

## **Abstract**

The efficiency of a mass consistent model for wind field adjustment depends on several parameters that arise in various stages of the process. On one hand, 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 stability parameter 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 some of the wind velocities observed at the measurement station are regenerated as accurately as possible by the model. In addition, we study the effect of the mesh refinement on the parameter estimation in several numerical experiments.

**Keywords:** Parameter estimation, genetic algorithms, wind field modelling, mass consistent models, adaptive mesh refinement, finite element method.

## **1 Introduction**

A three-dimensional finite element model for wind field adjustment is developed. In general, these problems are defined over regions with complex terrain, therefore a suitable discretization of the studied zone will be necessary. Here, we have used a technique for constructing tetrahedral meshes which are adapted to the terrain orography and have a higher density of nodes near the terrain surface [1]. In section 2, our mass consistent model is presented. It generates a velocity field for an incompressible fluid which adjusts to an initial one obtained from experimental measures and physical considerations. The construction of the initial field is developed in section 3. The first step is to carry out an horizontal interpolation at the height of the measurement stations

over the terrain. From these data, vertical profiles are constructed taking into account the atmospheric stability, the roughness of the terrain, the geostrophic wind and the atmospheric stratification. Once the initial field is computed, we formulate the mass continuity equation of an incompressible fluid with non-flow-through boundary condition on the terrain surface. The adjustment is carried out by a least square function. The Lagrange multipliers technique leads to an elliptic problem which is solved by using the finite elements method. However, there may exist some zones of the domain where more accuracy of the numerical solution is required due to the irregularity of the terrain as well as to strong variations of the solution. In order to improve the solution, an adaptable refinement of the three-dimensional mesh is proposed in section 4. First, for each element of the mesh to be refined, an error indicator is computed attending to the current numerical solution. These point out what elements must be refined. The proposed refinement technique, based on the subdivision in 8-subtetrahedra, allows a higher discretization of the selected zones without excessive propagation along the mesh. This process may be repeated until the error indicators of the numerical solution satisfy the imposed tolerance. In section 5, we remark the parameters of the wind model to be estimated which lead us to construct the fitness function. Genetic algorithms are used to solve this parameter estimation problem and their properties and possibilities are briefly described in section 6. Section 7 is devoted to solve the same numerical experiments presented in [2] in order to show the improvements obtained with the mesh refinement. Finally, our conclusions are presented in section 8.

## 2 Mass Consistent Model in 3-D

This model [3] is based on the continuity equation for an incompressible flow where the air density is constant in the domain  $\Omega$  and *no-flow-through* conditions on  $\Gamma_b$  (terrain and top) are considered

$$\vec{\nabla} \cdot \vec{u} = 0 \quad \text{in } \Omega \quad (1)$$

$$\vec{n} \cdot \vec{u} = 0 \quad \text{on } \Gamma_b \quad (2)$$

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

$$E(\vec{u}) = \int_{\Omega} [\alpha_1^2 ((\tilde{u} - u_0)^2 + (\tilde{v} - v_0)^2) + \alpha_2^2 (\tilde{w} - w_0)^2] d\Omega \quad (3)$$

where the interpolated wind  $\vec{v}_0 = (u_0, v_0, w_0)$  is obtained from experimental measurements, and  $\alpha_1, \alpha_2$  are the Gauss precision moduli. This problem is equivalent to find a saddle point  $(\vec{v}, \phi)$  of the Lagrangian (see [4])

$$E(\vec{v}) = \min_{\vec{u} \in K} \left[ E(\vec{u}) + \int_{\Omega} \phi \vec{\nabla} \cdot \vec{u} d\Omega \right] \quad (4)$$

being  $\vec{v} = (u, v, w)$ ,  $\phi$  the Lagrange multiplier and  $K$  the set of admissible functions. The Lagrange multipliers technique is used to minimise the problem (4), whose

minimum comes to form the Euler-Lagrange equations

$$u = u_0 + T_h \frac{\partial \phi}{\partial x}, \quad v = v_0 + T_h \frac{\partial \phi}{\partial y}, \quad w = w_0 + T_v \frac{\partial \phi}{\partial z} \quad (5)$$

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  $\alpha_1$  and  $\alpha_2$  are constant in  $\Omega$ , the variational approach results in an elliptic problem substituting (5) in (1)

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{T_v}{T_h} \frac{\partial^2 \phi}{\partial z^2} = -\frac{1}{T_h} \left( \frac{\partial u_0}{\partial x} + \frac{\partial v_0}{\partial y} + \frac{\partial w_0}{\partial z} \right) \quad \text{in } \Omega \quad (6)$$

We consider Dirichlet condition for open or *flow-through* boundaries and Neumann condition for terrain and top

$$\phi = 0 \quad \text{on } \Gamma_a \quad (7)$$

$$\vec{n} \cdot T \vec{\nabla} \mu = -\vec{n} \cdot \vec{v}_0 \quad \text{on } \Gamma_b \quad (8)$$

The problem given by (6)-(8), is solved using tetrahedral finite elements (see [1]) that leads to a set of  $4 \times 4$  elemental matrices and  $4 \times 1$  elemental vectors. These are assembled to form a symmetric linear system of equations which is solved by a preconditioned conjugate gradient method.

### 3 Interpolated Wind

The first step for constructing the interpolated wind is the so-called horizontal interpolation. The wind speeds measured are interpolated at station height  $z_m$  using the distance and the height difference between each point and the station [3]

$$\vec{v}_0(z_m) = \varepsilon \frac{\sum_{n=1}^N \frac{\vec{v}_n}{d_n^2}}{\sum_{n=1}^N \frac{1}{d_n^2}} + (1 - \varepsilon) \frac{\sum_{n=1}^N \frac{\vec{v}_n}{|\Delta h_n|}}{\sum_{n=1}^N \frac{1}{|\Delta h_n|}} \quad (9)$$

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  $\varepsilon$  is a weighting parameter ( $0 \leq \varepsilon \leq 1$ ), which allows to give more importance to one of these interpolation criteria.

In the vertical profile of wind, we assume that this model does not take into account the turbulence phenomena near the terrain due to its roughness. Thus, we establish

$$\vec{v}_0(z) = 0 \quad z \leq z_0 \quad (10)$$

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

$$\vec{v}_0(z) = \frac{\bar{v}^*}{k} \left( \log \frac{z}{z_0} - \Phi_m \right) \quad z_0 < z \leq z_{sl} \quad (11)$$

where  $\bar{v}^*$  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. The value of  $\Phi_m$  depends on the air stability

$$\begin{aligned} \Phi_m &= 0 && \text{(neutral)} \\ \Phi_m &= -5 \frac{z}{L} && \text{(stable)} \\ \Phi_m &= \log \left[ \left( \frac{\theta^2 + 1}{2} \right) \left( \frac{\theta + 1}{2} \right)^2 \right] - 2 \arctan \theta + \frac{\pi}{2} && \text{(unstable)} \end{aligned} \quad (12)$$

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*)

$$\bar{v}^* = \frac{k \vec{v}_0(z_m)}{\log \frac{z_m}{z_0} - \Phi_m} \quad (13)$$

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

$$z_{pbl} = \frac{\gamma |\bar{v}^*|}{f} \quad (14)$$

where  $f = 2\omega \sin \varphi$  is the Coriolis parameter ( $\omega$  is the earth rotation and  $\varphi$  the latitude), and  $\gamma$  is a parameter depending on the atmospheric stability. The mixing height  $h$  coincides with  $z_{pbl}$  in neutral and unstable conditions. In stable conditions, Zilitinkevich suggests (see [5])

$$h = \gamma' \sqrt{\frac{|\bar{v}^*| L}{f}} \quad (15)$$

where  $\gamma'$  is another constant of proportionality. The height of the 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

$$\vec{v}_0(z) = \rho(z) \vec{v}_0(z_{sl}) + [1 - \rho(z)] \vec{v}_g \quad z_{sl} < z \leq z_{pbl} \quad (16)$$

$$\rho(z) = 1 - \left( \frac{z - z_{sl}}{z_{pbl} - z_{sl}} \right)^2 \left( 3 - 2 \frac{z - z_{sl}}{z_{pbl} - z_{sl}} \right) \quad (17)$$

Finally, this model assumes

$$\vec{v}_0(z) = \vec{v}_g \quad z > z_{pbl} \quad (18)$$

## 4 Adaptive Mesh Refinement

Nowadays, 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 is necessary due, on one hand, to the geometry and, on the other hand, to the numerical solution. It is carried out the computation of error estimators or at least suitable error indicators of the numerical solution for determining the elements to be refined or derefined in a mesh. Here we propose one error indicator which takes into account the gradient of the solution in each element. We obtain the initial mesh using a non-structured mesh generator [1, 6] and then apply a refinement technique of nested meshes based on this error indicator.

Some 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., [7, 8, 9, 4]). In these works, a version of Rivara 4-T local refinement algorithm [10] 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 [11, 12, 13] and those which use the 8-subtetrahedron subdivision [14, 15, 16]. In fact, the algorithm developed in [13] may be understood as the generalisation 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 algorithms proposed in [14, 15, 16], which generalise the subdivision in 4 subtriangles of Bank et al [17] in 3-D, are simpler due to a lower number of possible subdivisions of a tetrahedron. We propose a refinement algorithm based on the 8-subtetrahedron subdivision developed in [16]. Consider an initial triangulation  $\tau_1$  of the domain given by a set of  $n_1$  tetrahedra  $t_1^1, t_2^1, \dots, t_{n_1}^1$ . Our goal is to build a sequence of  $m$  levels of nested meshes  $T = \{\tau_1 < \tau_2 < \dots < \tau_m\}$ , such that the level  $\tau_{j+1}$  is obtained from a local refinement of the previous level  $\tau_j$ . The error indicator  $\epsilon_i^j$  associated to the element  $t_i^j \in \tau_j$  which has been used is gradient type and it is defined as follows,

$$\epsilon_i^j = d_i^p \left| \vec{\nabla} \phi_h \right| \quad (19)$$

where the parameter  $p$  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  $\epsilon_i^j$  represents the maximal variation of  $\phi_h$  in the element  $t_i^j$ . Once the error indicator  $\epsilon_i^j$  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  $\tau_j$ . From a constructive point of view, initially we shall obtain  $\tau_2$  from the initial mesh  $\tau_1$ , 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 1(a) shows; 6 new nodes are introduced in the middle point of its edges and each one of its faces are subdivided into four subtriangles following the division proposed by

Bank [17]. Thus, four subtetrahedra are determined 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 octahedron which result inside  $t_i^1$ .

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.

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

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

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

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.

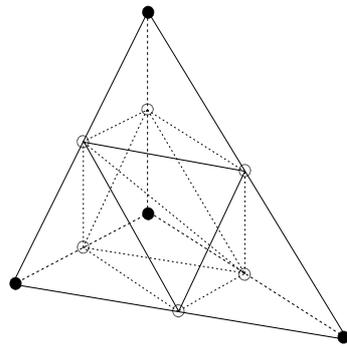
In the following cases, we shall not mark any edge, i.e., any new node will not be introduced in a tetrahedron for conformity. We shall subdivide them creating subtetrahedra which will be called *transient subtetrahedra*.

e.2. If the 3 marked edges are located on the same face of the tetrahedron, then 4 transient subtetrahedra are created as Figure 1(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  $\tau_1$  with these characteristics will be inserted in the set of *type II* tetrahedra.

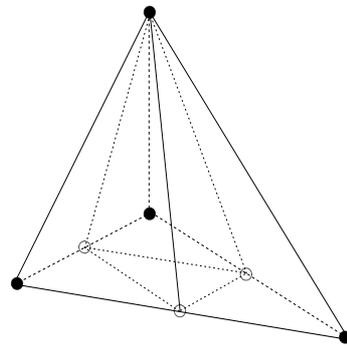
f) *Tetrahedra with 2 new nodes.* Also in this case, we shall distinguish two situations:

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

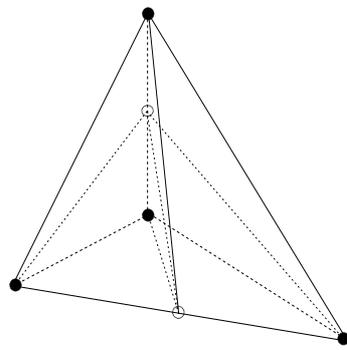
f.2. If the two marked edges are located on the same face, then 3 transient subtetrahedra are generated as Figure 1(d) shows. The face determined by both marked edges is divided into 3 subtriangles, connecting the new node located in the longest edge with the vertex opposite and with the another new 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*.



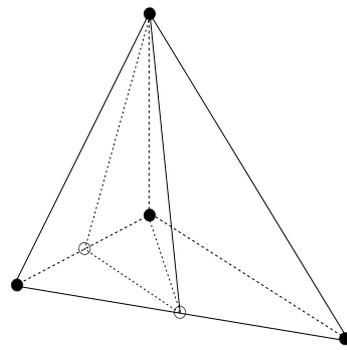
(a) *Type I*



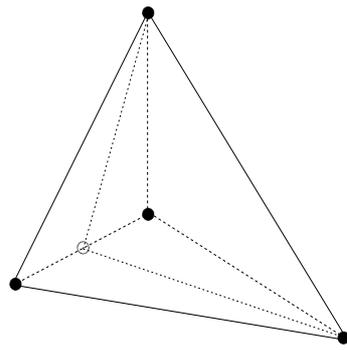
(b) *Type II*



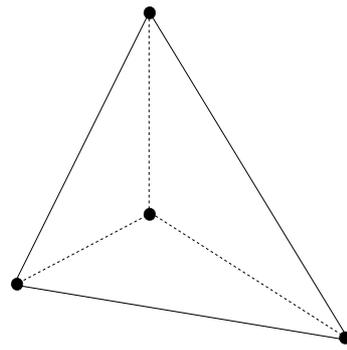
(c) *Type III.a*



(d) *Type III.b*



(e) *Type IV*



(f) *Type V*

Figure 1: Subdivision classification of a tetrahedron in function of the new nodes (white circles).

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  $\tau_1$  are not divided and they will be inherit by the refined mesh  $\tau_2$ . We call them *type V* tetrahedra; see Figure 1(f).

Generally, when we want to refine the level  $\tau_j$  in which there already exist transient tetrahedra, we shall perform it in the same way as from  $\tau_1$  to  $\tau_2$ , except for the following variation: if an edge of any transient tetrahedron must be marked, due to the error indicator or even to conformity reasons, then all the transient tetrahedra are eliminated from their parent (deleting process), all the parent edges are marked and this tetrahedron is introduced into the set of *type I* tetrahedra.

## 5 Discussion on the Parameters to Be Estimated

In the following we define and discuss the four parameter of the wind model that we are interested in estimating. First, we will consider the so called stability parameter

$$\alpha = \frac{\alpha_1}{\alpha_2} = \sqrt{\frac{T_v}{T_h}} \quad (20)$$

since the minimum of the functional given by (3) is the same if we divide it by  $\alpha_2^2$ . On the other hand, for  $\alpha \gg 1$  flow adjustment in the vertical direction predominates, while for  $\alpha \ll 1$  flow adjustment occurs primarily in the horizontal plane. Thus, the selection of  $\alpha$  allows the air to go over a terrain barrier or around it, respectively [18]. Moreover, the behaviour of mass consistent models in many numerical experiments has shown that they are very sensitive to the values chosen for  $\alpha$ . 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  $\alpha$  have limited the possible wide use of mass-consistent models in complex terrain. Sherman [19], Kitada et al. [20] and Davis et al. [21], proposed to take  $\alpha = 10^{-2}$ , i.e., proportional to the magnitude of  $w/u$ . Other authors such as Ross et al. [22] and Moussiopoulos et al. [23] related  $\alpha$  to the Froude number. Geai [24], Lalas et al. [25] and Tombrou et al. [26], make the  $\alpha$  parameter vary in the vertical direction. Finally, Barnard et al. [27] proposed a procedure to obtain  $\alpha$  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  $\alpha$ . 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  $\alpha$  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 [27], using genetic algorithms as optimisation technique which lead to an automatic selection of  $\alpha$ .

The second parameter to be estimate is the weighting coefficient  $\varepsilon$  ( $0 \leq \varepsilon \leq 1$ ) of (9). Note that  $\varepsilon \rightarrow 1$  signifies more importance of the *horizontal distance* from

each point to the measurement stations, while  $\varepsilon \rightarrow 0$  signifies more importance of the *height difference* between each point and the measurement stations [3]. 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  $\varepsilon$  should be more useful.

The next parameter to discuss is  $\gamma$ , given in (14) and related to the height of the planetary boundary layer. There exist different versions of where to search this parameter. Panofsky et al. [28] proposed the interval [0.15,0.25]. On the other hand, Ratto [29] directly suggested  $\gamma = 0.3$  in the *WINDS* code, while  $\gamma$  is located in [0.3,0.4] by de Baas [30]. Therefore, in our simulations, the search space for  $\gamma$  must include all these possibilities.

Finally, we are interested in obtaining suitable values of the parameter  $\gamma'$  involved in the computation of the mixing height for stable atmosphere, see (15). Garratt [31] proposed  $\gamma' = 0.4$ . Also in the *WINDS* code one may find bounds of  $\gamma'$  around 0.4. Thus, the value of  $\gamma'$  will be searched in the surroundings of 0.4.

## 6 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 [32]. 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 [33]. The second, known as steady-state, only replaces a few individuals each generation [34, 35, 36]. 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 [32]: 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 [37].

The mutation operator is better used after crossover [38]. 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  $\nu$  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  $\nu$  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, we look for optimal values of  $\alpha$ ,  $\varepsilon$ ,  $\gamma$  and  $\gamma'$ . 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

$$F(\alpha, \varepsilon, \gamma, \gamma') = \frac{\sum_{n=1}^{N_r} \frac{|\vec{v}_n - \vec{v}(x_n, y_n, z_n)|}{|\vec{v}_n|}}{N_r} \quad (21)$$

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.

## 7 Numerical Experiments

We study the same wind field problem (cases I and III) related to the southern area of La Palma Island (Canary Islands) which was defined in [39, 2]. A  $45600 \times 31200 \times 9000 \text{ m}^3$  domain with real data of the topography is discretized using the code developed in [1]. The maximum height in this zone of the island is  $2279 \text{ m}$ . The initial mesh has been refined around the location of the measurement stations and contains 11416 nodes and 55003 tetrahedra; see Figure 2. The wind measurements were taken in four stations: MBI, MBII, MBIII and LPA. In case I we consider softly unstable conditions and in case III 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 (9), i.e.,  $N = 3$ , and the measurement of MBIII is considered as reference station in the fitness function (21), i.e.,  $N_r = 1$ .

In the first application (case I), the parameter  $\gamma'$  is not involved in the modelling due to the unstable condition of the atmosphere, i.e.,  $h = z_{pbl}$ . Thus, only  $\alpha$ ,  $\varepsilon$  and  $\gamma$  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  $\varepsilon \in [0, 1]$ .

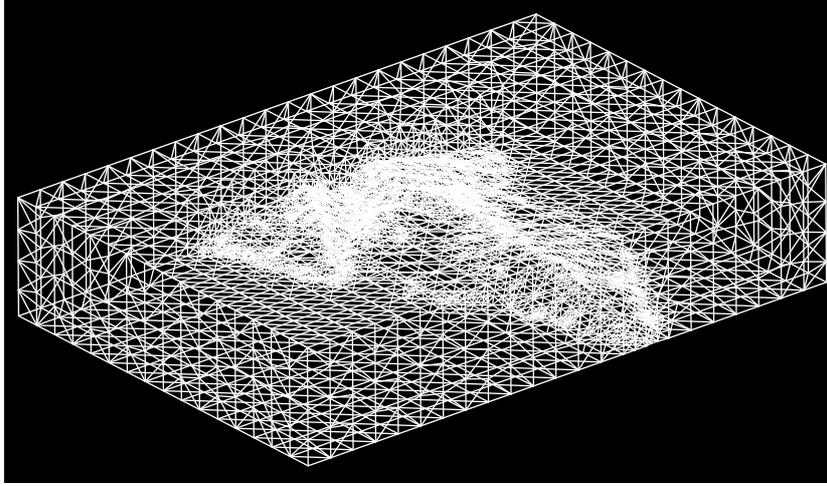


Figure 2: Detail of the finite element mesh used for the numerical experiment. Only the triangulation of the boundary is plotted in order to hold clarity.

The second column of Table 1 (*Stage 1*) shows the values obtained for  $\alpha$  and  $\varepsilon$ , 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 4.96%. 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,  $\alpha$ ,  $\varepsilon$  and  $\gamma \in [0.15, 0.5]$  are estimated. The results are also shown in the third column of Table 1. We observe that  $\alpha$  takes the maximum value of the space of search,  $\varepsilon$  remains around 0.5 and  $\gamma$  is reduced, such that the error at station MBIII is 4.76%. 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 68.07% and 34.62% in each *stage*, respectively. Therefore, the knowledge of suitable values 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 ( $\alpha$ ,  $\varepsilon$ ) is solved. The second column of Table 2 (*Stage 1*) shows the values obtained for  $\alpha$ ,  $\varepsilon$ . 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 remains 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 error obtained at station MBIII was about 11.87%, while the error related to the worst evaluation was 994.2% in *Stage 6*. In both experiments, the number of individuals of the initial population was 100, except for stage 6 in case III where it was 150.

Figures 3 and 4 show the evolution of the parameters and the error along successive generations for cases I and III, respectively. Note that  $\alpha$  has been scaled in order to

	Stage 1	Stage 2
GAs strategy	BT, U, R	SU, U, G
Iterations	88	135
CPU time (s)	10385	16194
Best Fitness	0.496	0.476
$\alpha$	9.978	10.000
$\varepsilon$	0.609	0.484
$\gamma$	(0.300)	0.150

Table 1: First experiment corresponding to the case I analysed in [39]. 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				
Iterations	81	82	93	123	435	431
CPU time (s)	9613	9478	10970	14758	50849	75692
Best Fitness	0.181	0.161	0.125	0.121	0.119	0.119
$\alpha$	10.000	9.968	(9.968)	9.922	9.995	9.999
$\varepsilon$	0.672	0.780	0.808	(0.808)	0.810	0.808
$\gamma$	(0.300)	0.244	0.234	0.230	(0.230)	0.231
$\gamma'$	(0.400)	(0.400)	0.164	0.151	0.150	0.150

Table 2: Second experiment corresponding to the case III analysed in [39]. Strategy of genetics algorithms, best evaluation of the fitness function and values of the parameters (*fixed values are written between brackets*)

plot all the evolutions together.

Iterations and CPU timings on a five nodes cluster of 1,6 GHz Pentium 4 are shown in Tables 1 and 1 for each stage, running 2 processes each node. We also try 3 and 4 processes each node, however, the strategy of running 2 processes per node was the fastest. Evidently, the running time is considerably reduced working in a parallel environment, where GAs become competitive with other optimisation methods. We remark that the evaluation of one individual of any generation means the resolution of a wind problem by the finite element method using two adaptive mesh refinement steps (i.e., three meshes).

If we compare the results obtained here for both cases with those obtained in [2], we observe that the error has been reduced about a half in each experiment.

Finally, as example, Figures 5 and 6 show the refined meshes obtained in the two steps of adaptive mesh refinement for the computation of the wind field in the second experiment, 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 inter-

polated wind field. Figures 7 and 8 illustrate the streamlines and the velocities of wind obtained by the model at a height of 500 m.

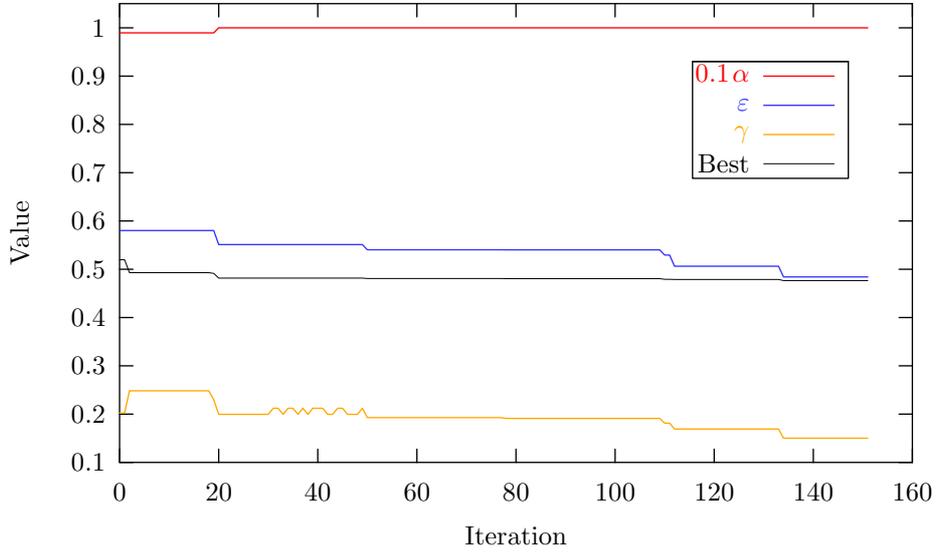


Figure 3: Evolution of the parameters and Best Fitness along the successive generation for *stage 2* of case I.

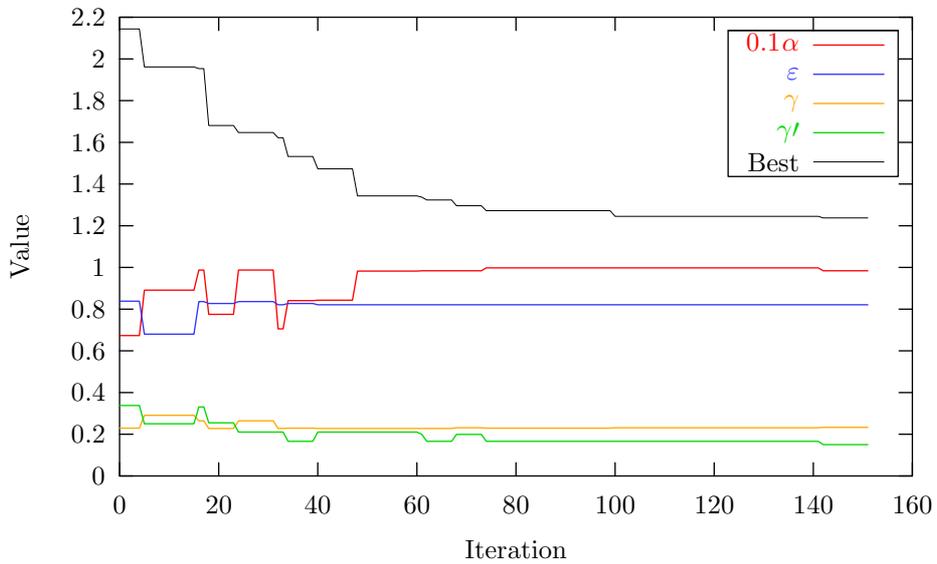


Figure 4: Evolution of the parameters and Best Fitness along the successive generation for *stage 6* of case III.

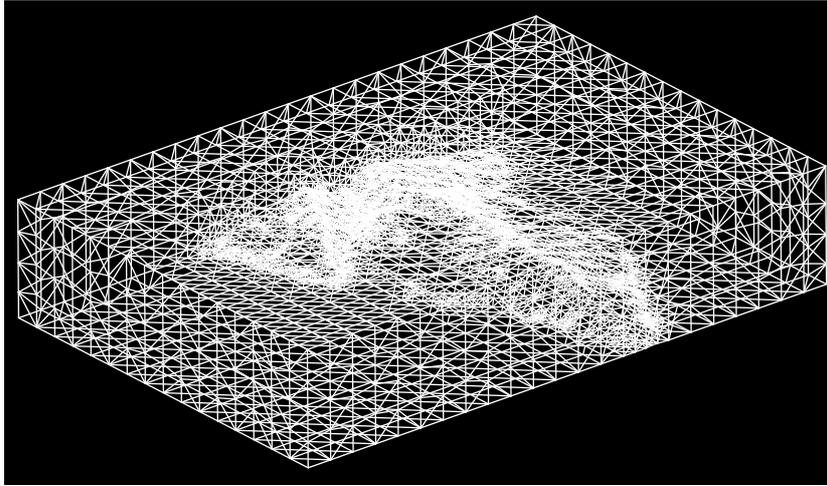


Figure 5: Detail of the refined mesh obtained in the first step of refinement.

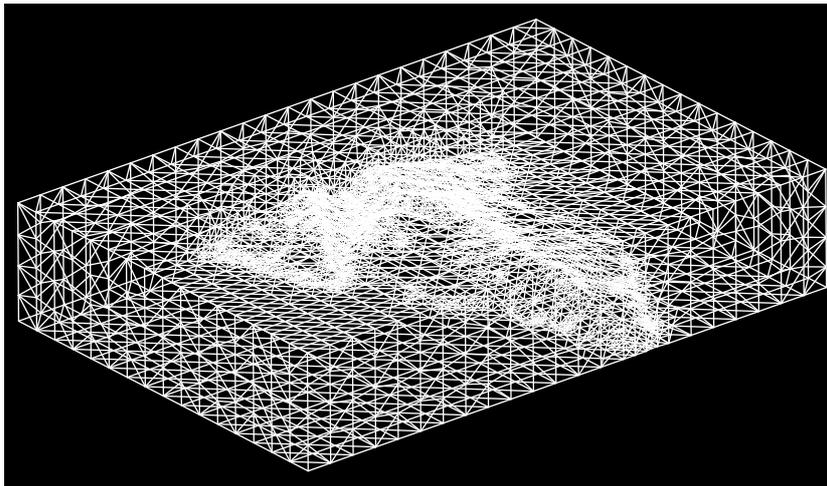


Figure 6: Detail of the refined mesh obtained in the second step of refinement.

## 8 Conclusions

We have pointed out that 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  $\alpha$ ,  $\varepsilon$ ,  $\gamma$  and  $\gamma'$ . 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 [40]). Adaptive mesh refinement techniques allow us to reduce the error in the reference stations. Finally, the resolution by GAs using a cluster of computers leads to competitive timings compared to other optimisation solvers.

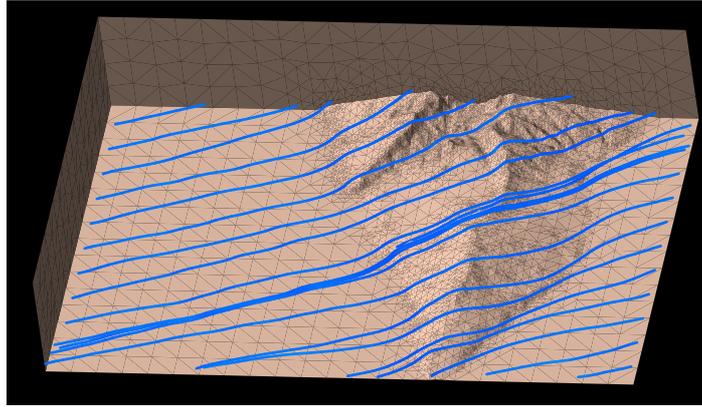


Figure 7: Streamlines of wind velocities related to the second experiment at a height of 500 *m*



Figure 8: Wind velocities related to the second experiment at a height of 500 *m*

## Acknowledgements

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

## References

- [1] R. Montenegro, G. Montero, J.M. Escobar, E. Rodriguez, J.M. Gonzalez-Yuste, “*Tetrahedral Mesh Generation for Environmental Problems over Complex Ter-*

- rains”, Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg New York, 335-344, 2002.
- [2] E. Rodriguez, G. Montero, R. Montenegro, J.M. Escobar, J.M. Gonzalez-Yuste, “*Parameter Estimation in a Three-dimensional Wind Field Model Using Genetic Algorithms*”, Lecture Notes in Computer Science, Springer Verlag, Berlin Heidelberg New York, 950-959, 2002.
- [3] G. Montero, R. Montenegro, J.M. Escobar, “*A 3-D Model for Wind Field Adjustment*”, J. Wind Eng. & Ind. Aer., 74-76, 249-261, 1998.
- [4] G. Winter, G. Montero, L. Ferragut, R. Montenegro, “*Adaptive Strategies Using Standard and Mixed Finite Elements for Wind Field Adjustment*”, Solar Energy, 54(1), 49-56, 1995.
- [5] J.A. Businger, S.P.S. Arya, “*Heights of the Mixed Layer in the Stable, Stratified Planetary Boundary Layer*”, Adv. Geophys., 18A, 73-92, 1974.
- [6] J.M. Escobar, R. Montenegro, “*Several aspects of three-dimensional Delaunay triangulation*”, Advances in Engineering Software, 27(1/2), 27-39, 1996.
- [7] L. Ferragut, R. Montenegro, A. Plaza, “*Efficient refinement/derefinement algorithm of nested meshes to solve evolution problems*”, Comm. Num. Meth. Eng., 10, 403-412, 1994.
- [8] A. Plaza, R. Montenegro, L. Ferragut, “*An improved derefinement algorithm of nested meshes*”, Advances in Engineering Software, 27(1/2), 51-57, 1996.
- [9] R. Montenegro, A. Plaza, L. Ferragut, I. Asensio, “*Application of a nonlinear evolution model to fire propagation*”, Nonlinear Analysis, Th., Meth.& App., 30(5), 2873-2882, 1997.
- [10] M.C. Rivara, “*A grid generator based on 4-triangles conforming. Mesh-refinement algorithms*”, Int. J. Num. Meth. Eng., 24, 1343-1354, 1987.
- [11] D.N. Arnold, A. Mukherjee, L. Pouly, “*Locally adapted tetrahedral meshes using bisection*”, SIAM J. Sci. Comput., 22(2), 431-448, 2000.
- [12] M.C. Rivara, C. Levin, “*A 3-d refinement algorithm suitable for adaptive multi-grid techniques*”, J. Comm. Appl. Numer. Meth., 8, 281-290, 1992.
- [13] A. Plaza, G.F. Carey, “*Local refinement of simplicial grids based on the skeleton*”, Appl. Numer. Math., 32, 195-218, 2000.
- [14] F. Bornemann, B. Erdmann, R. Kornhuber, “*Adaptive multilevel methods in three space dimensions*”, Int. J. Numer. Meth. Eng., 36, 3187-3203, 1993.
- [15] A. Liu, B. Joe, “*Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision*”, Mathematics of Comput., 65(215), 1183-1200, 1996.
- [16] R. Lohner, J.D. Baum, “*Adaptive h-refinement on 3D unstructured grids for transient problems*”, Int. J. Num. Meth. Fluids, 14, 1407-1419, 1992.
- [17] R.E. Bank, A.H. Sherman, A. Weiser, “*Refinement algorithms and data structures for regular local mesh refinement*”, in Scientific Computing IMACS, Amsterdam, North-Holland, 3-17, 1983.
- [18] C.F. Ratto, “*An Overview of Mass-consistent Models*”, In: Lalas, D.P., Ratto, C.F. (eds.): Modelling of Atmospheric Flow Fields. World Sci., Singapore, 379-400, 1996.

- [19] C.A. Sherman, "A Mass-Consistent Model for Wind Fields over Complex Terrain", J. Appl. Meteorol., 17, 312-319, 1978.
- [20] T. Kitada, A. Kaki, H. Ueda, L.K. Peters, "Estimation of Vertical Air Motion from Limited Horizontal Wind Data - A Numerical Experiment", Atmos. Environ., 17, 2181-2192, 1983.
- [21] C.G. Davis, S.S. Bunker, J.P. Mutschlecner, "Atmospheric Transport Models for Complex Terrain", J. Climate Appl. Meteorol., 23, 235-238, 1984.
- [22] D.G. Ross, I.N. Smith, P.C. Manins, D.G. Fox, "Diagnostic Wind Field Modelling for Complex Terrain: Model Development and Testing", J. Appl. Meteorol., 27, 785-796, 1988.
- [23] N. Moussiopoulos, Th. Flassak, G. Knittel, "A Refined Diagnostic Wind Model", Environ. Software, 3, 85-94, 1988.
- [24] P. Geai, "Methode d'Interpolation et de Reconstitution Tridimensionnelle d'un Champ de Vent: le Code d'Analyse Objective MINERVE", Technical Report DER/HE/34-87.03, EDF, Chatou, France, 1985.
- [25] D.P. Lalas, M. Tombrou, M. Petrakis, "Comparison of the Performance of Some Numerical Wind Energy Siting Codes in Rough Terrain", In European Community Wind Energy Conference, Herning, Denmark, 1988.
- [26] M. Tombrou, D.P. Lalas, "A Telescoping Procedure for Local Wind Energy Potential Assessment" In: Palz, W. (ed.): European Community Wind Energy Conference, H.S. Stephens & Associates, Madrid, 1990.
- [27] J.C. Barnard, H.L. Wegley, T.R. Hiester, "Improving the Performance of Mass Consistent Numerical Model Using Optimization Techniques", J. Climate Appl. Meteorol., 26, 675-686, 1987.
- [28] H.A. Panofsky, J.A. Dutton, "Atmospheric Turbulence", John Wiley, New York, 1984.
- [29] C.F. Ratto, "The AIOLOS and WINDS Codes", In: Lalas, D.P., Ratto, C.F. (eds.): Modelling of Atmospheric Flow Fields. World Sci., Singapore, 421-431, 1996.
- [30] A.F. de Baas, "Scaling Parameters and their Estimation", In: Lalas, D.P., Ratto, C.F. (eds.): Modelling of Atmospheric Flow Fields, World Sci., Singapore, 87-102, 1996.
- [31] J.R. Garratt, "Observations in the Nocturnal Boundary Layer", Boundary-Layer Meteorol., 22(1), 21-48, 1982.
- [32] D. Levine, "A Parallel Genetic Algorithm for the Set Partitioning Problem", Ph. D. Thesis, Illinois Institute of Technology / Argonne National Laboratory, 1994.
- [33] J. Holland, "Adaption in Natural and Artificial Systems", MIT Press, 1992.
- [34] G. Syswerda, "Uniform Crossover in Genetic Algorithms", Proc. of the Third International Conference on Genetic Algorithms, 1989.
- [35] D. Whitley, "The GENITOR Algorithm and Selection Pressure: Why Rank-based Allocation of Reproductive Trials is Best", Proc. of the Third International Conference on Genetic Algorithms, 1989.
- [36] D. Whitley, "GENITOR: A Different Genetic Algorithm", Rocky Mountain Conference on Artificial Intelligence, 1988.
- [37] W. Spears, K. DeJong, "On the Virtues of Parametrized Uniform Crossover",

- Proc. of the Fourth International Conference on Genetic Algorithms, 1991.
- [38] L. Davis, "*Handbook of Genetic Algorithms*", Van Nostrand Reinhold, 1991.
- [39] G. Montero, N. Sanin, "*3-D Modelling of Wind Field Adjustment Using Finite Differences in a Terrain Conformal Coordinate System*", J. Wind Eng. & Ind. Aer., 89, 471-488, 2001.
- [40] G. Montero, "*Solving Optimal Control Problems by GAs*", Nonlin. Anal., Theor., Meth., & Appl., 30(5), 2891-2902, 1997.