INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2000; **00**:1–6 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

Simultaneous untangling and smoothing of moving grids

Ezequiel J. López*, Norberto M. Nigro and Mario A. Storti

Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC),
INTEC-CONICET-UNL, Güemes 3450, 3000 Santa Fe, ARGENTINA

SUMMARY

In this work, a technique for simultaneous untangling and smoothing of meshes is presented. It is based on an extension of an earlier mesh smoothing strategy developed to solve the computational mesh dynamics stage in fluid-structure interaction problems. In moving grid problems, mesh untangling is necessary when element inversion happens as a result of a moving domain boundary. The smoothing strategy, formerly published by the authors, is defined in terms of the minimization of a functional associated with the mesh distortion by using a geometric indicator of the element quality. This functional becomes discontinuous when an element has null volume, making it impossible to obtain a valid mesh from an invalid one. To circumvent this drawback, the functional proposed is transformed in order to guarantee its continuity for the whole space of nodal coordinates, thus achieving the untangling technique. This regularization depends on one parameter, making the recovery of the original functional possible as this parameter tends to zero. This feature is very important, consequently, it is necessary to regularize the functional in order to make the mesh valid, then, it is advisable to use the original functional to make the smoothing optimal. Finally, the simultaneous untangling and smoothing technique is applied to several test cases, including 2D and 3D meshes with simplicial elements. As an additional example, the application of this technique to a mesh generation case is presented.

KEY WORDS: Mesh untangling, Mesh smoothing, Moving meshes, ALE.

1. Introduction

Several scientific and industrial applications of computational mechanics problems involve moving meshes. Examples of problems related to moving meshes include free surfaces, two-fluid interfaces, fluid-object and fluid-structure interactions, and moving mechanical components.

In computation of fluid problems with moving boundaries and interfaces, either an interface-tracking or interface-capturing technique can be used, depending on the complexity of the interface as well as on other aspects of the problem. An interface-tracking technique requires meshes that “track” the interfaces, then, the mesh needs to be updated as the flow evolves.

*Correspondence to: Ezequiel J. López, Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), INTEC-CONICET-UNL, Güemes 3450, 3000 Santa Fe, ARGENTINA

Received 15 November 2006

Besides, in an interface-capturing technique for two-fluid flows, the computations are based on fixed spatial domains, where an interface function, marking the location of the interface, needs to be computed to "capture" the interface within the resolution of the finite element mesh covering the area where the interface is [1, 2].

In fluid-structure interaction (FSI) problems, one of the most popular interface tracking techniques is the Arbitrary Lagrangian Eulerian (ALE) formulation [3, 4, 5] In this case, the mesh is updated at every time step due to the motion of the domain boundary, causing mesh quality deterioration and, in some situations, generating an invalid mesh where any of the elements in the grid is inverted. It is well known that poor quality elements have strong influence on stability, convergence and accuracy of the numerical methods used. In Computational Mechanics, the strategies developed to solve the mesh motion are grouped in a special topic named CMD (Computational Mesh Dynamics). Its importance may be assessed simply inspecting the current bibliography [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17].

In this work, the main interest is the resolution of FSI problems. These type of applications require the solution of two problems, coupled in the physical domain: the flow problem dictates the generalized forces acting over the structure, while the structural problem determines the geometry variation caused by its deformation and also its translational and rotational motion. Nowadays, efficient solution of FSI problems with large displacement related to the boundary is still a challenging problem in Computational Mechanics. Mainly, these problems are solved in a partitioned fashion. First, the fluid solution is obtained using a CFD (Computational Fluid Dynamics) code with structural displacements estimated by a predictor stage. The pressure and possibly the viscous stresses are stored at each point of the fluid-structure interface. After that, these time-dependent loads are transferred to a CSD (Computational Structure Dynamics) code, which finds the deformation of the structure in time [18, 19]. Finally, a new fluid solution is found with the updated position of the structure. Although the moving mesh is only an artificial field in the coupled three-field FSI problem (CFD+CSD+CMD), it strongly influences the performance, robustness and accuracy of the overall approach.

In general, the CFD stage consumes most of the CPU time, adding restrictions to the time step, specially in regions where the mesh is refined. Besides, the boundary motion may also cause the inversion of some elements, giving rise to a forced reduction of the time step due to CMD reasons. This is not desirable because the CMD problem is artificially introduced to follow the moving boundary. In order to avoid this limitation, it is advisable to introduce the possibility of fixing the inverted elements instead of reducing the time step. A mesh untangling strategy makes this solution possible.

This paper contributes in that sense. The untangling makes the CMD implementation more robust, showing no influence on the time step selection. Furthermore, the smoothing capability is necessary to optimize the mesh quality, improving the solution accuracy.

Mesh smoothing methods adjust the positions of the nodes in the grid while preserving its topology (graph connectivity). Most of them are based on local algorithms, i.e. the free nodes are relocated one by one iteratively, keeping the remainder fixed until the convergence is reached. Laplacian smoothing is the most commonly used strategy due to its low computational cost and its simple implementation. This method moves the internal nodes towards the geometrical center of their neighbors. However, this method does not always work unless a valid mesh is guaranteed. Moreover, in cases where convergence is not reached, the final mesh may depend on the nodal sequence ordering. This is frequently found in local smoothing techniques. Other kind of smoothing related to the Laplacian technique is Winslow smoothing, which is more robust in terms of avoiding the inversion of the elements in the mesh. This method is based on logical variables with the requirement that these variables are harmonic functions [20]. Originally, this technique was presented for structured meshes, being lately extended to unstructured meshes by Knupp [21]. There are several smoothing methods based on the resolution of optimization problems having the common goal of improving the mesh quality in terms of a quality indicator [22, 23, 24, 25, 26, 27, 28]. The main disadvantage of such methods is their computational cost. Depending on the way in which the system is solved, there are local and global methods. While the global methods update the nodal position simultaneously for the whole set of nodes, the local counterpart applies its methodology over each subset of nodes until the whole set of nodes is updated. Local methods are to global ones as explicit schemes are to implicit schemes for the resolution of differential equations with preponderant diffusive character. Furthermore, there is no guarantee that a solution for a global strategy may always be reached for a local method.

The smoothing technique may encounter situations where the mesh is invalid, then an untangling methodology should be used. These methods are normally based on the element volume [29] and, in general, both procedures, smoothing and untangling, are treated separately. This tedious task may be incorporated for mesh generation purposes where the user is interactively looking for a good quality mesh. Thinking of FSI problems, the CMD method should have the capability of solving the mesh motion even though inverted elements were found, guaranteeing a smooth mesh at each time. Therefore, a simultaneous procedure of smoothing and untangling is preferable [30, 31, 32, 33]. In this paper, a simultaneous mesh untangling and smoothing technique is proposed, based on the optimization of the grid quality. The strategy used arises from the regularization of a previously presented functional [28], which is then applied to solve the mesh dynamics in FSI problems. The optimization problem is solved in a global way in order to avoid the drawbacks that local methods present.

The proposed method can be useful for mesh generation. In this case, the topology is generated in an auxiliary domain in which the mesh may be generated in a structured way. Then, the boundary nodes in that mesh are relocated in the real boundary. This sharp movement of the boundary nodes is similar to the situation faced in mesh dynamics. Using the untangling and smoothing technique herein presented, a valid mesh is generated.

In this work, the mesh smoothing method formerly published [28] is briefly presented. The modifications introduced to avoid the relaxation of the initial mesh, the functional regularization and the strategy used for solving the optimization problem are discussed. Finally, several results for CMD in 2D and 3D are included and a 2D mesh generation application of this strategy is shown. At the end, some conclusions and future work are included.

2. Original mesh smoothing strategy

In a previous work, the authors have presented a mesh smoothing technique useful for CMD problems [28]. The technique is based on an optimization problem solved in a global way, where the functional indicates the mesh distortion. Such a functional was defined in the following way

$$F(\mathbf{x}) = \sum_e F_e(\mathbf{x}) \tag{1}$$

with

$$F_e(\mathbf{x}) = q_e^n \tag{2}$$

being q_e some element quality indicator and n a negative integer.

Written in this way, the functional allows its application for any kind of elements properly defining the quality indicator. In this work, the following geometric quality indicators are proposed:

- Simplicial elements:

$$q = Cq_S \tag{3}$$

where

$$q_S = \frac{V}{\sum_j l_j^{nd}} \tag{4}$$

being l_j the length of the $j-$edge, V the volume and nd the number of spatial dimensions. $C = 4\sqrt{3}$ for triangles and $C = 36\sqrt{2}$ for tetrahedra.

- Non-simplicial elements:

$$q = C\prod_{i=1}^{M} q_{S,i} \tag{5}$$

where C is a normalization constant such that $0 < q \leq 1$, M is the total number of possible subdivision of the element in simplicial ones and q_S is given by equation (4).

Due to the fact that the quality indicator for non-simplicial elements is based on those defined for simplicial ones, only the simplicial element case is analyzed.

The proposed functional is continuous if $q_e \neq 0$ for all the elements in the mesh, but $F_e(\mathbf{x})$ tends to infinity when q_e tends to zero. This last situation happens when there is at least one element in the mesh with $V_e \to 0$ and $\sum_j l_j^{nd}$ is bounded below (i.e. the simplex is not collapsed to a single point). Therefore, the application of this technique is restricted only to valid meshes, since infinite barriers arise when the element volume tends to zero, making it impossible to recover a valid mesh starting from an invalid one. In FSI problems, this limitation is sometimes by-passed decreasing the time step size and, thus, avoiding the tangling of the grid caused by the motion of the boundary. However, the computational cost suffers a large increase, specially if some clustering of nodes is used close to the moving boundary to capture fluid dynamics details like boundary layers. It is in this sense that the CMD strategy, enhanced with simultaneous untangling and smoothing, is useful, providing a way to recover a valid mesh despite starting with an invalid one.

Another approach to solve the mesh dynamics in FSI problems is to move the mesh for as long as it is possible, and remesh when it is tangled or too distorted. It is possible to go back to the previous time step (or a few time steps before the mesh is tangled) to generate a (fully or partially) new mesh, project the solution from the old mesh to the new one, and keep on moving the new mesh. The projection introduces an error over the solution and it must be properly designed in order to conserve some physical quantities. Then, it is remarkable that one of the main disadvantages of remeshing lies on its inherent non-compliance of some conservation laws for the solution of FSI problems. In some sense, simultaneous mesh untangling and smoothing is a procedure that allows the minimization of the amount of remeshing that can be done during a simulation. With this feature, the error introduced by the projection needed for remeshing may be bounded to a minimum.

2.1. Avoiding the relaxation of the initial mesh

Pseudo-elastic CMD strategies have the property that they do not move the initial mesh unless the domain boundary is deformed. This may be justified by using elastic energy minimization arguments.

This property is not shared by the proposed functional (1), because sometimes the initial mesh introduced by the user is not the optimal mesh with respect to this functional. Consider, for instance, the structured mesh *M1* shown in figure 1. The mesh is composed of 200 triangular elements. Even if the mesh has a good quality, the optimization strategy tends to bring each element to a regular (equilateral) shape, so that after a relaxation stage the mesh *M3* is obtained. In this case, during the relaxation process the nodes on sides AB, CD are fixed, whereas those on BC, AD are left to slide on the horizontal direction. As a consequence of the optimization problem, the elements near vertexes A and C tend to shrink, whereas those near vertexes B and D tend to grow. This effect is caused by the particular way in which the squares have been split up into triangles. Note how the elements tend to reach the equilateral shape in the relaxed mesh. After the mesh has relaxed, subsequent displacements of the boundary nodes produce displacement of the internal nodes, as described before.

This initial *"relaxation"* stage may or may not be desirable. If the initial mesh has bad quality, then this stage may tend to get a new better mesh. However, if the initial mesh has some *ad-hoc* refinement, then it is possible that the relaxation stage will revert this refinement. Consider for instance mesh *M2* in figure 1, which has a refinement towards side AB in such a way that the horizontal spacing near CD is 3.5 times larger than the one at AB. As a result of the relaxation process, the relaxed mesh *M3* is reached. The resulting relaxed mesh depends only on the topology of the mesh and on the constraints on the boundary nodes, but not on the initial position of the internal nodes. In fact, both meshes *M2* and *M1* (with and without refinement) produce the same final mesh *M3* after relaxation.

The functional can be easily modified in order to keep the initial refinement. First, note that for simplicial elements there is a unique linear transformation $(\mathbf{x}_0, \mathbf{T})$ that transforms the coordinates $\{\mathbf{x}_{\mathrm{reg},j}\}$ of the *regular* element (i.e. equilateral triangle in 2D, regular tetrahedron in 3D) to the actual element coordinates $\{\mathbf{x}_j\}$

$$\mathbf{x}_j = \mathbf{x}_0 + \mathbf{T}\,\mathbf{x}_{\mathrm{reg},j} \tag{6}$$

It is easy to see that the functional can be put as a function of the transformation matrix $\mathbf{T}$

$$F_e = g(\mathbf{T}) \tag{7}$$

This fact can be seen because the functional can be computed by taking the nodal coordinates of the regular elements, applying the transformation and finally computing the side lengths, volume, and the functional. All these computations are encapsulated in the function $g(\,\cdot\,)$. Of course, the functional does not depend on the translation $\mathbf{x}_0$. In fact, it does only depend on the metric of the transformation $\mathbf{T}^T\mathbf{T}$, because it is independent of rotations. However, for the analysis that follows, it is needed only to accept that it depends on the transformation matrix, as reflected in (7). By construction, g has a minimum when $\mathbf{T} = c\,\mathbf{O}$, with c as a scaling factor and $\mathbf{O}$ as an orthogonal matrix, since in this case the current element is similar to the regular one.

The purpose is to modify the functional so that the optimal element shape for F_e is not the regular element shape, or else the shape of some reference element with coordinates $\{\mathbf{x}_{\mathrm{ref},j}\}$

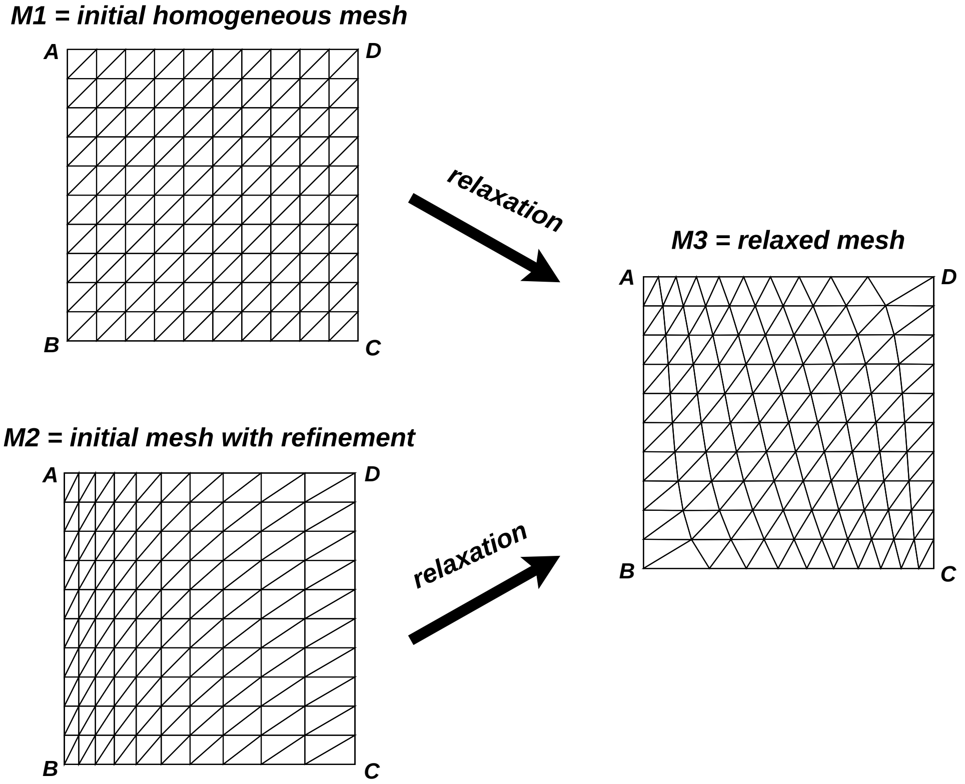
M1 = initial homogeneous mesh
A
D
B
C
relaxation
M3 = relaxed mesh
A
D
B
C
M2 = initial mesh with refinement
A
D
B
C
relaxation

Figure 1. Relaxation of meshes

(see figure 2). It is easy to see that this can be done by considering the transformation from the reference element to the current element

$$F_e = g(\mathbf{T}\,\mathbf{T}'^{-1}) \tag{8}$$

where $\mathbf{T}'$ transforms the regular element to the reference element. For instance, as mentioned above, a minimum is reached when $\mathbf{T}\,\mathbf{T}'^{-1} = c\,\mathbf{O}$, i.e. when the current element is similar in shape to the reference element.

Note that, this modification can be simply introduced by computing transformations $\mathbf{T}, \mathbf{T}'$ and then computing the functional with the coordinates $\mathbf{x}'_j = \mathbf{T}\,\mathbf{T}'^{-1}\mathbf{x}_{\mathrm{ref},j}$.

An example can be seen in figure 3. The original mesh on the right has a refinement ratio of 1:10 near the AB side. Then, it is deformed at the AB side with a ramp with amplitude 0.2 (resulting in the mesh shown on the left). Note that as no initial relaxation is produced, the final mesh still has the refinement towards the AB side.

Computations of the analytical jacobians are also straightforward. The jacobians with respect to the $\mathbf{x}'_j$ are computed in the standard way, and then they are composed with the

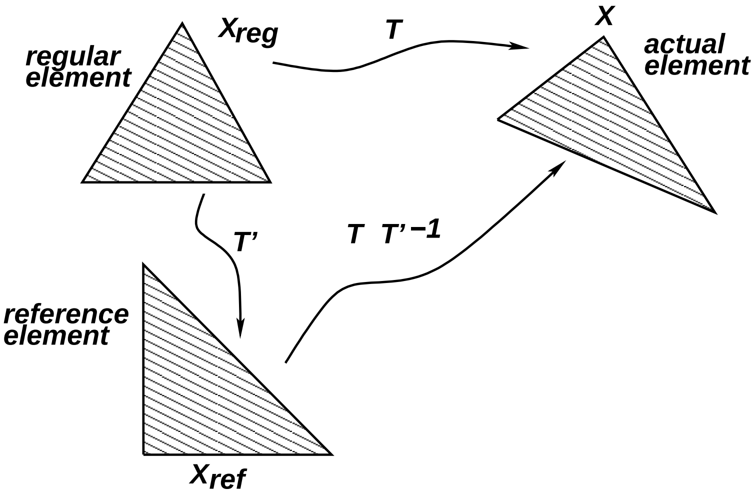

Figure 2. Compensation for initial deformation in reference mesh

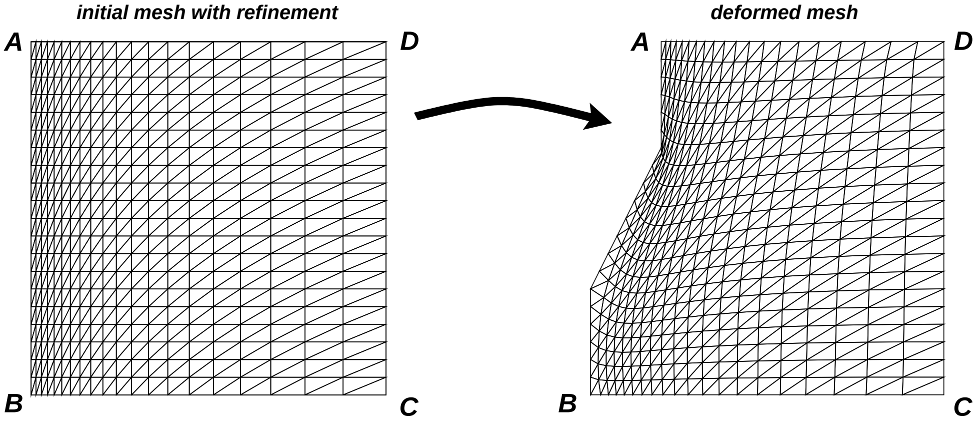

Figure 3. Deformation of mesh with surface refinement

jacobian

$$\frac{\partial \mathbf{x}'_j}{\partial \mathbf{x}_j} = \mathbf{T}\,\mathbf{T}'^{-1}\,\mathbf{T}^{-1} \tag{9}$$

3. Functional regularization

Being the main goal of the present work to include the untangling capability to the formerly published mesh smoothing technique [28], the functional was modified using an idea proposed in [32], that makes the functional continuous for all element volume values. This modification consists in replacing V in equation (4) by the function

$$h(V) = \frac{1}{2}(V + \sqrt{V^2 + 4\delta^2}) \tag{10}$$

This is a strictly increasing function of the volume and it is also a positive function for all V (see figure 4). The parameter δ represents the value of the function for a null volume.

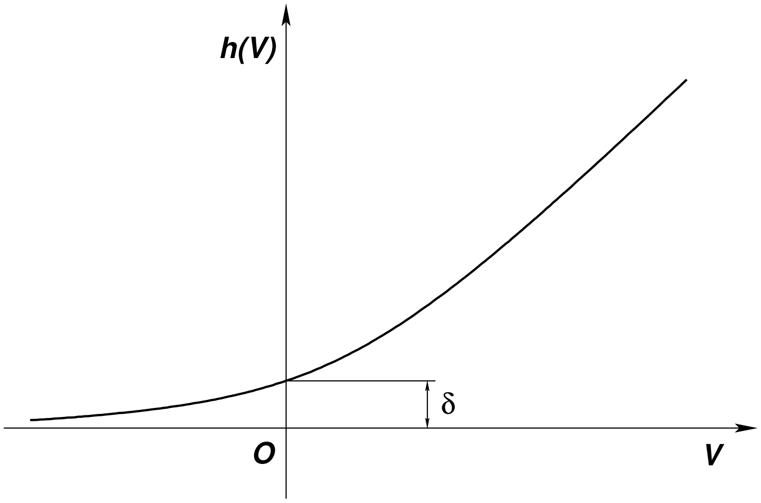

Figure 4. Function $h(V)$.

Then, the modified functional is written as

$$F_e^*(\mathbf{x}) = q_e^{*n}, \quad \text{with} \quad q_e^* = C\frac{h(V)}{\sum_j l_j^{nd}} \tag{11}$$

The dependence of $h(V)$ with the parameter δ is such that

$$\lim_{\delta \to 0} h(V) = \begin{cases} V & \text{if} \quad V \geq 0 \\ 0 & \text{if} \quad V < 0 \end{cases}$$

Therefore, when the parameter δ tends to zero, the modified functional tends to the original one for $V \geq 0$, and also, the modified optimal solution tends to the original one. In the limit when $\delta \to 0$, $F^*(\mathbf{x}) \to F(\mathbf{x})$ pointwise.

4. Solution strategy

In this paper, a global strategy for solving the system of equations is applied, taking as variables the coordinates of the free nodes in the mesh, simultaneously. Assuming that a valid mesh exists for the given topology and boundary position, the goal is to look for the position of the nodes in order to make the original functional optimal or to find an approximate solution close to the original one. This can be done by decreasing the value of the parameter δ below a prefixed tolerance. According to numerical examples, the lower the parameter δ, the slower the convergence rate of the optimization algorithm, without guarantee of a final convergence. Moreover, if δ is not small enough the optimal mesh may be invalid, being even worse for high relative domain deformations. Therefore, two main problems arise:

- finding the decreasing sequence of the parameter δ to ensure the final convergence to a valid mesh.
- finding the initial value for the parameter δ.

To determine an equation that allows the decrease the parameter δ, we include it inside the functional as a new global variable only for theoretical purposes; i.e. considering $F^* = F^*(\mathbf{x}, \delta)$. Due to the functionality of $h(V)$ with the parameter δ, the value of the optimal regularized functional increases when δ decreases ($\delta \neq 0$). If the optimization problem is posed in terms of the variables $(\mathbf{x}, \delta)$ and using a Newton-like solver, the problem is written as follows:

$$\begin{bmatrix} \frac{\partial^2 F^*}{\partial \mathbf{x}^2} & \frac{\partial^2 F^*}{\partial \mathbf{x}\partial\delta} \\ \frac{\partial^2 F^*}{\partial\delta\partial \mathbf{x}} & \frac{\partial^2 F^*}{\partial\delta^2} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\delta \end{bmatrix} = - \begin{bmatrix} \frac{\partial F^*}{\partial \mathbf{x}} \\ \frac{\partial F^*}{\partial\delta} \end{bmatrix}$$

Rewriting the above equation in the following way

$$\begin{aligned} \frac{\partial^2 F^*}{\partial \mathbf{x}^2}\Delta\mathbf{x} + \frac{\partial^2 F^*}{\partial \mathbf{x}\partial\delta}\Delta\delta &= -\frac{\partial F^*}{\partial \mathbf{x}} \\ \frac{\partial^2 F^*}{\partial\delta\partial \mathbf{x}}\Delta\mathbf{x} + \frac{\partial^2 F^*}{\partial\delta^2}\Delta\delta &= -\frac{\partial F^*}{\partial\delta} \end{aligned} \tag{12}$$

it is observed that this system may be solved in an uncoupled way if the parameter δ is kept fixed for the first equation. This is equivalent to solve the system (13) using the block Gauss-Seidel method. Thus, the variable increments $\Delta\mathbf{x}$ and $\Delta\delta$ are written as follows:

$$\begin{aligned} \Delta\mathbf{x} &= -\left(\frac{\partial^2 F^*}{\partial \mathbf{x}^2}\right)^{-1} \frac{\partial F^*}{\partial \mathbf{x}} \\ \Delta\delta &= -\frac{\left(\frac{\partial F^*}{\partial\delta} + \frac{\partial^2 F^*}{\partial\delta\partial \mathbf{x}}\Delta\mathbf{x}\right)}{\frac{\partial^2 F^*}{\partial\delta^2}} \end{aligned} \tag{13}$$

The expression for $\Delta\delta$ in (13) is adopted as the maximum value to reduce δ, thus limiting the decreasing rate of the sequence to avoid losing convergence.

Therefore, the updated δ in the iteration k is defined in the following way:

$$\delta^k = \max(\delta^{k-1} - \alpha|\Delta\delta|, \beta\delta^{k-1}) \tag{14}$$

being α and β constants lower than one.

It was found that the off-diagonal terms in the element-wise matrix have a strong influence on the convergence of this optimization method. In the untangling stage, it is advisable to relax these off-diagonal terms to make the matrix more diagonal-dominant. However, in the smoothing stage these terms should be restored to take advantage of the convergence rate of full Newton schemes. Here, the relaxation parameter for these off-diagonal terms ($\gamma \leq 1$)

may be constant or variable with respect to the iterations. For example, for a 2D case the element-wise matrix is modified in the following way:

$$\mathbf{K^e} = \begin{bmatrix} \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_1^2}\right)_e & \gamma \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_1 \partial \mathbf{x}_2}\right)_e \\ \gamma \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_2 \partial \mathbf{x}_1}\right)_e & \left(\frac{\partial^2 F^*}{\partial \mathbf{x}_2^2}\right)_e \end{bmatrix}$$

The problem is solved by the Newton-Raphson method with Armijo inexact line search [34]. At each iteration δ is diminished only if the line search strategy gives a unit step as result.

4.1. Differential predictor

The optimization strategy means that at each time step the unknown node positions are obtained by solving a minimization problem. The mesh coordinates vector $\mathbf{x}$ is composed of nodes at the boundary $\mathbf{x}_b$ and the internal nodes $\mathbf{x}_{\text{int}}$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_{\text{int}} \end{bmatrix} \tag{15}$$

At each time step, the minimization problem consists in finding the $\mathbf{x}$ that minimizes the functional $F(\mathbf{x})$. Due to the fact that some components of $\mathbf{x}$ (those in $\mathbf{x}_b$) are fixed by the boundary conditions

$$\mathbf{x}_{\text{int}}^n = \underset{\tilde{\mathbf{x}}_{\text{int}}}{\operatorname{argmin}} F\left(\begin{bmatrix} \mathbf{x}_b^n \\ \tilde{\mathbf{x}}_{\text{int}} \end{bmatrix}\right) \tag{16}$$

The recurrence formula from the Newton-Raphson strategy is

$$\mathbf{x}_{\text{int}}^{n,k+1} = \mathbf{x}_{\text{int}}^{n,k} - (\mathbf{K}^k)^{-1}\mathbf{R}^k \tag{17}$$

where

$$\begin{aligned} \mathbf{R} &= \frac{\partial F}{\partial \mathbf{x}_{\text{int}}} \\ \mathbf{K} &= \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}} \end{aligned} \tag{18}$$

This generates a sequence $\mathbf{x}_{\text{int}}^{n,k}$ that, if it converges, it gives the solution for the optimization problem

$$\lim_{k\to\infty} \mathbf{x}_{\text{int}}^{n,k} = \mathbf{x}_{\text{int}}^n \tag{19}$$

The simplest choice for the initial value $\mathbf{x}_{\text{int}}^{n,0}$ is to take the unknown vector at the previous time step

$$\mathbf{x}_{\text{int}}^{n,0} = \mathbf{x}_{\text{int}}^{n-1,\infty} \tag{20}$$

However, this has the drawback that, if the elements near the moving boundary are small, then the initial combination $[\mathbf{x}_{\text{int}}^{n-1,\infty}, \mathbf{x}_b^n]$ may lead to invalid elements, even for small time steps. In fact, the time step is limited by the element size at the wall, and the limit time step of the problem of the moving mesh diminishes with mesh refinement.

To avoid this, a linear predictor for the initial mesh is performed. If the solution $\mathbf{x}_{\text{int}}(t)$ for each t in the range $t^{n-1} \leq t \leq t^n$ is considered, then

$$\mathbf{R}(\mathbf{x}_{\text{int}}(t), \mathbf{x}_b(t)) = 0 \tag{21}$$

Taking derivatives with respect to time and making an evaluation at $t = t^{n-1}$

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}}\right)_{t^{n-1}} \dot{\mathbf{x}}_{\text{int}}(t^{n-1}) + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_b}\right)_{t^{n-1}} \dot{\mathbf{x}}_b(t^{n-1}) = 0 \tag{22}$$

then the Newton-Raphson sequence can be initialized with the extrapolation

$$\mathbf{x}_{\text{int}}^{n,0} = \mathbf{x}_{\text{int}}^{n-1,\infty} + \Delta t\, \dot{\mathbf{x}}_{\text{int}}(t^{n-1}) \tag{23}$$

For instance, consider a 1D problem with a homogeneous mesh of N linear elements in the interval $[0, 1]$. The right boundary is fixed and the left boundary moves to the right with velocity 1. With the standard initialization strategy, the limit time step is initially $\Delta t_{\text{CMD}} = h = 1/N$, since a larger time step will cause the left boundary to pass over the position of the first internal node (initially at $x = h$). Besides, with the differential predictor, the limit time step is $\Delta t_{\text{CMD}} = 1$, since, in fact, the differential predictor gives the optimal solution, and the subsequent Newton-Raphson iteration is not needed. It has been verified through numerical experiments that with differential predictor the limiting time step Δt_{CMD} is independent of the mesh refinement.

5. Numerical results

In this section, the numerical results for several test examples are presented. These examples show the capability of the proposed strategy for different deformations of the boundary, from medium (50 %) to high deformations (99 %) carried out in only one step. In all these cases, initially inverted elements were found. 2D and 3D mesh dynamics problems are presented, as well as a 2D mesh generation problem . In the whole set of test cases, the following convergence criteria had been applied:

- Valid mesh.
- For the iteration k,

$$\frac{|q_{\min}^k - q_{\min}^{k-1}|}{q_{\min}^k} < \epsilon_q$$

being $q_{\min} = \min_e q_e$ and ϵ_q a prefixed tolerance.

The relaxation coefficient for the Hessian matrix was chosen as $\gamma = 0.5$ for the untangling stage and $\gamma = 1$ for the smoothing stage. When the mesh is initially invalid, the starting value of the parameter δ is chosen according to the following criterion based on the minimum volume ($V_{\min} = \min_e V_e$). Due to the fact that $h(V)$ is a strictly increasing function, then

$$h_{\min} = h(V_{\min}) = \frac{1}{2}\left(V_{\min} + \sqrt{V_{\min}^2 + 4\delta^2}\right) \tag{24}$$

Adopting $h^*_{\min} = h_{\min}/\delta$ as an user-defined parameter and getting δ from the last equation, the following criterion to initialize δ arises:

$$\delta = \begin{cases} \frac{h^*_{\min} V_{\min}}{{h^*_{\min}}^2 - 1} + \epsilon_\delta & \text{if} \quad V \leq 0 \\ 0 & \text{if} \quad V > 0 \end{cases}$$

where $\epsilon_\delta > 0$ is the minimum value given to the initial δ such that $\delta > 0$ when $V_{min} = 0$.

In the whole set of numerical examples, the following set of parameter values were used: $n = -1$, $\alpha = 1$, $\beta = 0.1$, $\epsilon_q = 0.01$, $h^*_{\min} = 0.75$ and $\epsilon_\delta = 1 \times 10^{-6}$. In these tests, the reference element used was the regular element.

In order to compare the performance of the proposed technique with the smoothing one, some tests were solved using both methods. The time required for the computation is compared for both techniques. When the smoothing strategy (S) was used, Δt and the law of movement of the boundary were chosen in such a way that the mesh remains valid during the whole time interval. For the untangling and smoothing technique (U-S), the tests were solved in the following way: for a given relative deformation of the domain, a linear law of movement of the boundary was applied using different number of time steps. The domain deformations were chosen varying from 10 % to the maximum deformation, with steps of 10 %.

5.1. Test 1

Figure 5 shows the original domain and the deformation sequence for this problem. The test was solved for a relative domain deformation of 50 %, 90 % and 99 %. A structured mesh with 200 triangular elements and 121 nodes was employed. The initial tangled mesh and the resulting valid meshes are presented (figures 6 to 9). The evolution of the minimum quality with iterations is also included (figure 10).

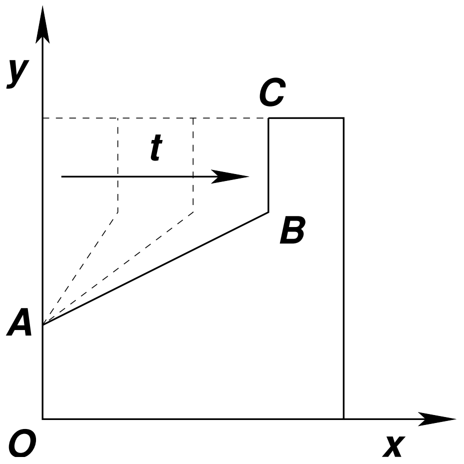

Figure 5. Test 1 - Domain.

Figure 11 shows the comparison between S and U-S strategies. It is observed that the elapsed time in the computation with the U-S technique is approximately independent of the

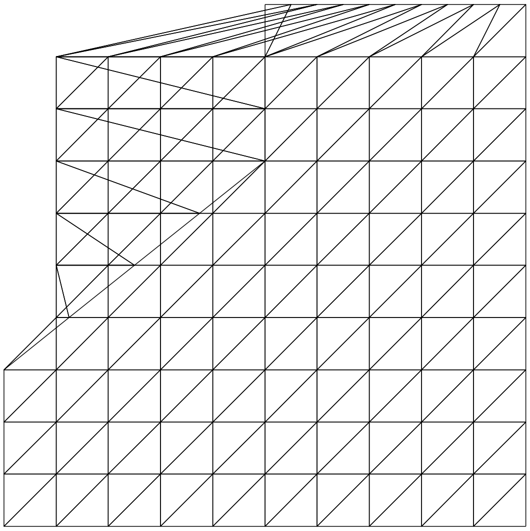

Figure 6. Test 1 - 50 % def. - Initial mesh.

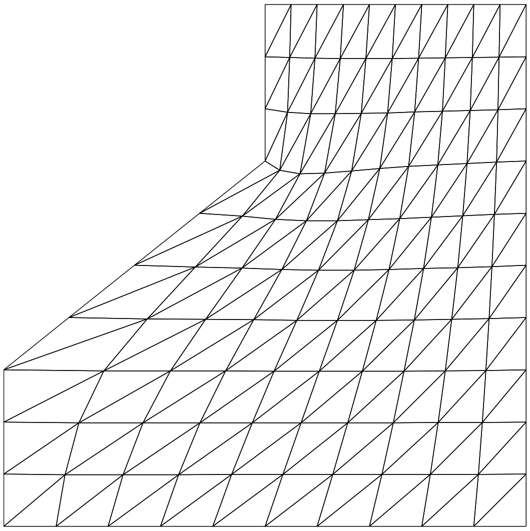

Figure 7. Test 1 - 50 % def. - Final mesh.

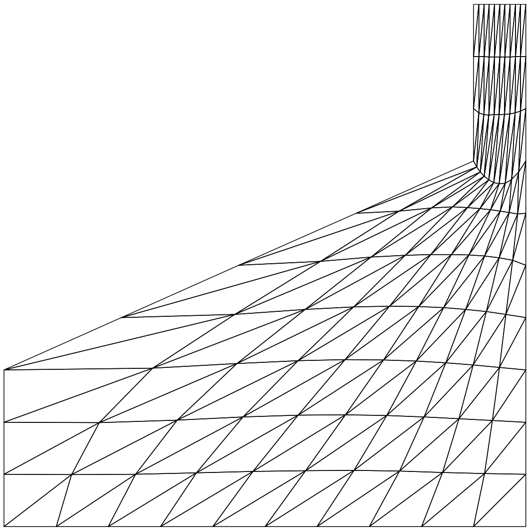

Figure 8. Test 1 - 90 % def. - Final mesh.

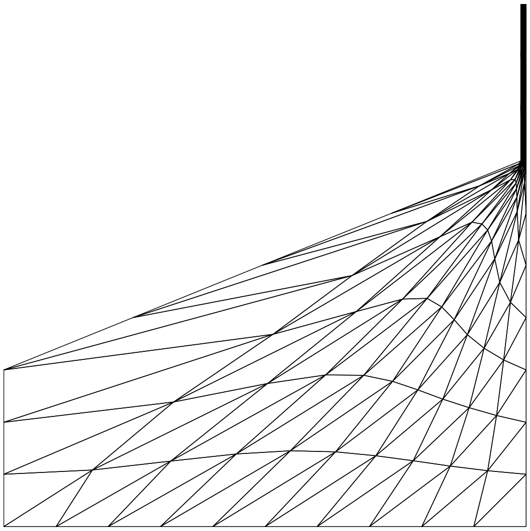

Figure 9. Test 1 - 99 % def. - Final mesh.

deformation. It does depend on the number of time steps used, and tends to the elapsed time of the S method as the number of time steps are increased.

In order to verify the utility of the differential predictor (DP) explained in section 4.1, the test was solved for a relative domain deformation of 90 %, varying the amount of time steps used. In table I the total number of iterations in each case (with and without DP) is presented. As it is observed, the use of the DP makes it possible to diminish in a noticeable manner the

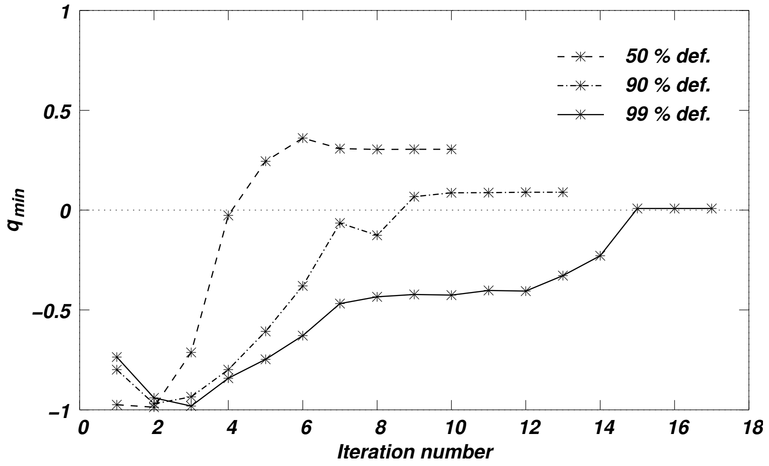

Figure 10. Test 1 - Evolution of $q_{\min}$.

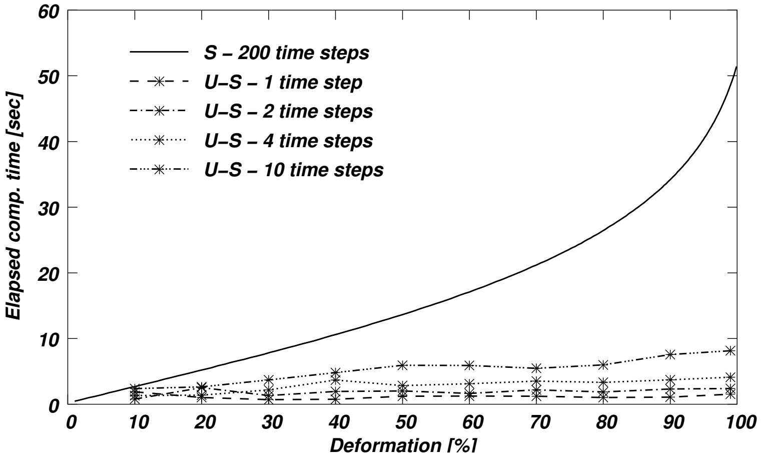

Figure 11. Test 1 - Computational time in terms of the relative domain deformation.

number of iterations (and therefore the cost), mainly when the time step used diminishes.

5.2. Test 2

This test contains two squares, one inside the other initially centered, as observed in figure 12. The internal square is displaced without contact towards one of the sides of the external square. The mesh has 710 triangular elements and 415 nodes. Three cases with different domain deformations were solved: 50 %, 90 % and 99 %. The results achieved are presented in figures 13 to 18.

Table I. Total number of iterations to reach a relative domain deformation of 90 % for test 1

time steps	with DP	without DP
1	15	13
2	19	27
4	34	47
8	39	69

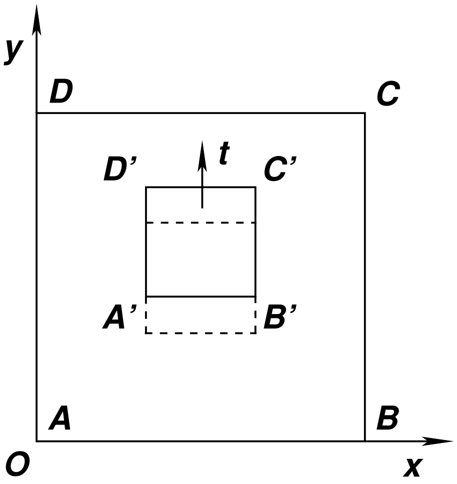

Figure 12. Test 2 - Domain.

Table II. Total number of iterations to reach a relative domain deformation of 90 % for test 2

time steps	with DP	without DP
1	9	13
2	13	21
4	17	38
8	24	51

Table II presents the total number of iterations necessary to solve the test with 90 % of relative domain deformation when the amount of time steps is increased. This table shows a comparison of what happens when the DP is applied and when it is not. As it is observed, the advantages of using the DP are evident when the amount of time steps increases.

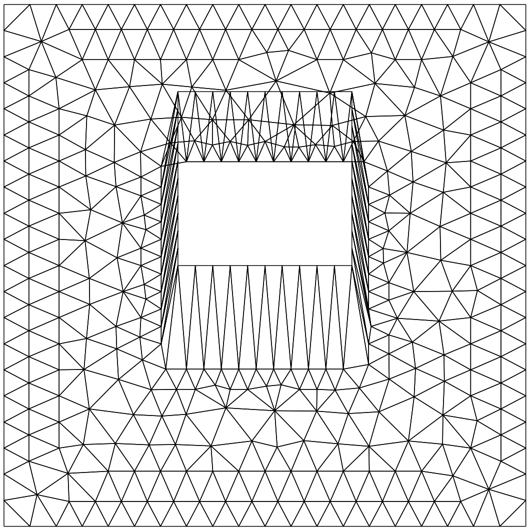

Figure 13. Test 2 - 50 % def. - Initial mesh.

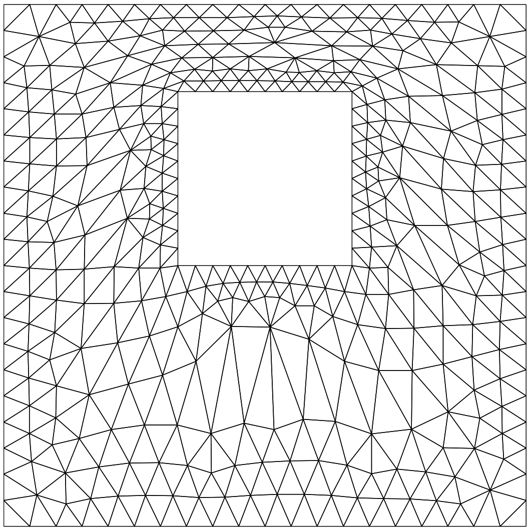

Figure 14. Test 2 - 50 % def. - Final mesh.

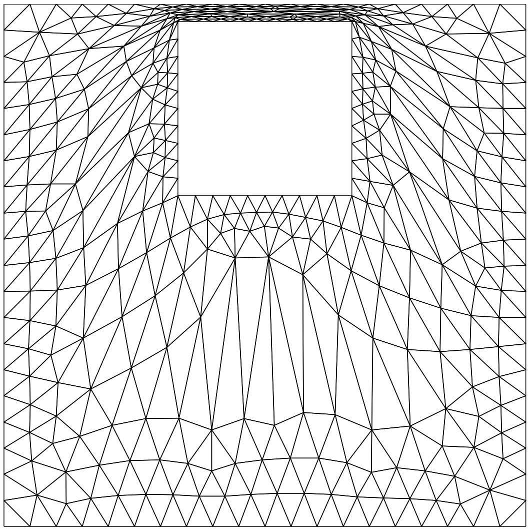

Figure 15. Test 2 - 90 % def. - Final mesh.

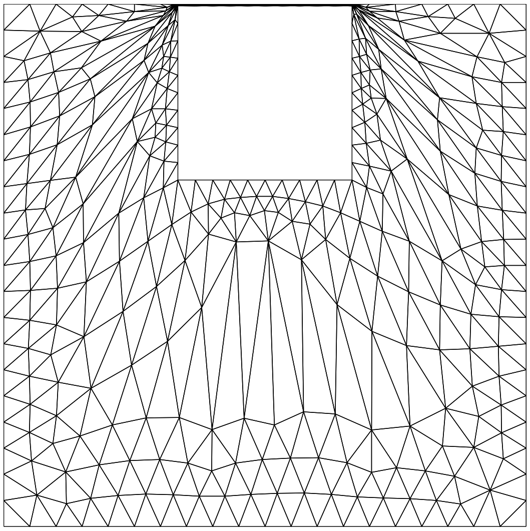

Figure 16. Test 2 - 99 % def. - Final mesh.

5.3. Test 3

This test is the 3D extension of the test presented in subsection 5.1. Figure 19 shows the domain for different deformations. The top face of the cube is moved in vertical direction. During the deformation, this face is transformed into two planes at different heights joined by a truncated cone with upper radius r and lower radius R.

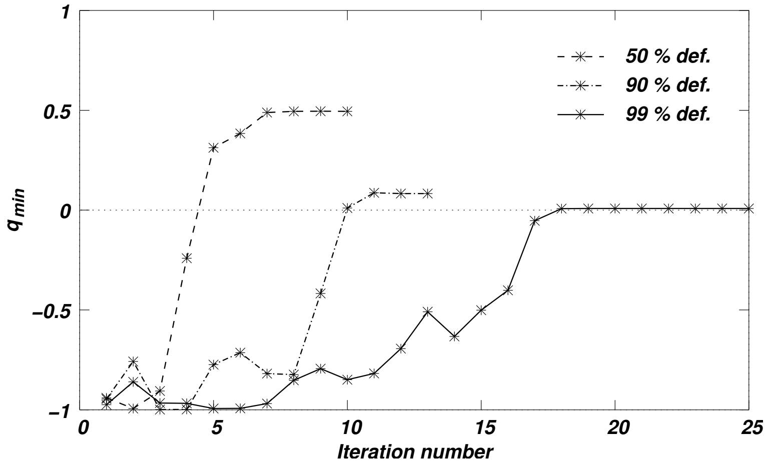

Figure 17. Test 2 - Evolution of $q_{\min}$.

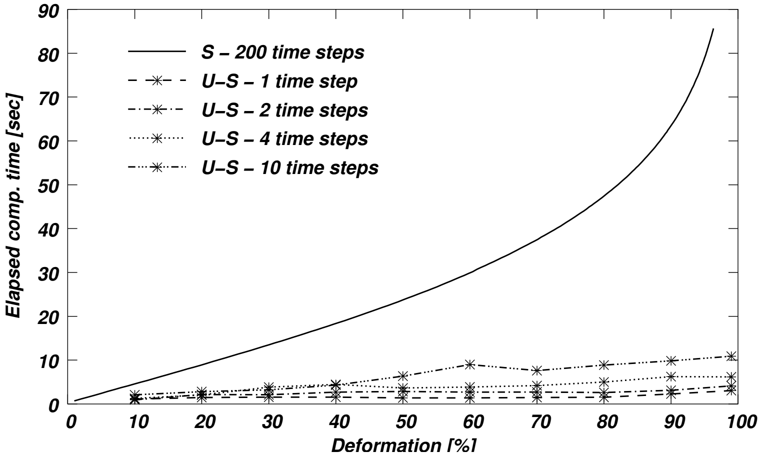

Figure 18. Test 2 - Computational time in terms of the relative domain deformation.

A mesh with 1080 tetrahedral elements and 343 nodes was used. This test was solved for 50 % and 87 % of relative domain deformation. Figures 20 to 25 show the results achieved.

Again, as in previous tests, the advantage of applying the DP can be observed in table III. In this case, the problem was solved for a relative domain deformation of 70 % varying the size of the time step.

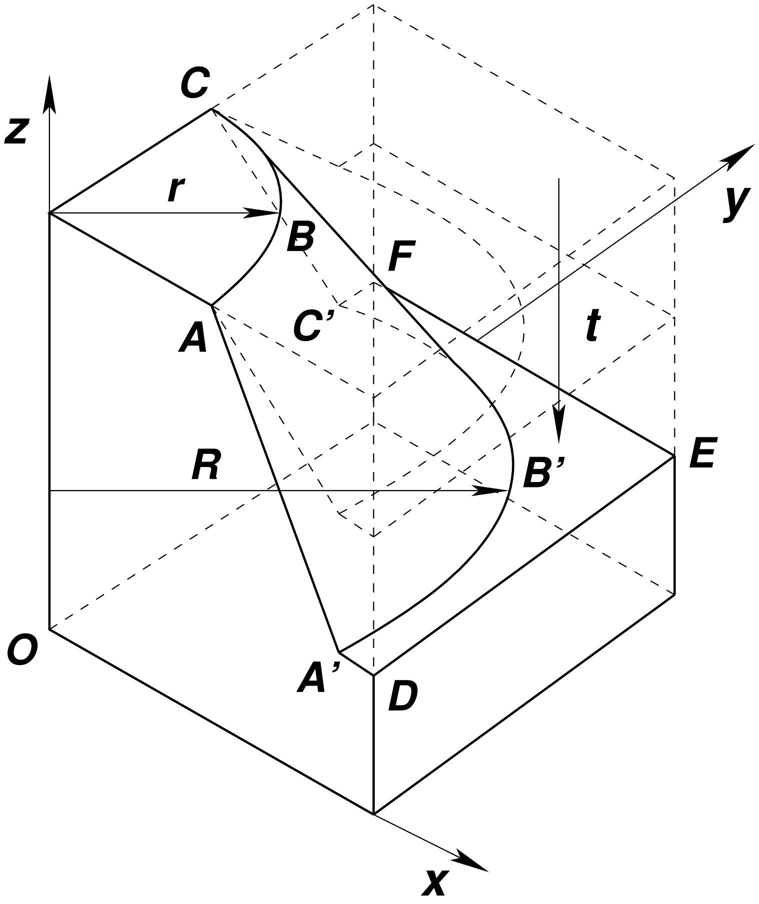

Figure 19. Test 3 - Problem definition.

Table III. Total number of iterations to reach a relative domain deformation of 70 % for test 3

time steps	with DP	without DP
1	8	7
2	9	12
4	13	24
8	20	48

5.4. Test 4

This test was taken from [33] and it consists in a unit cube with 625 tetrahedras and 216 nodes. The invalid initial mesh was obtained by transforming the cube into another cube of 10

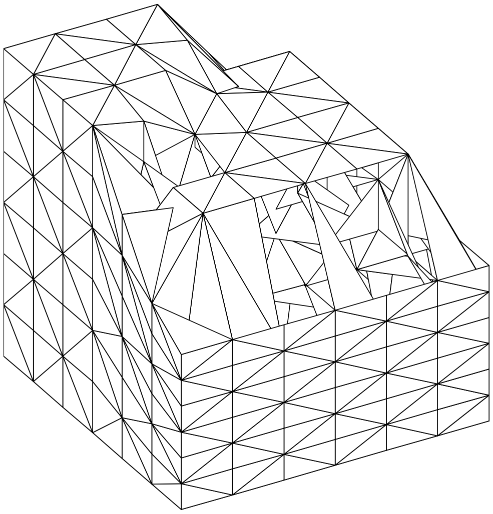

Figure 20. Test 3 - 50 % def. - Initial mesh.

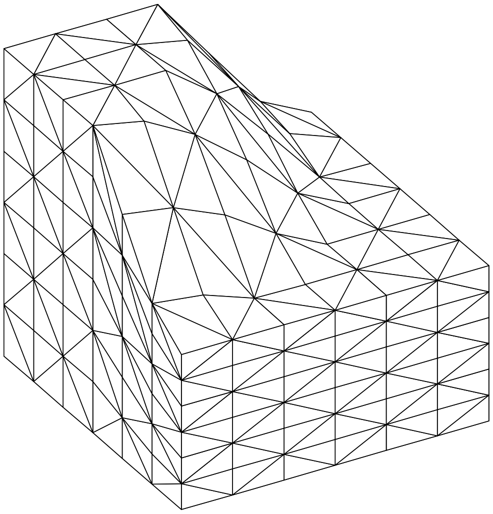

Figure 21. Test 3 - 50 % def. - Final mesh.

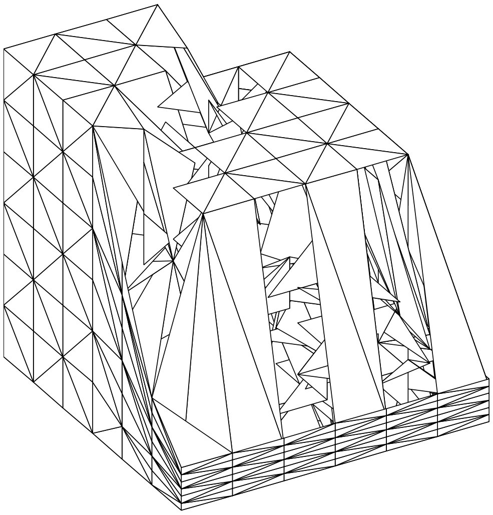

Figure 22. Test 3 - 87 % def. - Initial mesh.

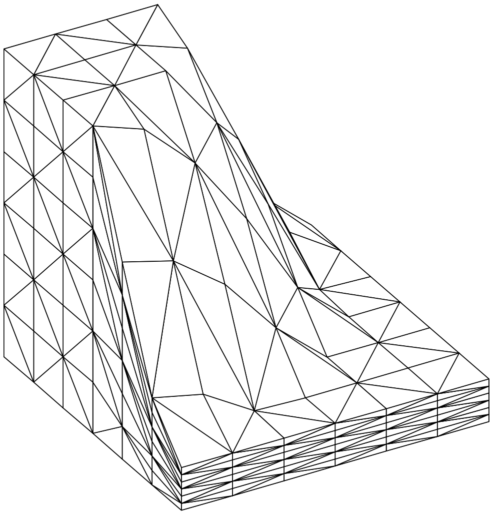

Figure 23. Test 3 - 87 % def. - Final mesh.

units of side length, changing the coordinates of some nodes in the following way: the internal nodes were kept fixed, those nodes lying on the edges were relocated on the edges of the new cube and those nodes lying on the faces of the original cube were projected on the new faces, respectively. (see figure 26). Figure 27 shows the final mesh obtained.

In figure 28 the minimum quality as a function of the number of iterations is presented.

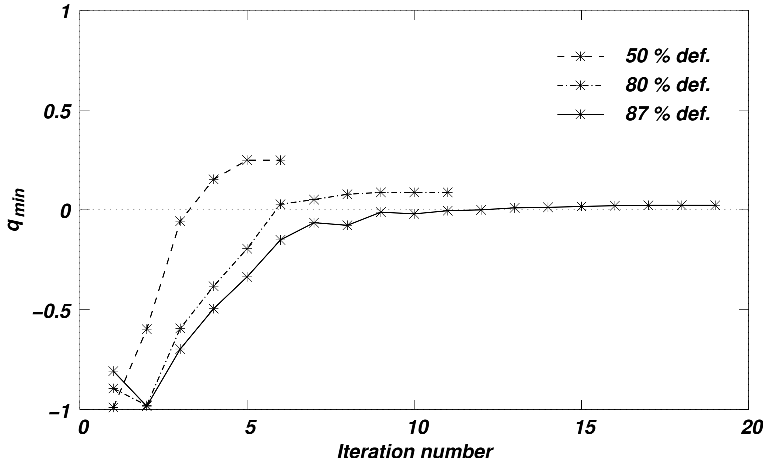

Figure 24. Test 3 - Evolution of $q_{\min}$.

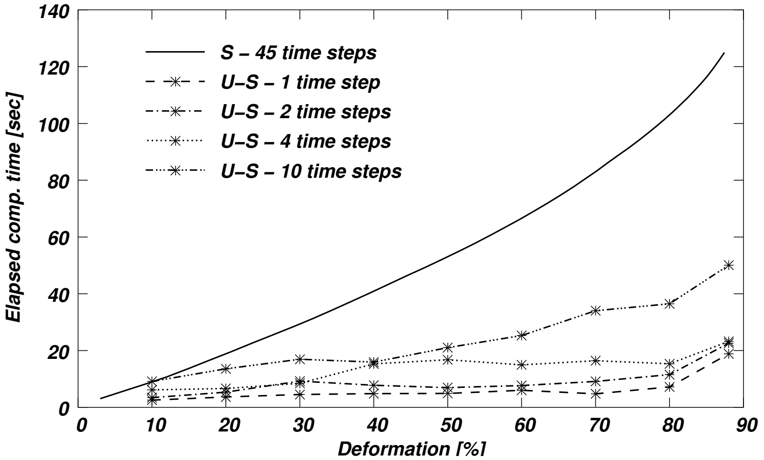

Figure 25. Test 3 - Computational time in terms of the relative domain deformation.

5.5. Mesh generation

As it was previously mentioned, the strategy of simultaneous mesh untangling and smoothing may serve as an algorithm for mesh generation. Here, a 2D test example is presented to give an idea of its potentiality. This case was extracted from [35] and it consists in finding the mesh for the domain bounded by $x = 0$, $x = 1$, $y = 1$ and $y = 0.75 + 0.25 \sin(\pi(0.5 + 2x))$. To this purpose, and initial mesh composed of 800 triangular elements and 441 nodes covering a unit square was chosen. Deforming the boundaries of the original square to match the desired final boundary, an invalid mesh is temporarily obtained. To fix this mesh, the proposed technique

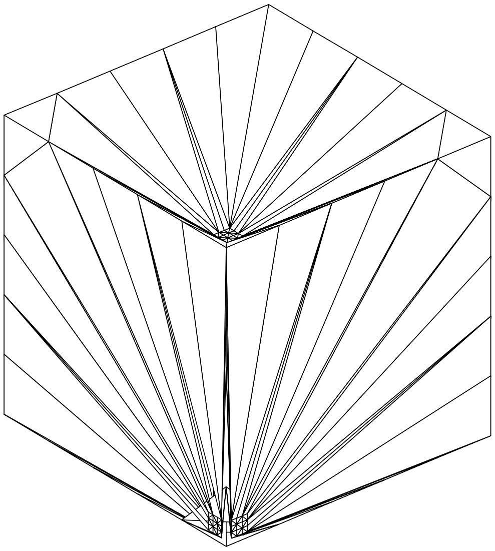

Figure 26. Test 4 - Initial mesh.

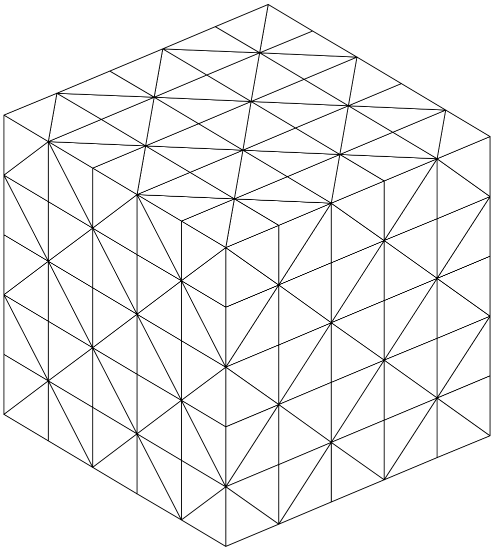

Figure 27. Test 4 - Final mesh.

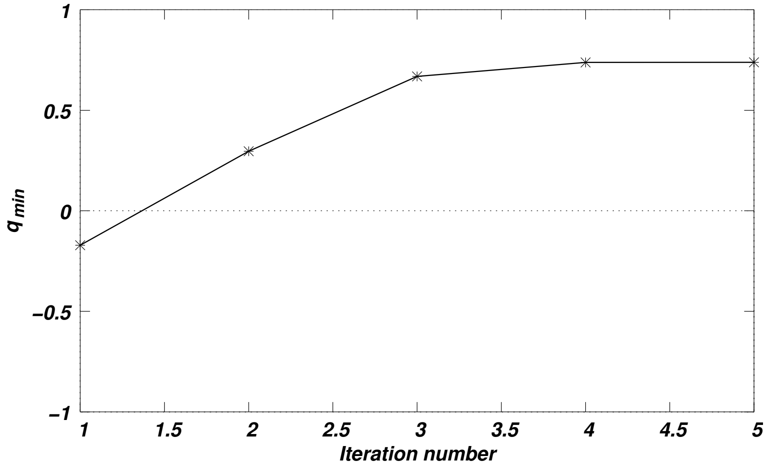

Figure 28. Test 4 - Evolution of q_{min}.

is applied as in mesh dynamics problems. The achieved results are shown in figures 29 and 30.

6. Conclusions

In this work, a simultaneous mesh untangling and smoothing technique based on the solution of an optimization problem solved in a global way was presented. From the results obtained

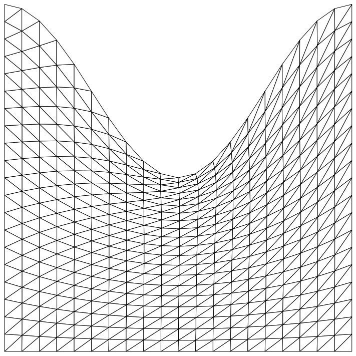

Figure 29. Final mesh.

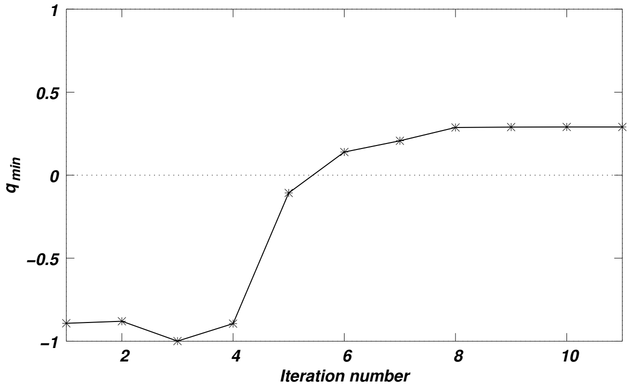

Figure 30. Evolution of q_{min}.

for several tests cases in both 2D and 3D with medium, large and extra large deformation, it may be possible to reach the conclusion that this procedure is very robust. Normally, in FSI problems the time step size is restricted by one of the two physical problems, being the mesh dynamics an auxiliary problem, it is expected that it will not be more restrictive than any of the other two. However, in several applications the refinement imposes the reduction of the time step size to the mesh dynamics in order to avoid the element inversion. The enhancement of the CMD with simultaneous untangling and smoothing circumvents this drawback. A global solver is very attractive to make this procedure more user-independent. The computational cost of each time step is scarcely higher than the one of the original mesh smoothing strategy. However, taking into account that this new procedure does not alter the time step size in FSI problems, it generally makes the global computational cost cheaper and the procedure more robust. As a side effect, this proposed technique was successfully applied to mesh generation of a simple 2D domain. In future works, it is expected that this capability will be proved in others 3D meshes in more complex domains, and that the functional will be extended to get nearly conformal meshes to be applied in airfoil meshing.

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Silvia Montoro for her useful discussion about English grammar.

This work has received financial support from Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina, grant PIP-02552/2000), Universidad Nacional del Litoral (Argentina, grants CAI+D 2000/43) and ANPCyT(Argentina, grants PICT 6973/99 (Proa), PID-74/99 (Flags), PID-76/99 (Melt), PICT 12-14573/2003 (Lambda), PME 209/2003 (Cluster)). Extensive use of freely distributed software as GNU/Linux OS, MPI, PETSc, gcc compilers, Octave, Open-DX among many others is made.

REFERENCES

1. Tezduyar TE. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids* 2003; **43**: 555–575.

2. Tezduyar TE. *Finite Element Methods for Fluid Dynamics with Moving Boundaries and Interfaces*, volume 3: Fluids. John Wiley & Sons (eds. E. Stein, R. De Borst and T.J.R. Hughes), New York, 2004.
3. Hughes TJR, Liu WK, Zimmermann TK. Lagrangian-Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering* 1981; **29**: 239–349.
4. Belytschko T, Flanagan DP, Kennedy JM. Finite element methods with user-controlled meshes for fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering* 1982; **33**: 669–688.
5. Donea J, Giuliani S, Halleux JP. An arbitrary, Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering* 1982; **33**: 689–700.
6. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering* 1994; **119**: 73–94.
7. Löhner R, Yang C. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering* 1996; **12**(10): 599–608.
8. Farhat C, Degand C, Koobus B, Lesoinne M. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering* 1998; **169**: 231–245.
9. Kjellgren P, Hyvärinen J. An arbitrary Lagrangean-Eulerian finite element method. *Computational Mechanics* 1998; **21**: 81–90.
10. Koobus B, Farhat C. Second order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering* 1999; **170**(1): 103–129.
11. Blom FJ. Considerations on the spring analogy. *Numerical Methods in Fluids* 2000; **32**(6): 647–668.
12. Chiandussi G, Bugeda G, Oñate E. A simple method for automatic update of finite element meshes. *Communications in Numerical Methods in Engineering* 2000; **16**(1): 1–19.
13. Farhat C, Geuzaine P, Grandmont C. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics* 2001; **174**(2): 669–694.
14. Bar-Yoseph PZ, Mereu S, Chippada S, Kalro VJ. Automatic monitoring of element shape quality in 2-D and 3-D computational mesh dynamics. *Computational Mechanics* 2001; **27**(5): 378–395.
15. Stein K, Tezduyar TE, Benney R. Automatic mesh update with the solid-extension mesh moving technique. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**: 2019–2032.
16. Farhat C, Geuzaine P. Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**: 4073–4095.
17. Burg CO. A robust unstructured grid movement strategy using three-dimensional torsional springs. 34^{th} *AIAA Fluid Dynamics Conference and Exhibit* 2004; AIAA Paper 2004-2529.
18. Storti M, Nigro N, Paz R. Strong coupling strategy for fluid-structure interaction problems in supersonic regime via fixed point iteration. *Journal of Sound and Vibration* 2006; submitted.
19. Tezduyar TE, Sathe S, Keedy R, Stein K. Space-time finite element techniques for computation of fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**: 2002–2027.
20. Winslow AM. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics* 1997; **135**(2): 128–138.
21. Knupp PM. Winslow smoothing on two-dimensional unstructured meshes. *Engineering with Computers* 1999; **15**(3): 263–268.
22. Parthasarathy VN, Kodiyalam S. A constrained optimization approach to finite element mesh smoothing. *Finite Elements in Analysis and Design* 1991; **9**: 309–320.
23. Amezua E, Hormaza MV, Hernandez A, Ajuria MBG. A method for the improvement of 3D solid finite-element meshes. *Advances in Engineering Software* 1995; **22**(1): 45–53.
24. Zavattieri PD, Dari EA, Buscaglia GC. Optimization strategies in unstructured mesh generation. *International Journal for Numerical Methods in Engineering* 1996; **39**: 2055–2071.
25. Cannan S, Tristano J, Staten M. An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral and quad-dominant meshes. *Proceedings of the 7^{th} International Meshing Roundtable*, Dearborn, MI, 1998; pages 479–494.
26. Amenta N, Bern M, Eppstein D. Optimal point placement for mesh smoothing. *Journal of Algorithms* 1999; **30**(2): 302–322.
27. Freitag LA, Knupp PM. Tetrahedral element shape optimization via the jacobian determinant and condition number. *Proceedings of the 8^{th} International Meshing Roundtable*, Sandia National Laboratories, 1999; pages 247–258.
28. López EJ, Nigro NM, Storti MA, Toth JA. A minimal element distortion strategy for computational mesh dynamics. *International Journal for Numerical Methods in Engineering* 2007; **69**: 1898–1929.
29. Knupp P. Hexahedral mesh untangling and algebraic mesh quality metrics. *Proceedings of 9^{th} International*

Meshing Roundtable 2000; pages 173–183.

30. Freitag LA, Plassmann P. Local optimization-based simplicial mesh untangling and improvement. *International Journal for Numerical Methods in Engineering* 2000; **49**(1): 109–125.
31. Delanaye M, Hirsch Ch, Kovalev K. Untangling and optimization of unstructured hexahedral meshes. *Computational Mathematics and Mathematical Physics* 2003; **43**(6): 807–814.
32. Escobar JM, Rodríguez E, Montenegro R, Montero G, González-Yuste JM. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**: 2775–2787.
33. Montenegro R, Escobar JM, Rodríguez E, Montero G, González-Yuste JM. Improved objective functions for tetrahedral mesh optimisation. *International Conference on Computational Science* 2003; **2657**: 568–580.
34. Papalambros PY, Wilde DJ. *Principles of Optimal Design. Modeling and Computation.* Cambridge University Press, 1988.
35. Haussling HJ, Coleman RM. A method for generation of orthogonal and nearly orthogonal boundary-fitted coordinate systems. *Journal of Computational Physics* 1981; **43**: 373–381.