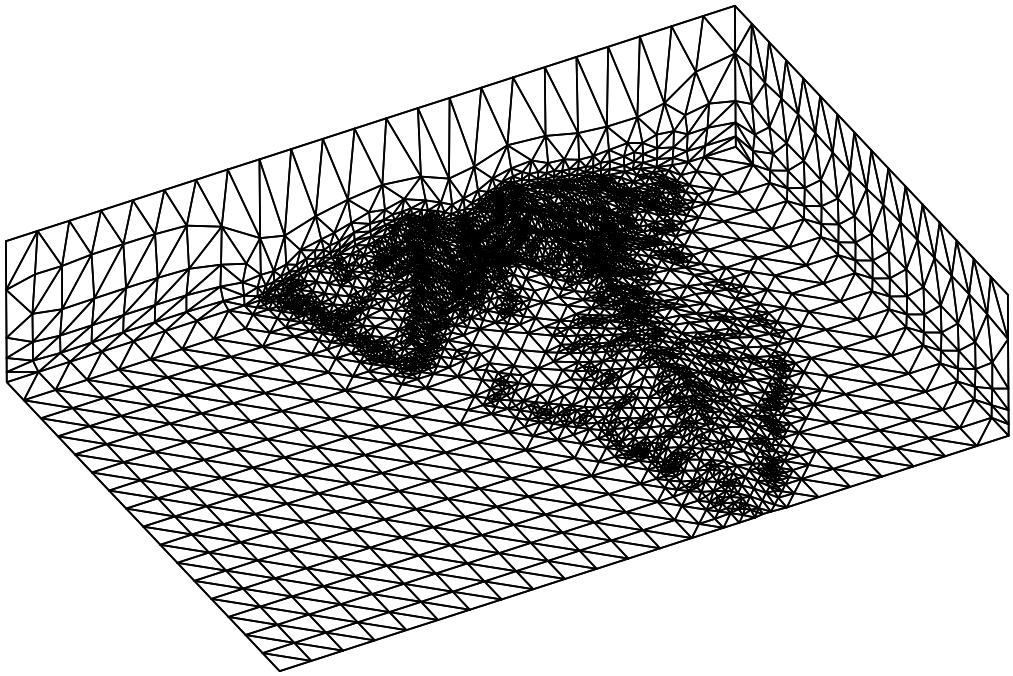


Figure 15. Detail of the optimised finite element mesh used for the numerical experiment

Table II. First experiment corresponding to the case I analysed in [33]. Strategy of genetics algorithms, best evaluation of the fitness function and values of the parameters (*fixed values are written between brackets*)

	Stage 1	Stage 2
GAs strategy	BT, U, R	SU, U, G
Iterations	17	1
CPU time (s)	1548	108
Best Fitness	0.107	0.107
α	9.810	9.727
ε	0.010	0.029
γ	(0.300)	0.284

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

Table III. Second experiment corresponding to the case III analysed in [33]. Strategy of genetics algorithms, best evaluation of the fitness function and values of the parameters (*fixed values are written between brackets*)

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Stage 6
GAs strategy	SU, U, G	SU, U, R	SU, U, R	SU, U, R	SU, U, R	SU, U, R
Iterations	110	147	37	16	29	61
CPU time (s)	8514	12120	2958	1362	2406	7050
Best Fitness	0.234	0.227	0.222	0.223	0.222	0.222
α	4.182	5.041	(5.041)	4.765	5.699	6.080
ε	0.003	0.272	0.281	(0.281)	0.292	0.282
γ	(0.300)	0.490	0.493	0.498	(0.498)	0.494
γ'	(0.400)	(0.400)	0.154	0.153	0.154	0.153

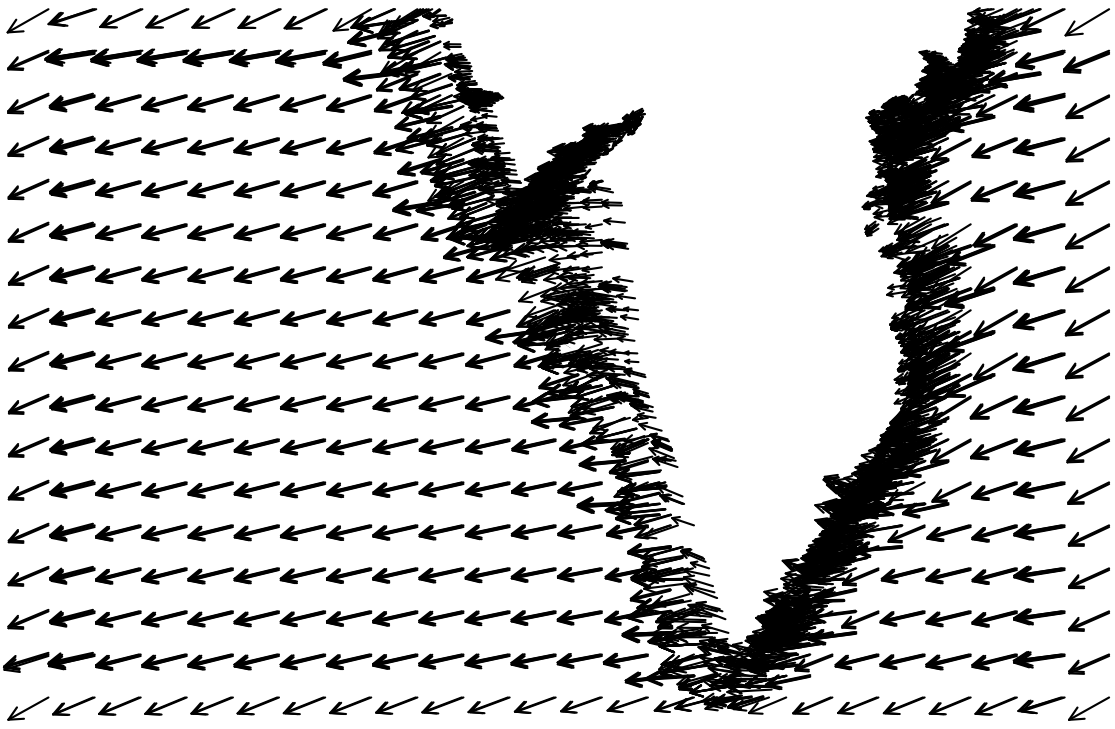


Figure 16. Wind field solution related to the second experiment at a height of 200 m

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

3. CONCLUSIONS AND FUTURE RESEARCH

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 [19] and normal offsetting [20]. Finally, the procedure here proposed for optimising the generated mesh allows us to solve the tangling problems and mesh quality at the same time.

In adittion, this work represents a study of an adaptive model for wind field adjustment. We have applied the techniques of tetrahedral mesh refinement proposd by the authors in [11] to the wind model developed in [32, 33]. The results allow us to conclude that, even starting from a mesh adapted to the terrain topography, the variations of the solution make necessary the use of techniques of adaptive mesh refinement. In this problem, we remark the good behaviour of the gradient-like indicator used in this work, which has allowed to detect those regions near the terrain where there are strong variations of the speed and direction of the wind, efficiently. Evidently, if we look for a secuence of wind field solutions adjusted to experimental data at different times, we should think of adding a mesh derefinement code to the model, what would allow us to work with quasi-optimal meshes at each time step.

The estimation of several parameters is essential for the efficiency of a 3-D mass consistent model for wind field adjustment. The numerical experiments have shown that these codes are very sensitive to the values chosen for α, ε, γ and γ'. A methodology for solving these parameter estimation problems is proposed. Genetic algorithms have proved to be an efficient and robust tool for these optimisation problems when several parameters are involved (see also [31]).

Further research is necessary in order to adapt our model for offshore wind simulation. On one hand, it is essential to introduce in the model the contribution of thermal effects which not only can deviate wind speed profiles from our log-linear one but produce sea breezes and low level jets. On the other hand, the modelling of wake effects within large offshore wind farms should be studied and added to our wind model.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

REFERENCES

1. Arnold, D.N., Mukherjee, A. y Pouly, L., Locally adapted tetrahedral meshes using bisection. *SIAM J. Sci. Comput.* 2000; **22**(2):431-448.
2. Bank, R.E., Sherman, A.H., Weiser, A., Refinement algorithms and data structures for regular local mesh refinement. In *Scientific Computing IMACS*. North-Holland:Amsterdam, 1983; 3-17.
3. Barnard, J.C., Wegley, H.L., Hiester, T.R., Improving the Performance of Mass Consistent Numerical Model Using Optimization Techniques. *J. Climate Appl. Meteorol* 1987; **26**:675-686.
4. Bornemann, F., Erdmann, B., Kornhuber, R., Adaptive multilevel methods in three space dimensions. *Int. J. Numer. Meth. Eng.* 1993; **36**:3187-3203.
5. Businger, J.A., Arya, S.P.S., Heights of the Mixed Layer in the Stable, Stratified Planetary Boundary Layer. *Adv. Geophys.* 1974; **18A**:73-92.
6. Davis, C.G., Bunker, S.S., Mutschlecner, J.P., Atmospheric Transport Models for Complex Terrain. *J. Climate Appl. Meteorol.* 1984; **23**:235-238.
7. Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.
8. de Baas, A.F., Scaling Parameters and their Estimation. In *Modelling of Atmospheric Flow Fields*, Lalas, D.P., Ratto, C.F. (eds.) World Sci.:Singapore, 1996; 87-102.
9. 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
10. Escobar, J.M., Montenegro, R., Several Aspects of Three-Dimensional Delaunay Triangulation. *Adv. Eng. Soft.* 1996; **27**(1/2):27-39.
11. Escobar, J.M., González, J.M., Montenegro, R., Montero, G., Un refinamiento local de mallas de tetraedros. In *XVII CEDYA / VII*, Salamanca, 2001. Published in CD-ROM.
12. Escobar, J.M., Montenegro, R., Montero, G., Rodríguez, E., Generación eficiente de mallas tridimensionales adaptadas para la simulación de problemas definidos sobre orografía irregular. Parte I: Fundamentos and Parte II: Estrategias y Aplicaciones. In *XVII CEDYA / VII*, Salamanca, 2001. Published in CD-ROM.
13. Ferragut, L., Montenegro, R., Plaza, A., Efficient Refinement/Derefinement Algorithm of Nested Meshes to Solve Evolution Problems. *Comm. Num. Meth. Eng.* 1994; **10**:403-412.
14. 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.
15. Garratt, J.R., Observations in the Nocturnal Boundary Layer. *Boundary-Layer Meteorol.* 1982; **22**(1):21-48.
16. Geai, P., Methode d'Interpolation et de Reconstitution Tridimensionelle d'un Champ de Vent: le Code d'Analyse Objective MINERVE. *Technical Report DER/HE/34-87.03*, EDF, Chatou, France, 1985.
17. George, P.L., Hecht, F., Saltel, E., Automatic Mesh Generation with Specified Boundary. *Comp. Meth. Appl. Mech. Eng.* 1991; **92**:269-288.
18. Holland, J., *Adaption in Natural and Artificial Systems*, MIT Press, 1992.
19. Jin, H., Tanner, R.I., Generation of Unstructured Tetrahedral Meshes by Advancing Front Technique. *Int. J. Num. Meth. Eng.* 1993; **36**:1805-1823.
20. Johnston, B.P., Sullivan Jr., J.M., A Normal Offsetting Technique for Automatic Mesh Generation in Three Dimensions. *Int. J. Num. Meth. Eng.* 1993; **36**:1717-1734.
21. Kitada, T., Kaki, A., Ueda, H., Peters, L.K., Estimation of Vertical Air Motion from Limited Horizontal Wind Data - A Numerical Experiment. *Atmos. Environ.* 1983; **17**:2181-2192.
22. Lalas, D.P., Ratto, C.F., *Modelling of atmospheric flow fields*, World Scientific Publishing:Singapore, 1996.
23. Lalas, D.P., Tombrou, M., Petrakis, M., Comparison of the Performance of Some Numerical

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

Wind Energy Siting Codes in Rough Terrain. In *European Community Wind Energy Conference*, Herning, Denmark, 1988.
24. Levine, D.: *A Parallel Genetic Algorithm for the Set Partitioning Problem*, Ph. D. Thesis, Illinois Institute of Technology / Argonne National Laboratory, 1994.
25. Liu, A., Joe, B., Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision, *Mathematics of Comput.* 1996; **65**(215):1183-1200.
26. Löhner, R., Baum, J.D., Adaptive *h*-refinement on 3D unstructured grids for transient problems. *Int. J. Num. Meth. Fluids* 1992; **14**:1407-1419.
27. Löhner, R., Parikh, P., Three-dimensional grid generation by advancing front method. *Int. J. Num. Meth. Fluids* 1988; **8**:1135-1149.
28. McRae, G.J., Goodin, W.R., Seinfeld, J.H., Development of a second generation mathematical model for urban air pollution I. Model formulation. *Atm. Env.* 1982; **16**(4):679-696.
29. Montenegro, R., Montero, G., Escobar, J.M., Rodríguez, E., González-Yuste, J.M., Tetrahedral Mesh Generation for Environmental Problems over Complex Terrains. *Lecture Notes in Computer Science*, Springer Verlag:Berlin Heidelberg New York, 2002. accepted.
30. Montenegro, R., Plaza, A., Ferragut, L., Asensio, I., Application of a Nonlinear Evolution Model to Fire Propagation. *Nonlin.r Anal., Theor., Meth. Appl.* 1997; **30**(5):2873-2882.
31. Montero, G., Solving Optimal Control Problems by GAs. *Nonlin. Anal. Theor. Meth. Appl.* 1997; **30**(5):2891-2902.
32. Montero, G., Montenegro, R., Escobar, J.M., A 3-D Model for Wind Field Adjustment. *J. Wind Eng. Ind. Aer.* 1998; **74-76**:249-261.
33. Montero, G., Sanín, N., 3-D Modelling of Wind Field Adjustment Using Finite Differences in a Terrain Conformal Coordinate System. *J. Wind Eng. Ind. Aer.* 2001; **89**:471-488.
34. Moussiopoulos, N., Flassak, Th., Knittel, G., A Refined Diagnostic Wind Model. *Environ. Software* 1988; **3**:85-94.
35. Murphy, M., Mount, D.M., Gable, C.W., A Point-Placement Strategy for Conforming Delaunay Tetrahedralization. In *Symposium on Discrete Algorithms*, 2000; 67-74.
36. Panofsky, H.A., Dutton, J.A., *Atmospheric Turbulence*, John Wiley:New York, 1984.
37. Plaza, A., Montenegro, R., Ferragut, F., An Improved Derefinement Algorithm of Nested Meshes. *Adv. Eng. Soft.* 1996; **27**(1/2):51-57.
38. Plaza, A., Carey, G.F., Local refinement of simplicial grids based on the skeleton. *Appl. Numer. Math.* 2000; **32**:195-218.
39. Ratto, C.F., An Overview of Mass-consistent Models. In *Modelling of Atmospheric Flow Fields*, Lalas, D.P., Ratto, C.F. (eds.), World Sci:Singapore, 1996; 379-400.
40. Ratto, C.F., The AIOLOS and WINDS Codes. In *Modelling of Atmospheric Flow Fields*, Lalas, D.P., Ratto, C.F. (eds.), World Sci:Singapore, 1996; 421-431.
41. Rivara, M.C., A Grid Generator Based on 4-Triangles Conforming. Mesh-Refinement Algorithms. *Int. J. Num. Meth. Eng.* 1987; **24**:1343-1354.
42. Rivara, M.C., Levin, C., A 3-d refinement algorithm suitable for adaptive multigrid techniques. *J. Comm. Appl. Numer. Meth.* 1992; **8**:281-290.
43. Ross, D.G., Smith, I.N., Manins, P.C., Fox, D.G., Diagnostic Wind Field Modelling for Complex Terrain: Model Development and Testing. *J. Appl. Meteorol.* 1988; **27**:785-796.
44. Sherman, C.A., A Mass-Consistent Model for Wind Fields over Complex Terrain. *J. Appl. Meteorol.* 1978; **17**:312-319.
45. Spears, W., DeJong, K., On the Virtues of Parametrized Uniform Crossover. In *Proc. of the Fourth International Conference on Genetic Algorithms*, 1991.
46. Syswerda, G., Uniform Crossover in Genetic Algorithms. In *Proc. of the Third International Conference on Genetic Algorithms*, 1989.
47. Tombrou, M., Lalas, D.P., A Telescoping Procedure for Local Wind Energy Potential Assessment. In *European Community Wind Energy Conference*, Palz, W. (ed.), H.S. Stephens & Associates:Madrid, 1990.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls

48. Whitley, D., The GENITOR Algorithm and Selection Pressure: Why Rank-based Allocation of Reproductive Trials is Best. In *Proc. of the Third International Conference on Genetic Algorithms*, 1989.
49. Whitley, D., GENITOR: A Different Genetic Algorithm. In *Rocky Mountain Conference on Artificial Intelligence*, 1988.
50. Winter, G., Montero, G., Ferragut, L., Montenegro, R., Adaptive Strategies Using Standard and Mixed Finite Elements for Wind Field Adjustment. *Solar Energy* 1995; **54**(1):49-56.
51. Zannetti, P., *Air pollution modeling*, Computational Mechanics Publications:Boston, 1990.

Copyright © 2002 U.L.P.G.C.

Prepared using iusianirep.cls