

MESH OPTIMIZATION: SOME RESULTS IN 3D ELASTICITY

GUSTAVO C. BUSCAGLIA, ENZO A. DARI, AND PABLO D. ZAVATTIERI

ABSTRACT. Though the mathematical theory of classical linear elasticity is well established, there still lack some ingredients toward the numerical solution of real technological problems. In this paper we address one of these critical ingredients, namely the automatic construction of three-dimensional meshes in arbitrary geometries. Several methods exist for this purpose, but further improvements are still required to achieve the needed robustness and generality. We present and discuss the idea of mesh optimization, namely the manipulation of the mesh geometry and topology so as to maximize some suitable quality measure. The effects of mesh optimization in the finite element solution of a linear thermoelastic problem are evaluated. Finally, we report on a recent method that couples mesh optimization with a posteriori error estimation ideas, so that mesh refinement in regions of high stress gradients is achieved through optimization using a suitable space-varying metric. Numerical results for this last techniques are restricted to two dimensions, as a 3D implementation is under way.

1. INTRODUCTION

The mathematical theory of linear elasticity being well established, it is the geometry of the solution domain that still makes it difficult to obtain an accurate approximation of the exact solution in real-life problems. Thin-walled structural components resist straightforward finite element treatment because of mesh degeneration and locking, and plate/shell theory is certainly the adequate mathematical and numerical tool to avoid this difficulty (the reader is referred to other lectures of this conference for state-of-the-art developments in plate/shell theory). There remain, of course, many problems in linear elasticity with domains that are truly three-dimensional. In these cases the bottleneck in the analysis is the construction of a three-dimensional mesh fitting in the domain under consideration.

Much research effort has been devoted, during the last years, to the development of effective algorithms for the generation of grids in general 3D domains. For this process to be automatic, the current choice is that of unstructured meshes of tetrahedra. Significant progress has been achieved concerning the robustness and flexibility of unstructured algorithms (frontal methods, Delaunay-based methods, octree-based methods, and variants of them). As a consequence, it has been possible to mesh complex domains,

Date: June 30, 1996.

1991 Mathematics Subject Classification. Primary 65N30, 65N50; Secondary 65C20, 65K10.

Key words and phrases. Mesh generation, Optimization, Adaptivity, Mesh Optimization, Unstructured Grids.

This paper is in final form and no version of it will be submitted for publication elsewhere.

allowing for massive application of finite element or finite volume solvers to industrial problems.

Among the several difficulties that still remain to be solved, this presentation concerns that associated with the geometrical quality of the resulting meshes. Badly distorted elements and over- or under-refined regions are not unusual in current 3D meshes. A complete, automatic control on the mesh quality is a primary goal in the present state of the field.

The application of optimization techniques during the generation procedure has proved useful for this purpose. This can be done during both the element-creation step [1, 2, 3, 4] and/or the *a posteriori* mesh improving stage [5, 6, 7, 8, 9, 10, 11, 12, 13]. In Section 2 we include the description of a complete quality-based mesh improving method, addressing the optimization of nodal positions and of the connectivity structure of the mesh. In particular, the node-repositioning technique consists of a non-differentiable optimization algorithm over the space of nodal positions. The connectivity changes are based on local cluster reconnection, a technique that can be seen as the extension of diagonal-swapping but involves significant complexity in 3D [3, 14, 15]. In Section 3 we address the question of how much does the quality of the mesh affect 3D solid mechanic calculations. For this purpose, we first solve a simple academic problem with analytical solution, which is an oversimplified version of a more interesting technological problem. The knowledge of the exact solution allows us to evaluate the numerical errors and look at the effect of optimizing the mesh. In Section 4 we report some sample calculations concerning the stress analysis of a nuclear fuel pellet. A linear thermoelastic model is used, and the geometries we treat are pellets with different crack sizes. No exact solution is available, but a consistent 20-30% reduction in the number of conjugate-gradient iterations evidences a better conditioning of the linear system upon optimization.

Finally, we discuss the coupling of our mesh optimization procedure with adaptivity. A mesh quality based upon a solution-adapted variable metric is proposed that automatically leads to suitable refinement and stretching. The adaptive method that results [16] is a robust, optimization-based, variant of the ideas of Peraire, Morgan and Peiró [1, 2, 17], further developed recently by Dompierre *et al* [18]. Though the ideas are easily extendable to three-dimensional problems, technical issues arise when dealing with boundaries in 3D that have delayed the implementation, so that examples are presented in two space dimensions.

2. THE MESH OPTIMIZATION ALGORITHM

The mesh optimizer is part of a 3D mesh generation package that has been built along the last years, more details can be found in [5]. The optimization sequence starts with a connectivity optimization, followed by repositioning the nodes at optimal locations for that structure, and so on. The CPU time required for the full optimization of a mesh consisting of 200,000 tetrahedra, for example, is 2-3 minutes on a 30 Mflop workstation. Several features of the optimizer are described below.

2.1. MESH QUALITY

One of the key points in the construction of mesh optimizers is the definition of the mesh quality. We have adopted a non-differentiable definition: The quality of the mesh τ is defined as

$$Q_\tau = \min \{Q_K, K \in \tau\} \quad (2.1)$$

where K is a simplicial element belonging to the mesh. For each element, its quality Q_K is defined as the product of a shape factor S_K times a size factor, namely

$$Q_K = S_K \exp[-\beta^2 \log^2(h_K/h^*)] \quad (2.2)$$

h_K is the diameter of element K , while h^* is the desired mesh size. The parameter β determines the width of the size quality function. For the shape factor S_K we have selected the expressions

$$S_K = \begin{cases} 20.78 V_K/P_K^2 & \text{in 2D} \\ 1832.82 V_K/P_K^3 & \text{in 3D} \end{cases} \quad (2.3)$$

with V_K the volume (area in 2D) of element K and P_K the sum of its edges. These shape quality functions are simple to evaluate and equivalent to other popular definitions of quality, such as the inscribed/circumscribed radius quotient[5].

2.2. NODE REPOSITIONING

With a given connectivity structure, the nodes are repositioned so as to maximize the mesh quality. Most previous approaches have disregarded the use of an optimization technique for this purpose and adopted some variant of the barycenter or Laplace technique; i.e., to move each node to the centroid of its neighbors. However, it is known that this procedure does not work in 3D [5, 19], and *ad hoc* modifications are necessary.

We propose to modify, if necessary, the nodal locations by means of a non-differentiable optimization technique over the space of nodal coordinates, using as objective function $\psi = -Q_\tau$ (the change in sign is just because the terminology is minimum oriented). It is however unthinkable to optimize the whole mesh simultaneously. Instead, we use the following

: Global algorithm:

: Initialization and Global Parameters: Specify a neighborhood level NL and a maximum number of iterations M . Set boundary nodes as non-movable. Initialize an integer auxiliary constant $PREV$ with 0.

: Iterations:

1. Identify the worst element among those elements of the mesh containing at least one *movable node*, $KWORST$. If the number of *movable nodes* is zero, stop.
2. If $KWORST = PREV$, set the four (three in 2D) nodes of $KWORST$ as *non-movable* and go back to 1.
3. Identify the nodes in a neighborhood of order NL of $KWORST$. The neighbors of order 0 are the nodes that belong to $KWORST$. Neighbors of order 1 are those nodes that are not neighbors of order 0, but are connected by an edge with at least one neighbor of order 0, and so on.
4. Set as *moving nodes* those identified in 3 that have not previously been set as *non-movable*. With these *moving nodes* apply the **Local Algorithm** below to optimize the quality of the sub-mesh

formed by the elements that contain *moving nodes*. This sub-mesh is called *optimization cluster*.

5. Assign to *PREV* the value *KWORST*.

6. Go back to 1, or Stop if the number of iterations is equal to M .

It has been observed [7] that it is sufficient to define the optimization cluster with $NL = 1$. The optimization space at each global iteration has dimension of the order of 100, which can be dealt with without difficulty using the following non-differentiable optimization technique [20] (see also [21])

: Local Algorithm:

: Let $\bar{x} \in \mathfrak{R}^n$ be the vector of coordinates of the moving nodes. We have to minimize $\psi = \max_K f_K$, where $f_K = -Q_K$ is the quality of element K , with negative sign. Of course, only those elements containing moving nodes are considered. Let us assign to these elements the numbers $1, 2, \dots, m$. If \bar{x}_0 represents the current position of the moving nodes, the update $\Delta\bar{x}$ results from the following iterative procedure

1. Compute the search direction

$$\bar{\delta} = - \sum_{K=1}^m \mu_K^* \nabla f_K(\bar{x}_0) \quad (2.4)$$

where $\{\mu_K^*\}_{K=1, \dots, m} \equiv \underline{\mu}^*$ is the solution of

$$\underline{\mu}^* = \arg \min_{\underline{\mu} \in \Xi} \left\{ \left[\sum_{K=1}^m \mu_K [\psi(\bar{x}_0) - f_K(\bar{x}_0)] \right]^2 + L^2 \left\| \sum_{K=1}^m \mu_K \nabla f_K(\bar{x}_0) \right\|^2 \right\} \quad (2.5)$$

with

$$\Xi = \left\{ \underline{\mu} \in \mathfrak{R}^m \mid \mu_K \geq 0, \sum_{K=1}^m \mu_K = 1 \right\} \quad (2.6)$$

This quadratic programming subproblem is solved using the Active Set Method (see, e.g., [22]). The scaling factor L in (2.5) is a typical length.

2. Given $\alpha, \beta \in (0, 1)$, compute the step length λ^* using Armijo's rule

$$\lambda^* = \max \left\{ \lambda \in \mathfrak{R} \mid \lambda = \beta^k (k = 0, 1, 2, \dots), \psi(\bar{x}_0 + \lambda \bar{\delta}) - \psi(\bar{x}_0) \leq \lambda \alpha \|\bar{\delta}\|^2 \right\} \quad (2.7)$$

3. $\Delta\bar{x} = \lambda^* \bar{\delta}$ and go back to 1.

REMARK 1. Solving the quadratic problem (2.5) to find the search direction can be very costly, especially in 3D where the number of elements containing moving nodes is large. The efficiency can be improved [20] including a user-defined tolerance ε and computing the set I_ε

$$K \in I_\varepsilon \iff |\psi(\bar{x}_0) - f_K(\bar{x}_0)| < \varepsilon \quad (2.8)$$

Now, the dimension of the quadratic problem is lowered replacing Ξ by Ξ_ε defined by

$$\Xi_\varepsilon = \left\{ \underline{\mu} \in \mathfrak{R}^m \mid \mu_K \geq 0, \sum_{K=1}^m \mu_K = 1, \mu_K = 0 \text{ if } K \notin I_\varepsilon \right\} \quad (2.9)$$

REMARK 2. The iterations of the Local Algorithm must be stopped when the change in the objective function is below some tolerance TOL_{loc} . A good scale for this tolerance is the value taken for ε : We usually adopt $TOL_{loc} = \varepsilon/10$.

2.3. NODE RECONNECTION

Node reconnection is also carried out iteratively. If \overline{K} is the worst element in the mesh (i.e., $Q_{\overline{K}} \leq Q_K, \forall K$), at each iteration the submesh formed by \overline{K} and its neighbors (elements sharing a node or an edge with \overline{K}) is considered. On this submesh several operations are *virtually* performed, until one is found that yields a better quality. This operation is then *really* performed (the details can be found in [7]). The operations considered are similar to those of [14, 15], they consist of defining a *cluster*, removing its interior, and either connecting all the faces (edges in 2D) in the cluster boundary to one of the boundary nodes or connecting them to a node at the center of the cluster. Two types of clusters are considered, *nodal* clusters are all elements that share some given node, and *edge* clusters are all elements that share some given edge. In the case of nodal clusters the center of the cluster is defined averaging the coordinates of the cluster vertices, while in the case of edge clusters it is defined as the center of the central edge. For the two-dimensional case, the operations are depicted in Fig. 1. Special care must be taken with edges and nodes belonging to the boundary. This is handled by means of projections in 2D, and is the main difficulty for the extension of the algorithm to 3D.

It is interesting to point out that the reconnection strategy above is much more efficient than the one earlier proposed in [6]. Marcum and Marcum & Weatherill [3, 4] have incorporated some local reconnection procedures *during* the creation of elements, with excellent results.

3. A TEST EXAMPLE WITH ANALYTICAL SOLUTION

In the first two examples we show here, β is set to zero, meaning that only the shape quality of the elements is considered. Moreover, the algorithm leaves the surface mesh untouched, so that we are looking for the optimal 3D mesh compatible with a given surface mesh.

An ongoing application of elastic analysis to the nuclear industry provided us with the occasion to evaluate the impact of the mesh quality on finite element computations. The strong ellipticity of the elastic operator suggests that this problem is not highly sensitive to the presence of a few distorted elements within the mesh. In addition, the test reported in this section was conducted on a problem with smooth analytical solution. Our results can thus be viewed as a lower bound for the influence of the mesh quality on actual numerical computations. Though the optimization of the mesh is restricted to, say, the worst 1 percent of the elements, it is observed that this significantly reduces the error in the determination of the maximum stresses, together with the number of conjugate-gradient iterations required to solve the linear system.

As an illustrative academic example, we tabulate below the results of a systematic study for a thermoelastic cylinder subject to a parabolic temperature profile with meshes of increasing size. The mesh spacings are approximately uniform throughout the domain. Listed are the number of elements

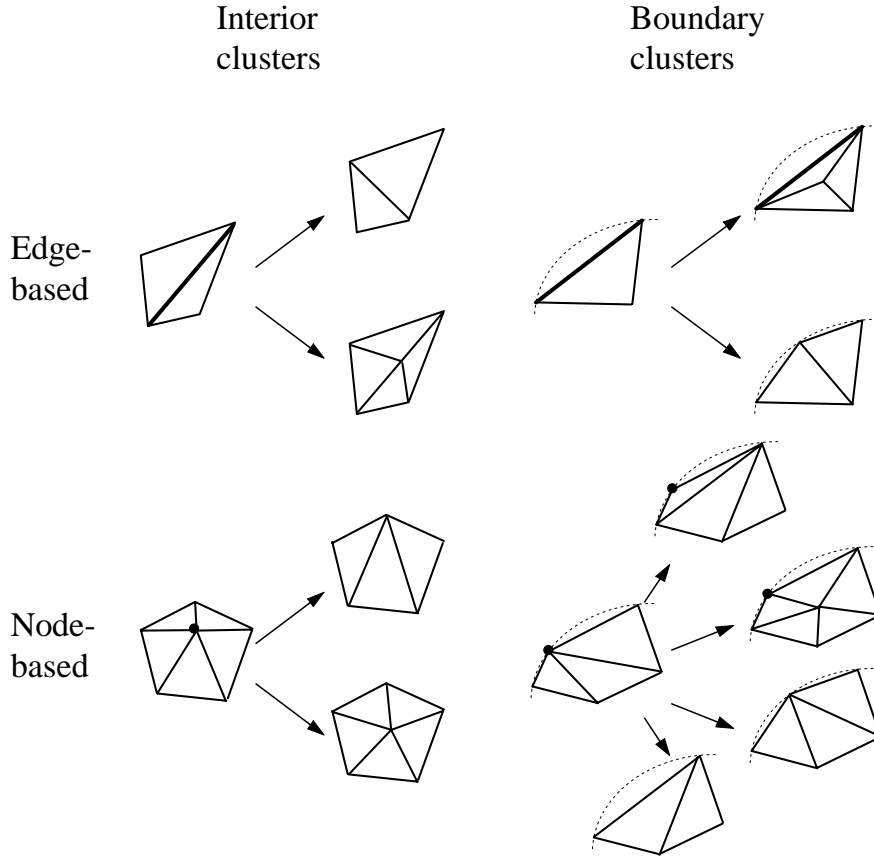


FIGURE 1. Operations allowed in the 2D mesh optimization algorithm. Notice the distinction between node and edge clusters, and between internal and boundary nodes. The 3D variant is analogous.

Mesh	A	B	C	D
Elements	21369	52464	68868	165769
Quality (as generated)	0.103	0.004	0.004	3E-06
L^∞ -error in σ_{rr}	8.5%	15%	10.1%	5.9%
L^∞ -error in σ_{tt}	16%	14%	13.9%	7.5%
CG iter.	307	453	518	1593
Quality (optimized)	0.479	0.386	0.363	0.404
L^∞ -error in σ_{rr}	8.8%	7.8%	7.2%	4.6%
L^∞ -error in σ_{tt}	6.1%	5.7%	5.5%	4.1%
CG iter.	256	326	369	477

TABLE 1. Effect of mesh optimization on the maximum error in the radial and tangential components of the stress, and on the number of conjugate gradient iterations needed to solve the linear system. The problem is that of a linear thermoelastic cylinder subject to a homogeneous heat source.

of each mesh, its quality, the errors in the sup norm of the radial and tangential components of the stress, and the number of (diag-preconditioned) conjugate gradient iterations to solve the linear system.

It is clear from Table 1 that even in this simple case stress pointwise errors can be reduced by half using optimized meshes, saving at the same time up to 70% of linear solver iterations. It is worth mentioning that the error in the displacement field is quite insensitive to the mesh quality (at least in this problem with smooth solution).

The results in Table 1 bring to light an interesting phenomenon related to adaptive analyses. Though the usual condition of "a regular family of triangulations" is always present in mathematical papers, it is seldom checked in practice. Notice that, though mesh B has 2.5 times the number of elements of mesh A and all meshes come from the same generation algorithm, the error in mesh B is larger than in mesh A if no optimization is performed. On the other hand, if we look at the results obtained with optimized meshes, the error decreases steadily when the number of elements is increased. Also, to reduce the error it is more convenient to optimize mesh A than to build the refined mesh C. Smaller elements lead to smaller errors only if their quality is high enough. The meshes required in adaptive analyses have large discretization gradients, a condition that imposes no difficulty to 2D mesh generators but could make 3D ones to yield bad quality elements. For adaptive procedures in 3D to be effective, both the elements' size and their quality have to be kept under control, as both have commensurate effects on the error (particularly on pointwise stress errors).

We should remark that the *average* element quality (\bar{Q}), which is quite high for our meshes, plays no role in the results shown in Table 1. As just the (few) worst elements are modified, \bar{Q} exhibits the following small changes due to optimization: Mesh A, $0.69 \rightarrow 0.71$; Mesh B, $0.72 \rightarrow 0.76$; Mesh C, $0.68 \rightarrow 0.71$; Mesh D, $0.697 \rightarrow 0.703$.

4. SAMPLE RESULTS

It is interesting to consider applications to more technological problems, including locally refined meshes for problems with singularities (see for other meshes in solid and fluid mechanics, our method has also proved useful as a way to delay remeshing in problems with moving boundaries [23]).

We consider a nuclear fuel pellet made of UO_2 , with a radius of 0.531 cm and height of 1.2 cm. The temperature inside it is approximated by $T(r) = [2.468 \times 10^3 - 5.286 \times 10^3 r^2]$ °C, corresponding to a linear power of 599.1 W/cm. For more details concerning the physical problem the reader is referred to [24] and references therein. Three pellets are modeled, with cracks of size 20%, 40% and 60% of the pellet radius. The corresponding meshes can be seen in Fig. 2. Though the purpose of the analysis was to determine the size of the region surrounding the crack where stresses are relieved, the geometry and meshes serve also to test the optimizing procedure in real problems.

In Fig. 3 top views of the pellets showing the distributions of von Mises' equivalent stresses for the three cases can be found. A side view is plotted in Fig. 4. Notice the low-stress zones surrounding the crack mouth. The quality of the Delaunay meshes was $0.002 - 0.004$, and the optimizer successfully took this value to 0.4. Remember that in these cases the surface mesh is *not* modified. Again, only about one percent of the elements were changed, the average quality remained practically the same after optimization (see Table 2).

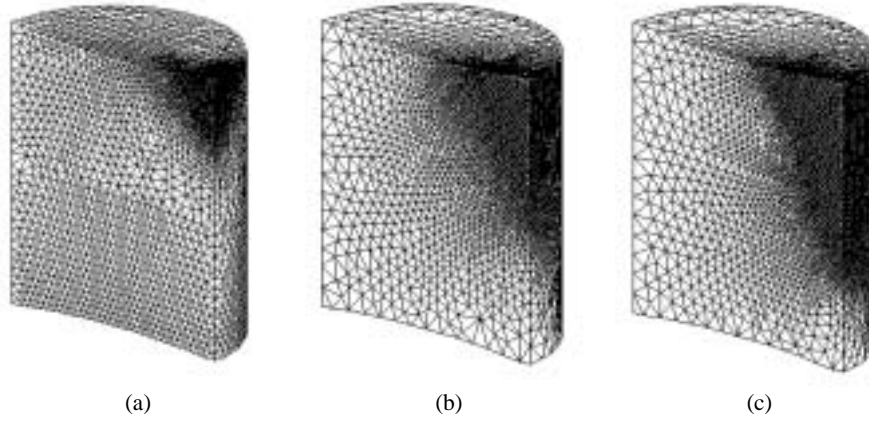


FIGURE 2. Meshes: (a) Crack size 20%, 15519 nodes, 80887 elements, (b) Crack size 40%, 14291 nodes, 74380 elements, (c) Crack size 60%, 17788 nodes, 93260 elements.

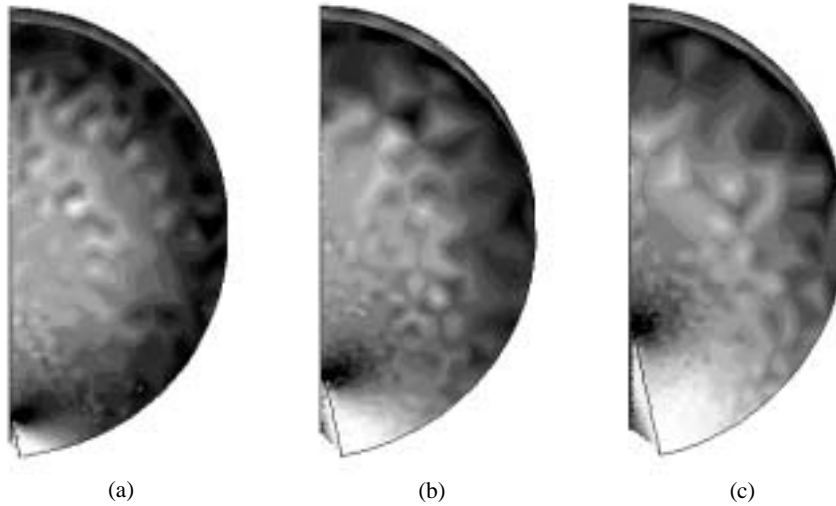


FIGURE 3. Top views of the distributions of von Mises' equivalent stresses. (a) Crack size 20%, (b) 40%, (c) 60%. Darker zones correspond to higher stresses. The domain has been deformed according to the displacement field (suitably scaled to render it visible).

Mesh	Crack 20%	Crack 40%	Crack 60%
Elements	80887	74380	93260
Quality (as generated)	0.002	0.002	0.004
Average quality	0.766	0.742	0.744
CG iter.	990	966	1062
Quality (optimized)	0.421	0.393	0.433
Average quality	0.772	0.743	0.748
CG iter.	775	711	781

TABLE 2. Results of optimization on the meshes for the cracked-pellet geometry.

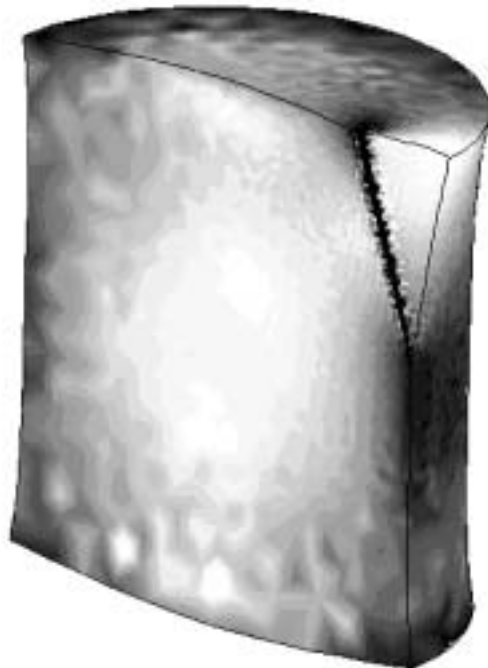


FIGURE 4. Side view of the distribution of von Mises' equivalent stresses for a pellet with crack size 40%.

Notice from Table 2 that mesh optimization consistently reduces the number of conjugate gradient iterations by 20-30%. This is typical of all our tests. From the analysis in the previous section we believe that the savings in CPU time brought by optimization have a counterpart in stress pointwise errors away from the singularity.

5. USING THE OPTIMIZER FOR ADAPTIVE PURPOSES: ADAPTIVE ANISOTROPIC MESH OPTIMIZATION

In the previous sections, we have shown how the optimizer succeeds in finding meshes of high shape quality compatible with some given surface triangulation. This was shown to improve the accuracy of finite element computations. One can go further and use the optimizer for adaptive purposes, as recently proposed [16]. We now turn to show how this can be done.

There exist two alternatives for coupling our optimizer with *a posteriori* error estimation. In Eq. 2.2, h^* can be defined at each point of the domain according to some local error indicator [25, 26]. In this alternative, the optimizer comes to replace the remeshing of the computational domain that is usually performed. We have adopted another strategy, h^* is assumed constant throughout the domain and the optimizer is run without modifications, *but lengths and volumes are evaluated using a solution-adapted metric, instead of the euclidean metric, in the domain.* The space-varying adapted metric finds its rationale in interpolation error estimates, and has been discussed in [1, 2, 18]. Some details of this adaptive optimization method are presented below.

5.1. SOLUTION-ADAPTED METRIC

We recall the solution-adapted metric proposed in Ref. [18]. Assume that a function u is to be interpolated with piecewise linear finite element functions on a mesh of conforming triangles. The interpolation error on an edge will not be isotropic but instead depend on the direction \mathbf{t} of the edge, according to the local value of

$$|D^2u(\mathbf{t}, \mathbf{t})| \equiv \left| \sum_{i,j} t_i t_j \frac{\partial^2 u}{\partial x_i \partial x_j} \right| \quad (5.1)$$

where $\{t_i\}$ stand for the cartesian components of the unit vector \mathbf{t} . This suggests the use of D^2u as a metric tensor, but it fails to be positive definite. Instead, Dompierre *et al* introduce the metric tensor $\mathbf{G} = |D^2u|$. As D^2u is symmetric, it accepts a decomposition of the form $D^2u = \mathbf{Q}\Lambda\mathbf{Q}^T$, with $\Lambda = \text{diag}\{\lambda_1, \lambda_2\}$ the diagonal eigenvalue matrix and \mathbf{Q} orthogonal. \mathbf{G} is defined as $\mathbf{Q}|\Lambda|\mathbf{Q}^T$, with $|\Lambda| = \text{diag}\{|\lambda_1|, |\lambda_2|\}$. This definition provides a space-varying anisotropic metric \mathbf{G} which is at least positive semi-definite, but a zero eigenvalue of \mathbf{G} poses no difficulty, as it leads to infinite mesh sizes that are automatically inhibited by the finiteness of the computational domain Ω .

In the elasticity case, a slight modification is needed since the unknown field is not a scalar but a vector. We have taken \mathbf{G} as

$$\mathbf{G} = + \sqrt{\sum_i (D^2u_i)^2} \quad (5.2)$$

where D^2u_i stands for the Hessian of the i -th component of the displacement field. The Hessians are squared and added, then the resulting matrix is diagonalized and the positive square root of each eigenvalue is used to construct \mathbf{G} .

We apply the optimization method with the quality defined according to (2.2), measuring lengths and areas under the metric \mathbf{G} . The only user-defined parameter is h^* , which is the same throughout the domain and governs the quantity of elements in the optimized mesh. An estimation of h^* is straightforward: The area of the domain with respect to \mathbf{G} is

$$\text{vol}_{\mathbf{G}}(\Omega) = \sum_{K \in \tau} V_K = \sum_{K \in \tau} \tilde{V}_K (\det \mathbf{G})^{1/2} \quad (5.3)$$

where \tilde{V}_K is the area of K in the euclidean metric. The number of elements in the optimized mesh will be close to $(4/\sqrt{3}) \text{vol}_{\mathbf{G}}(\Omega)/(h^*)^2$, and from this expression the desired h^* is readily obtained.

5.2. HESSIAN RECOVERY

It remains to introduce a suitable Hessian D^2u , the optimal one coming from the exact solution which is obviously not known. The alternative is to *recover* the second derivative of the numerical solution $u_h \in V_h$, V_h standing for the finite element space associated with the mesh that will be optimized.

We have implemented the method proposed in [1, 2], namely

$$H_{ij}^I = - (M_L^{II})^{-1} \int_{\Omega} \frac{1}{2} \left(\frac{\partial u_h}{\partial x_i} \frac{\partial N^I}{\partial x_j} + \frac{\partial u_h}{\partial x_j} \frac{\partial N^I}{\partial x_i} \right) d\Omega \quad (5.4)$$

where H_{ij}^I is the value at node I of the ij -component of the recovered Hessian, M_L^{II} is the I -th diagonal entry of the lumped mass matrix, u_h is the numerical solution on the previous mesh and N^I the I -th basis function.

Eq. 5.4 is a discrete version of the identity

$$\int_{\Omega} w D^2 u \, d\Omega = \int_{\Omega} \nabla w \otimes \nabla u \, d\Omega \quad (5.5)$$

which holds for any w vanishing on the boundary $\partial\Omega$. This equation is used to determine H_{ij} for all nodes not belonging to the boundary. H_{ij} is then extrapolated to the boundary by weakly imposing the condition $\partial H_{ij}/\partial n = 0$, with n the normal to $\partial\Omega$.

5.3. A TWO-DIMENSIONAL APPLICATION EXAMPLE

For the proposed adaptive procedure to be effective, both the boundary mesh and the interior mesh need to be modified. A suitable abstraction of the boundary is needed, typically by means of B-splines and other geometric design entities. This renders the implementation quite involved, and our software up to now only supports 2D cases.

Once the computational domain's boundary and applied forces are defined, a first mesh is constructed. This mesh can be very simple, i.e., if the domain is a rectangle, it can consist of just two triangles. This first mesh is then optimized using as metric the euclidean one, so as to get a second mesh with the desired number of elements. A finite element computation is carried out with this second mesh, and with the results a solution-adapted metric is constructed. The algorithm proceeds optimizing the mesh under this new metric, finding a new approximate solution with the improved mesh, and so on. Once two successive meshes are close enough, the process is said to have converged. This typically occurs after three to five adaptive steps.

As an example, let us apply our adaptive optimization procedure to analyze the interaction between two cracks. The cracks are in collinear configuration, with symmetry disturbed by rotation (see Fig. 5). The right crack is rotated a ten-degree angle, a situation that is known to maximize the interaction [27]. This is a quite challenging problem for adaptive techniques, since the stress intensity factors (SIFs) are not the same for all crack tips. The right tip of the left crack and the left tip of the right crack have SIFs about 1.6 times those of the remaining two crack tips. An effective adaptive technique should automatically refine the mesh near crack tips, with greater refinement near those tips of larger SIF.

In Fig. 6 we show the first mesh, with which the process began, the uniform mesh after optimization using the euclidean metric, and the adaptively optimized mesh after four steps. Details of the optimized mesh can be found in Fig. 7. The most critical crack tips have been correctly identified, the mesh size being approximately 5 times larger at the outer crack tips than in the inner ones. In Fig. 8 we include the isocontours of the von Mises' equivalent stress near the cracks.

6. CONCLUSIONS

As said in the introduction, while in other topics of elasticity and viscoelasticity there are several fundamental problems that remain open, the main difficulties in 3D linear elasticity are of numerical nature, one of the most challenging ones being the construction of suitable meshes in general

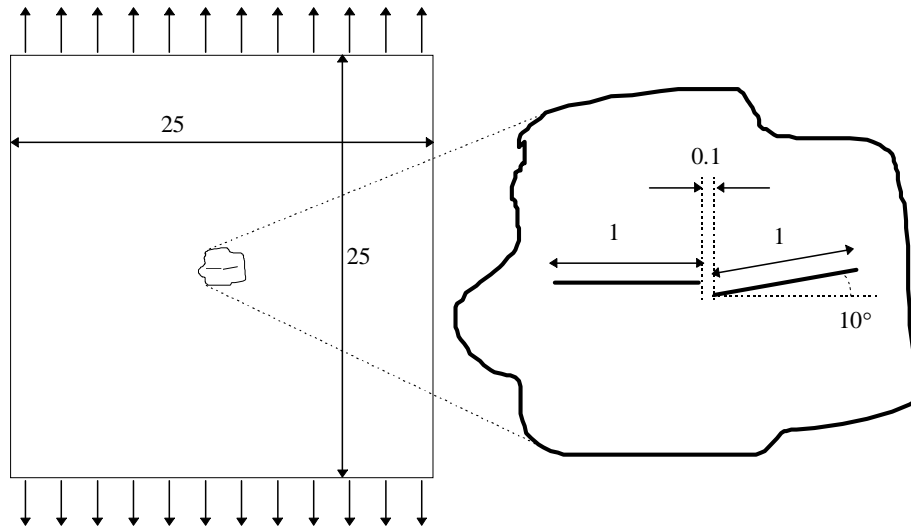


FIGURE 5. Domain definition for the two-crack problem.

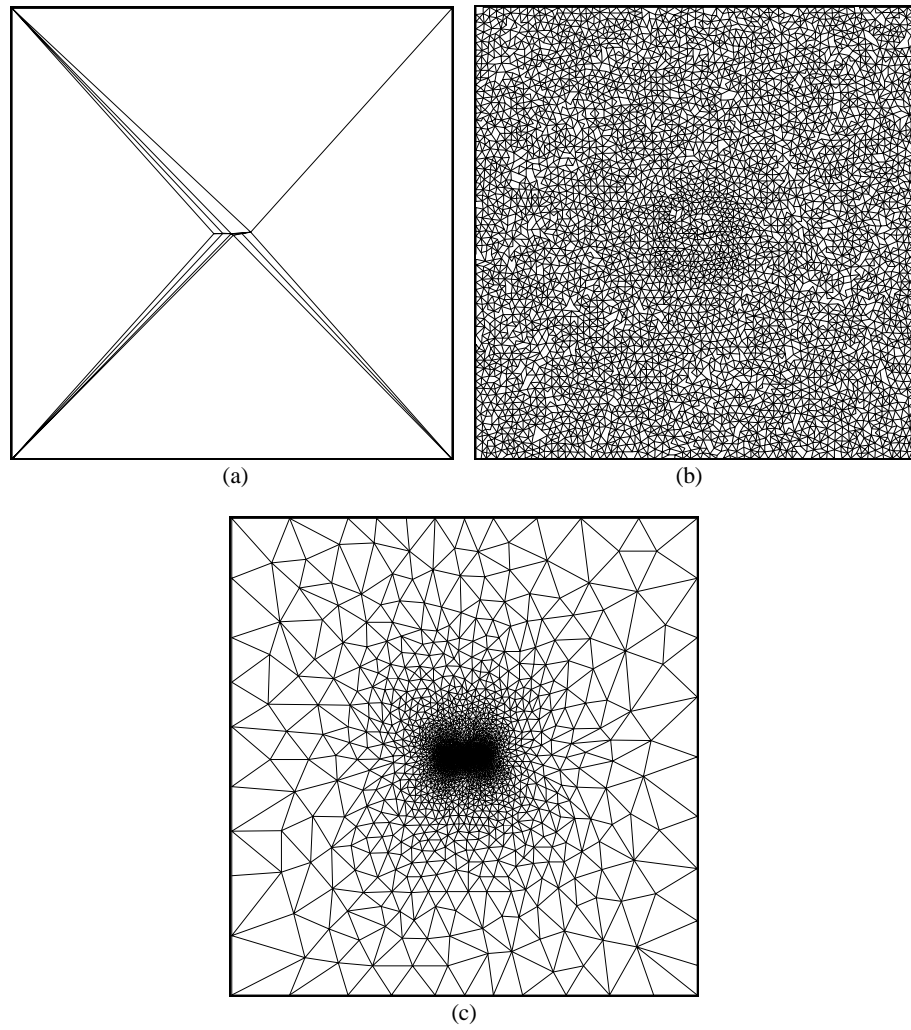


FIGURE 6. (a) First mesh. (b) Uniform mesh with the desired number of elements. (c) Adaptively optimized mesh, 4917 nodes, 9589 elements.

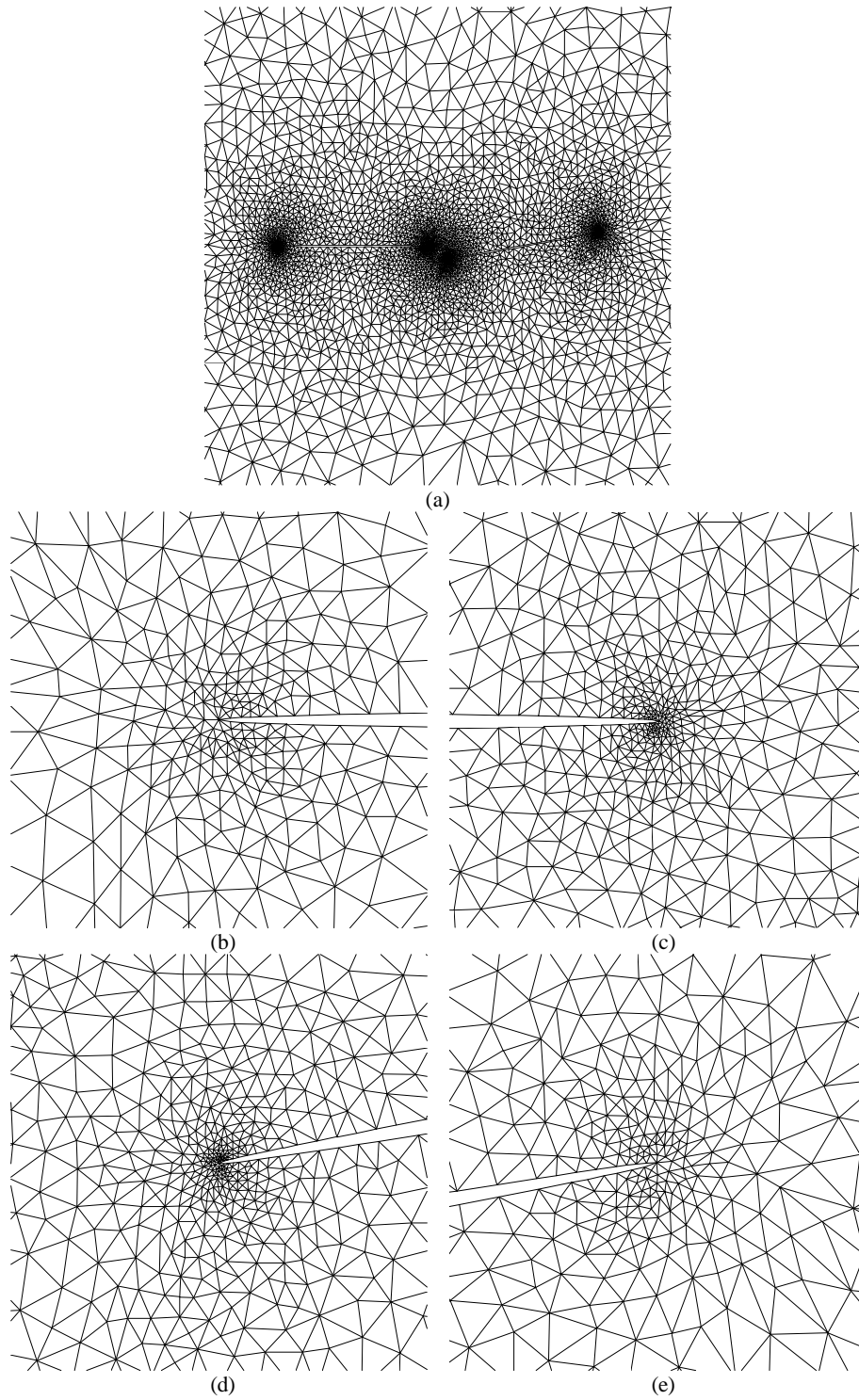


FIGURE 7. *Details of the optimized mesh.*

domains. By suitable meshes we understand meshes consisting of elements that exhibit both the appropriate shape and the appropriate size to avoid excessive numerical inaccuracies.

We have presented a full mesh optimization procedure that deals with both shape and size within a unified methodology. It was first shown to

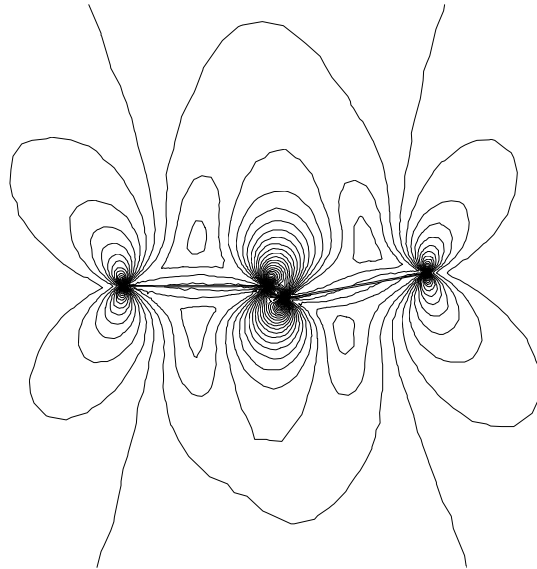


FIGURE 8. Isocontours of the von Mises' equivalent stress computed on the optimized mesh.

work for 3D meshes, as it eliminates the few badly-distorted elements that usually persist after the initial generation step (it was shown, in addition, that shape distortions can have an impact on accuracy of the same order as that of mesh size). Later, as the method is valid for any number of space dimensions but our implementation is limited to 2D, it was shown that, in fact, the optimizer can be used to generate adaptively refined meshes starting from "almost nothing" (i.e., a mesh as simple as that of Fig. 6(a)).

In this way, it is now possible to solve problems in elasticity or other fields that were not tractable a few years ago, and mesh optimization techniques have proved to be a valuable help for this purpose. Besides the full implementation handling 3D geometric abstractions, that is under way, an interesting extension concerns time-dependent problems. If the regions deserving refinement evolve with time, it is not obvious that the optimization algorithm as proposed above will be able to follow this evolution with a fast enough convergence rate to make it practical.

REFERENCES

- [1] J. PERAIRE, K. MORGAN AND J. PEIRO, *Unstructured mesh methods for CFD*, I.C. Aero Rep. 90-04, 1990, Imperial College, UK.
- [2] J. PEIRO, Ph. D. Thesis, Dept. Civil Eng., Univ. College of Swansea, 1989.
- [3] D. L. MARCUM AND N. P. WEATHERILL, *Unstructured grid generation using iterative point insertion and local reconnection*, AIAA paper 94-1926, 12th AIAA Appl. Aerodyn. Conf., Colorado Springs, USA, 1994.
- [4] D. L. MARCUM, *Generation of high quality unstructured grids for computational field simulation*, presented at VI Int. Symp. on Comp. Fluid Dyn., Lake Tahoe, USA, 4-8 September 1995. Also *Control of point placement and connectivity in unstructured grid generation procedures*, presented at IX Int. Conf. Finite Elements in Fluids, Venice, Italy, 15-21 October 1995.
- [5] E. A. DARI, Ph. D. Thesis, Instituto Balseiro, 1994.
- [6] E. A. DARI AND G. C. BUSCAGLIA, *Mesh optimization: How to obtain good unstructured 3D finite element meshes with not-so-good mesh generators*, Struct. Optim. 8, pp 181-188 (1994).

- [7] P. D. ZAVATTIERI, E. A. DARI AND G. C. BUSCAGLIA, *Optimization strategies in unstructured mesh generation*, Int. J. Numer. Meth. in Eng., to appear.
- [8] S. R. KENNON AND G. S. DULIKRACH, *Generation of computational grids using optimization*, AIAA J. 24, pp 1069-1073 (1986).
- [9] T. TEZDUYAR, S. ALIABADI, M. BEHR, A. JOHNSON AND S. MITTAL, *Parallel finite element computation of 3D flows - Computation of moving boundaries and interfaces, and mesh update strategies*, Paper 93-042, Army High Performance Computing Research Center, Univ. Minnesota, USA, 1993.
- [10] A. G. STAMATIS AND K. D. PAPALIOU, *An unstructured grid optimization method*, in Finite Elements in Fluids, New trends and applications (Ed. K. Morgan *et al*), Pineridge Press, 1993, pp 676-685.
- [11] J. CABELLO, R. LOHNER AND O.-P. JACQUOTTE, *Recent improvements of a variational method for the optimization of directionally stretched unstructured meshes*, in Numerical Grid Generation in Comp. Fluid Dynamics and Related Fields (Ed. N. P. Weatherill *et al*), Pineridge Press, 1994, pp 163-175.
- [12] P. L. GEORGE AND E. SEVENO, *The advancing-front mesh generation method revisited*, Int. J. Numer. Meth. in Eng. 37, pp 3605-3619 (1994).
- [13] H. ZHANG AND J.-Y. TREPANIER, *An algorithm for the optimization of directionally stretched triangulations*, Int. J. Numer. Meth. in Eng. 37, pp 1481-1497 (1994).
- [14] E. BRIERE DE L'ISLE AND P. L. GEORGE, *Optimization of tetrahedral meshes*, in Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations (Ed. I. Babuska *et al*), IMA Vol. 75, Springer-Verlag, 1993, pp 97-128.
- [15] T. COUPEZ, *A mesh improvement method for 3D automatic remeshing*, in Numerical Grid Generation in Comp. Fluid Dynamics and Related Fields (Ed. N. P. Weatherill *et al*), Pineridge Press, Swansea, 1994, pp 615-625.
- [16] G. C. BUSCAGLIA AND E. A. DARI, *Adaptive anisotropic mesh optimization: Automatic construction of solution-adapted finite element meshes*, submitted (1996).
- [17] J. PERAIRE, J. PEIRO AND K. MORGAN, *Adaptive remeshing for three-dimensional compressible flow computations*, J. Comp. Phys. 103, pp 269-285 (1992).
- [18] J. DOMPIERRE, M.-G. VALLET, M. FORTIN, W. G. HABASHI, D. AIT-ALI-YAHIA, S. BOIVIN, Y. BOURGAULT AND A. TAM, *Edge-based mesh adaptation for CFD*, Proc. Conf. Numer. Methods for the Euler and Navier-Stokes Eq., Montréal, 1995, pp 265-299.
- [19] N. A. GOLIAS AND T. D. TSIBOUKIS, *An approach to refining three-dimensional tetrahedral meshes based on Delaunay transformations*, Int. J. Numer. Meth. in Eng. 37, pp 793-812 (1994).
- [20] E. POLAK, *On the mathematical foundations of nondifferentiable optimization in engineering analysis*, SIAM Rev. 29, pp 21-89 (1987).
- [21] C. LEMARECHAL, *Nondifferentiable optimization*, in Optimization (G. L. Nemhauser *et al*, Eds.), Handbooks in Oper. Res. and Manag. Sci., Vol. 1, North Holland, 1989, pp 529-572.
- [22] R. FLETCHER, *Practical Methods of Optimization*, Vol. 2, Constrained Optimization, J. Wiley & Sons, 1981.
- [23] E. A. DARI AND G. C. BUSCAGLIA, *Topics on finite element meshes for problems with moving boundaries*, in Finite Elements in Fluids, New trends and applications (Ed. K. Morgan *et al*), Pineridge Press, 1993, pp 726-735.
- [24] A. C. MARINO, E. SAVINO AND S. HARRIAGUE, *Thermomechanical description of a nuclear pin, BACO code version 2.20*, in Trans. 13th Int. Conf. Struct. Mech. Reactor Technol. (SMiRT 13), Porto Alegre, Brazil, 1995, pp 811-819.
- [25] O. C. ZIENKIEWICZ AND J. Z. ZHU, *A simple error estimator and adaptive procedure for practical engineering analysis*, Int. J. Numer. Meth. in Eng. 24, pp 337-357 (1987).
- [26] D. L. ARNICA, E. A. DARI AND C. PADRA, *Selection strategies in adaptive refinement*, unpublished (1996).
- [27] M. KACHANOV, *Elastic solids with many cracks and related problems*, in Adv. in Appl. Math., Vol. 30 (J. Hutchinson and T. Wu, Eds.), Academic Press, 1993, pp 259-445. See Section IV.C.

Acknowledgements: The kind invitation made by the Organizing Committee and by Centre Jacques Cartier to attend this Conference is gratefully acknowledged. This work was partially supported by Consejo Nacional de

Investigaciones Científicas y Técnicas (CONICET) and by Cooperativa de Electricidad Bariloche. We thank R. Rodríguez for calling our attention to [18] and for proof reading, and J. Baranger, R. Codina, C. Padra, and M. Fortin for fruitful discussions.

INSTITUTO BALSEIRO AND CENTRO ATÓMICO BARILOCHE, 8400 BARILOCHE, ARGENTINA.

E-mail address: `gustavo@cab.cnea.edu.ar`