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 and fixed grids

Ezequiel J. López[1,*], Norberto M. Nigro[1] and Mario A. Storti[1]

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

SUMMARY

In this paper a useful technique for simultaneous untangling and smoothing of meshes is presented. It is based on an extension of an earlier mesh smoothing strategy developed by the authors and used to solve the computational mesh dynamics stage in fluid-structure interaction problems. In moving grid problems, mesh untangling is necessary when element inversion is happened due to a moving domain boundary. The original smoothing strategy is defined in terms of the minimization of a functional associated with the mesh distortion using an indicator of the element geometric quality. This functional becomes discontinuous when an element has null volume, making impossible to obtain a valid mesh from an invalid one. To circumvent this drawback, the original functional is transformed in order to guarantee its continuity for the whole space of nodal coordinates, achieving the untangling technique. The regularization depends on one parameter, allowing the recovery of the original functional when this parameter tends to zero. This feature is very important because at first it is necessary to regularize the functional in order to make the mesh valid, but then, it is advisable to use the original functional to make the smoothing optimal. This technique is applied to several test cases, including 2D and 3D meshes with simplicial elements. An additional example shows how this technique may be used for mesh generation too and also how to extend this technique for non-simplicial grids.

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

1. Introduction

Several scientific and industrial applications involve moving meshes. This class of problem include those with free surfaces, two-fluid interfaces, fluid-object and fluid-structure interactions, and moving mechanical components.

In computation of flow problems with moving boundaries and interfaces, depending on the complexity of the interface and other aspects of the problem, we can use an interface-tracking or interface-capturing technique. An interface-tracking technique requires meshes that "track" the interfaces. The mesh needs to be updated as the flow evolves. In an interface-capturing

*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

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].

In fluid-structure interaction (FSI) problems, one of the most popular interface tracking technique is the Arbitrary Lagrangian Eulerian (ALE) formulation [2, 3, 4]

One example for such a physical problem in solid mechanics with large deformations is metal forming. In these cases the mesh is updated at each time step due to the motion of the domain boundary, producing severe mesh quality deterioration and, in some situations, producing 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 mesh motion are grouped in a special topic named CMD (Computational Mesh Dynamics). Its importance may be assessed simply making an inspection to the current bibliography [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16].

Here, the main interest is the resolution of FSI problems. This kind of applications requires the solution of the fluid and the structure problems, coupled in the physical domain. The flow problem dictates the generalized forces acting over the structure. The structural problem determines the geometry variation due to its deformation and also its translational and rotational motion. Nowadays, efficient solution of FSI problems with large displacement relative of the boundary is still a challenging problem in Computational Mechanics. Generally 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 shear stresses are stored at each point of the fluid-structure interface. Then these time-dependent loads are transferred to a CSD (Computational Structure Dynamics) code, which finds the deformation of the structure in time [17, 1]. Then 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 also the accuracy of the overall approach.

Generally the CFD stage consumes most of the CPU time adding restrictions in time step, specially in regions where the mesh is refined. The boundary motion may also cause the inversion of some elements, forcing to reduce the time step due to CMD reasons. This situation is not desirable because the CMD problem is artificially introduced to follow the moving domain. To avoid this limitation it is advisable to introduce the possibility to fix the inverted elements instead of reducing the time step. This kind of solution is possible via a mesh untangling strategy.

This paper contributes in this sense, first the untangling makes the CMD implementation more robust without influence on the time step selection. Furthermore, the smoothing capability is necessary to optimize the mesh quality with consequences on 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 reallocated one by one iteratively keeping fixed the remainder until the convergence is reached.

The most popular strategy of mesh smoothing is the laplacian smoothing due to its low computational cost and its simple implementation. This method moves the internal nodes to the geometrical center of their neighbors. However, this method does not always work without

warranty of producing valid meshes. Also, in cases where convergence is not reached, the final mesh may depend of the nodal sequence ordering. This last feature is frequently found in local smoothing techniques. Other kind of smoothing related with the laplacian technique is the Winslow smoothing, which is more robust in terms of avoiding the inversion of the elements in the mesh. It is based on logical variables with the requirement of these variables be harmonic functions [18]. Originally this technique was presented for structured meshes, being lately extended to unstructured meshes by Knupp [19]. There are several smoothing methods based on optimization having the common goal of improving the mesh quality in terms of some quality indicator [20, 21, 22, 23, 24, 25, 26]. The main disadvantage of such methods is their computational cost. Depending on the way in which the system is solved, there are two main kind of methods: local methods and global methods. The global methods update the nodal position simultaneously for the whole set of nodes.

The smoothing technique may confront with situations where the mesh is invalid, then an untangling methodology need to be used. These methods are normally based on the element volume [27] and in general both procedures, smoothing and untangling, are treated separately. This tedious task may be incorporated for mesh generation where the user is interactively looking for a good quality mesh. Thinking in FSI problems, the CMD should have the capability of solving the mesh motion even though inverted elements were founded and guaranteeing a smooth mesh at each time. Therefore a simultaneous procedure of smoothing and untangling is preferable [28, 29, 30, 31]. In this paper a simultaneous untangling and smoothing technique is proposed based on the optimization of the grid quality. The strategy arises from the regularization of a functional presented previously [26] and applied to solve the mesh dynamics in FSI problems.

The proposed method can be useful for mesh generation too. In this case the topology is generated in an auxiliary domain in which the mesh may be generated in a structured way. 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 here presented, a valid mesh is generated.

The paper is organized in the following way: first the original mesh smoothing method is briefly presented with the modifications introduced to avoid relaxation of initial mesh, next the functional regularization and the strategy used for solving the optimization problem is presented. Then, several results for CMD in 2D and 3D are included and a 2D mesh generation application of this strategy is shown. Finally 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 [26]. The technique is based on an optimization problem solved in a global way, where the functional means the mesh distortion. Such a functional was defined in the following way

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

with

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

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

Written in this way, the functional allows to be applied for any kind of elements defining the quality indicator properly. In the present 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:

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

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

Due to the quality indicator for non-simplices elements is based on those defined for simplices ones, without lost of generality only the simplicial element case is analyzed.

The proposed functional is continuous if $q_e \neq 0$ for all the elements in the mesh. This property is lost when q_e tends to zero and consequently $F_e(\mathbf{x})_e$ tends to infinity. 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, being impossible to recover a valid mesh starting from an invalid one. In FSI problems this limitation is by-passed decreasing the time step size avoiding the tangling of the grid caused by the motion of the boundary. However, the computational cost largely increases, specially if some clustering of nodes are used close to the moving boundary to capture fluid dynamics details like boundary layers. It is in this sense that the computational mesh dynamics strategy enhanced with simultaneous untangling and smoothing is largely useful.

2.1. Avoiding relaxation of initial mesh

Many times the initial mesh introduced by the user is not the optimal mesh with respect to the functional, as proposed above. 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 in the horizontal direction. As a consequence of the optimization problem the elements near vertices A and C tend to shrink, whereas those near vertices B and D tend to grow. This effect is caused by the particular way the squares have been split in 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 be or may be not 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 that 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 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.

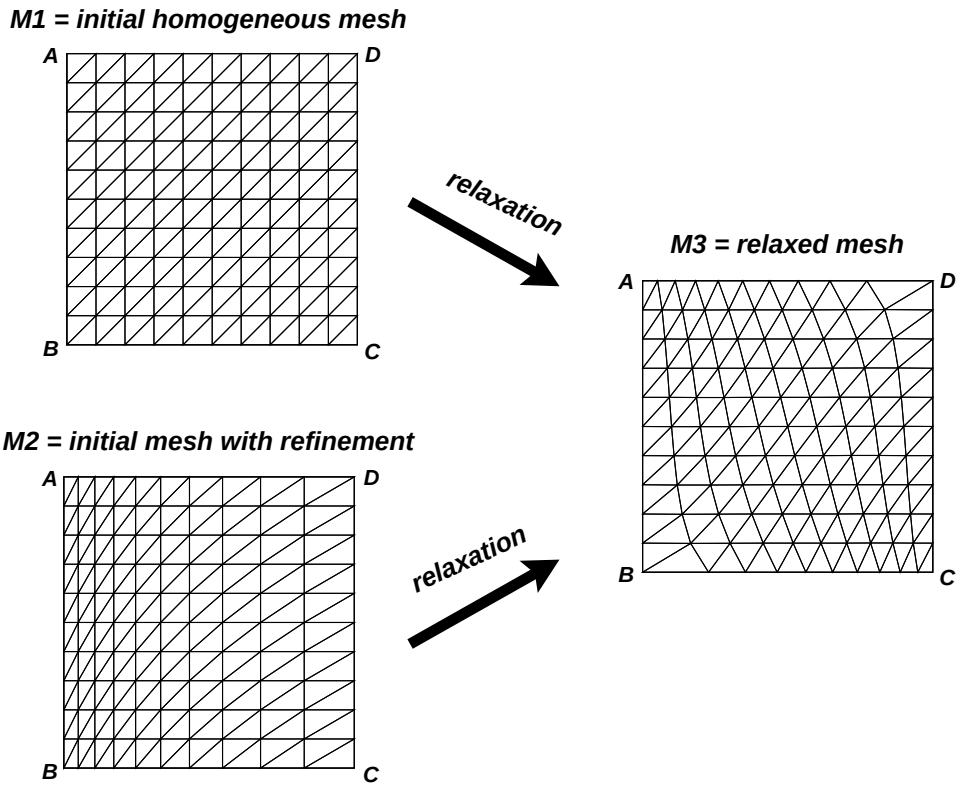

Figure 1. Relaxation of meshes

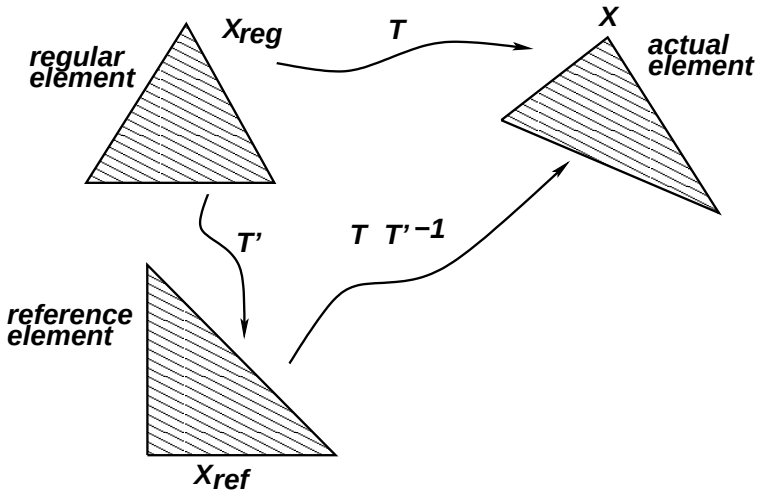

Figure 2. Compensating for initial deformation in reference mesh

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 transform 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 can be seen because, the functional can be computed by taking the nodal coordinates of the regular elements, applying the transformation and then compute the side lengths, volume, and finally the functional. All this computation is encapsulated in the function $g(\,\cdot\,)$. Of course, the functional doesn't depend on the translation $\mathbf{x}_0$. Moreover, it does depend only on the metric of the transformation $\mathbf{T}^T\mathbf{T}$, because it is independent of rotations, but for the analysis that will follow 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 a scaling factor and $\mathbf{O}$ an orthogonal matrix, since in this case the actual element is similar to the regular element.

The objective is to modify the functional so that the optimal element shape for F_e is not the regular element shape, besides the shape of some reference element with coordinates $\{\mathbf{x}_{\mathrm{ref},j}\}$ (see figure 2). It is easy to see that this can be done by considering the transformation from the reference element to the actual 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 actual 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 $\mathbf{x}'_j = \mathbf{T}\,\mathbf{T'}^{-1}\mathbf{x}_{\mathrm{reg},j}$.

An example can be seen in figure 3. The original mesh at 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 mesh shown at left). Note that as no initial relaxation is produced, the final mesh has still the refinement towards the AB side.

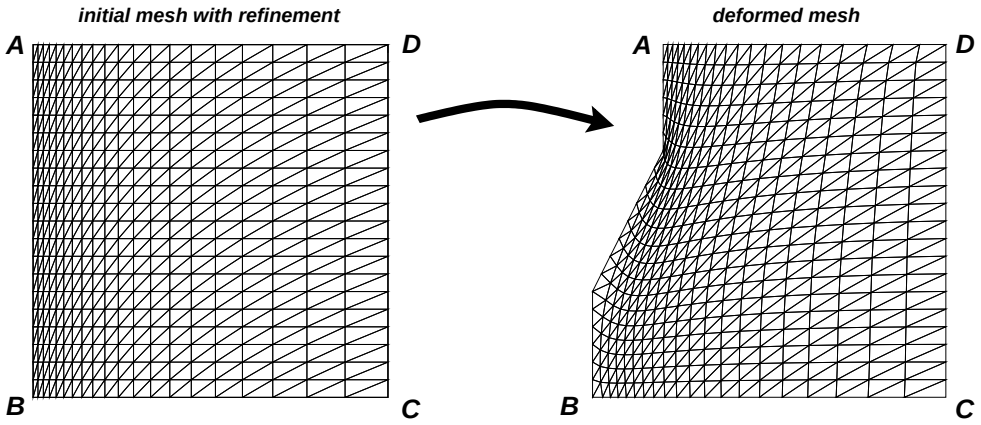

Figure 3. Deformation of mesh with surface refinement

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 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 paper to include the untangling capability to our original smoothing technique, the functional was modified using an idea proposed in [30], that makes the functional continuous for all element volume values. This modification consists in replacing V in equation (4) by the following 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 also it is a positive function for all V (see figure 4). The parameter δ represents the value of the function for a null volume.

The modified functional is then written as:

$$F^*_e(\mathbf{x})_e = q^{*n}_e \tag{11}$$

with

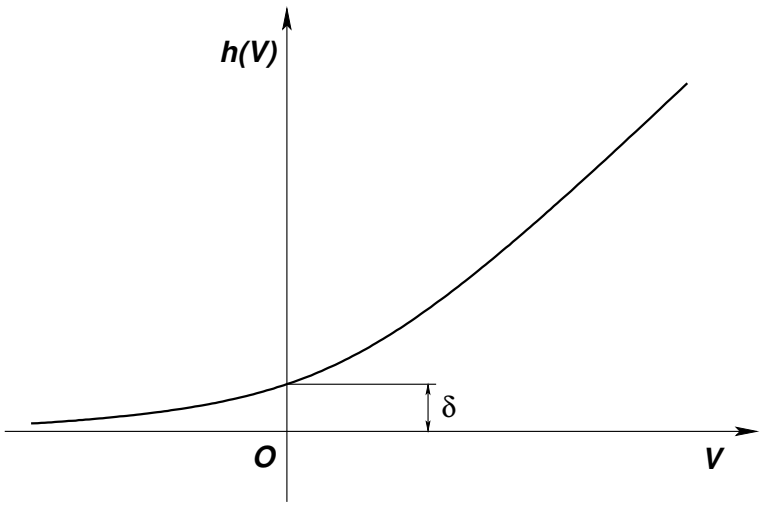

Figure 4. Function $h(V)$.

$$q_e^* = C\frac{h(V)}{\sum_i l_i^{nd}} \tag{12}$$

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 $\delta \to 0$, $F^*(\mathbf{x}) \to F(\mathbf{x})$ point-wise.

The following simple example extracted from [30] shows how the original and the modified functional work. Let the 2D mesh formed by the triangles ABp, BCp and CAp as it is observed in figure 5. The points $A(0,-1)$, $B(\sqrt{3},0)$ and $C(0,1)$ are kept fixed, while the point $p(x,y)$ has their coordinates free to change. A valid mesh is obtained if the point p lies on the interior of the triangle ABC, named the *feasible region* for the point p. In figure 6 the objective functions for several values of the parameter δ as a function of x for $y = 0$ are shown. Note that for $\delta \neq 0$ the functional is continuous in $\mathbb{R}$. It is also noticeable that decreasing the parameter δ, the modified optimal solution approximates the original one (see the curve for $\delta = 0$). In this very simplified example the modified functionals have their optimal solution within the feasible region, feature that not always may be guaranteed when the parameter δ is relatively high.

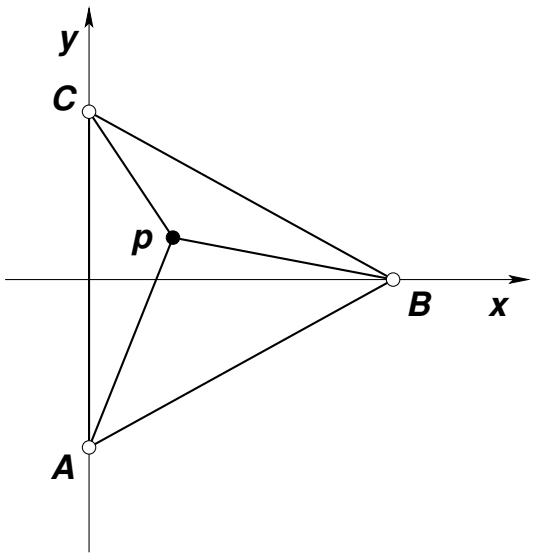

Figure 5. Mesh example 2D.

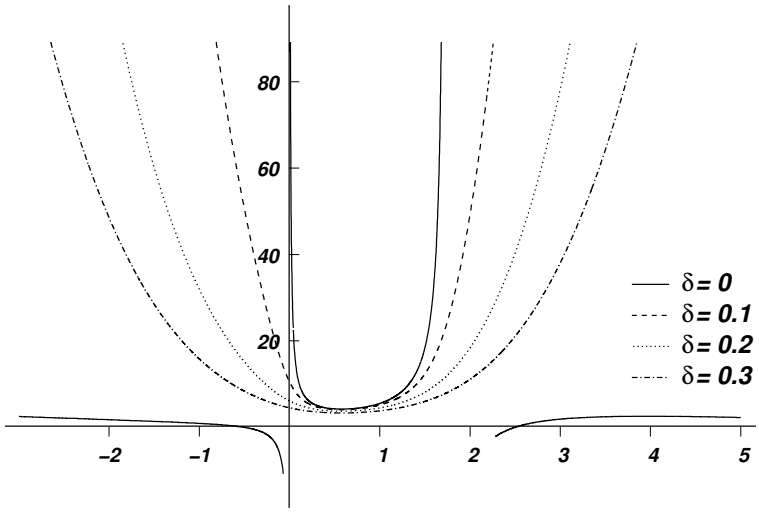

Figure 6. Regularized functional for some values of the parameter δ.

4. Solving strategy

In the present 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 optimal the original functional 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. In addition, if δ is relatively high the optimal mesh may be invalid, being even worse if the relative deformation of the domain is greater. Therefore, two main problems arise:

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

To determine an equation that allows to 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

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

Rewriting the above equation in the following way

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

it is observed that this system may be solved in an uncoupled way if the parameter δ is kept fixed for the first equation. Thus, the variable increments $\Delta \mathbf{x}$ and $\Delta \delta$ are written as

$$\begin{aligned} \Delta \mathbf{x} &= -\left(\frac{\partial^2 F^*}{\partial \mathbf{x}^2} \right)^{-1} \frac{\partial F^*}{\partial \mathbf{x}} \\ \Delta \delta &= -\frac{\left(\frac{\partial F^*}{\partial \delta} + \frac{\partial^2 F^*}{\partial \delta \partial \mathbf{x}} \Delta \mathbf{x} \right)}{\frac{\partial^2 F^*}{\partial \delta^2}} \end{aligned} \tag{14}$$

The expression for $\Delta \delta$ in (14) is adopted like the maximum value to reduce δ, limiting in this way the decreasing rate of the sequence to avoid loosing of convergence.

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

$$\delta^k = \max(\delta^{k-1} - \alpha |\Delta \delta|, \beta \delta^{k-1}) \tag{15}$$

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 do 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 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 [32]. At each iteration δ is diminished only if the line search strategy gives a unit step as result. This technique has a computational cost scarcely greater than the original one. Details of the CPU times for the original strategy were published in [26].

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 $\mathbf{x}_{\text{int}}$

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_b \\ \mathbf{x}_{\text{int}} \end{bmatrix} \tag{16}$$

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

$$\mathbf{x}_{\text{int}}^n = \underset{\tilde{\mathbf{x}}_{\text{int}}}{\text{argmin}}\, F\left(\begin{bmatrix} \mathbf{x}_b^n \\ \tilde{\mathbf{x}}_{\text{int}} \end{bmatrix}\right) \tag{17}$$

The recurrence formula from the Newton-Raphson strategy is

$$\mathbf{x}_{\text{int}}^{n,j+1} = \mathbf{x}_{\text{int}}^{n,j} - (\mathbf{K}^j)^{-1}\mathbf{R}^j \tag{18}$$

where

$$\begin{aligned} \mathbf{R} &= \frac{\partial F}{\partial \mathbf{x}_{\text{int}}} \\ \mathbf{K} &= \frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}} \end{aligned} \tag{19}$$

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

$$\lim_{j\to\infty} \mathbf{x}_{\text{int}}^{n,j} = \mathbf{x}_{\text{int}}^n \tag{20}$$

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

$$\mathbf{x}_{\text{int}}^{n,0} = \mathbf{x}_{\text{int}}^{n-1,\infty} \tag{21}$$

However, this has the drawback that, if the elements near the moving mesh 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 moving mesh problem 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

$$\mathbf{R}(\mathbf{x}_{\text{int}}(t), \mathbf{x}_b(t)) = 0 \tag{22}$$

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

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_{\text{int}}}\right)_{t^{n-1}} \dot{\mathbf{x}}_{\text{int}}(t^{n-1}) + \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_b}\right)_{t^{n-1}} \dot{\mathbf{x}}_b(t^{n-1}) = 0 \tag{23}$$

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

$$\mathbf{x}_{\text{int}}^{n,0} = \mathbf{x}_{\text{int}}^{n-1,\infty} + \Delta t\, \dot{\mathbf{x}}_{\text{int}}(t^{n-1}) \tag{24}$$

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 is moving to the right with velocity 1. With the standard initialization strategy the limit time step initially is $\Delta t_{\text{CMD}} = h = 1/N$, since a larger time step will cause the left boundary to pass over the position of the first internal node (initially at $x = h$). On the other hand, 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 and also a 2D mesh generation problem are presented. In the whole set of test cases the following convergence criteria had been applied:

- Valid mesh.
- For the k iteration,
$$\frac{|q_{min}^k - q_{min}^{k-1}|}{q_{min}^k} < \epsilon_q$$
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. These values had shown to be very robust. 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 ($V_{min} = \min_e V_e$). As $h(V)$ is a strictly increasing function, then

$$h_{min} = h(V_{min}) = \frac{1}{2}\left(V_{min} + \sqrt{V_{min}^2 + 4\delta^2}\right) \tag{25}$$

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:

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

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: $n = -1$, $\alpha = 1$, $\beta = 0$, $\epsilon_q = 0.01$, $h^*_{min} = 0.1875$ and $\epsilon_\delta = 10^{-6}$. In these tests, the reference element utilized was the regular element and no differential predictor was used.

5.1. Test 1

Figure 7 shows the original domain and the deformation sequence for this problem. The test was solved for a relative deformation of 50%, 90% and 99%. A structured mesh with 200 triangular elements and 121 nodes was employed. The results present the original invalid mesh with inverted elements and the final valid mesh (figures 8 to 13). It is also included the evolution of the minimum quality with iterations (figure 14).

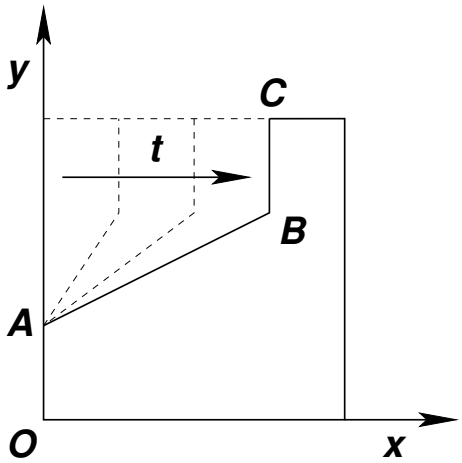

Figure 7. Test 1 - Domain.

5.2. Test 2

This test contains two squares, one inside the other initially centered as observed in figure 15. 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 deformations were solved: 50 %, 90 % and 99 %. The results achieved are presented in figures 16 to 22.

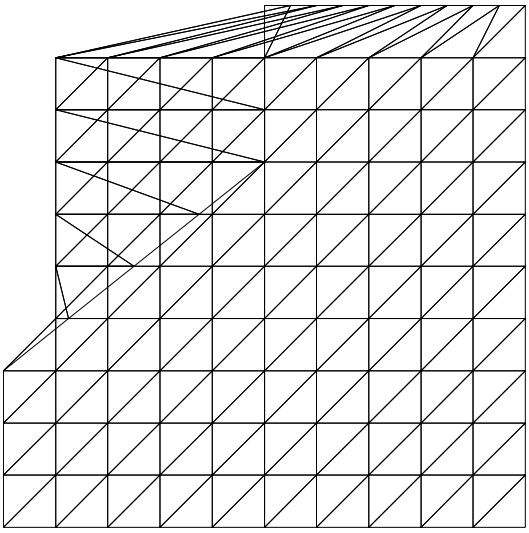

Figure 8. Test 1 - 50 % def. - Initial mesh.

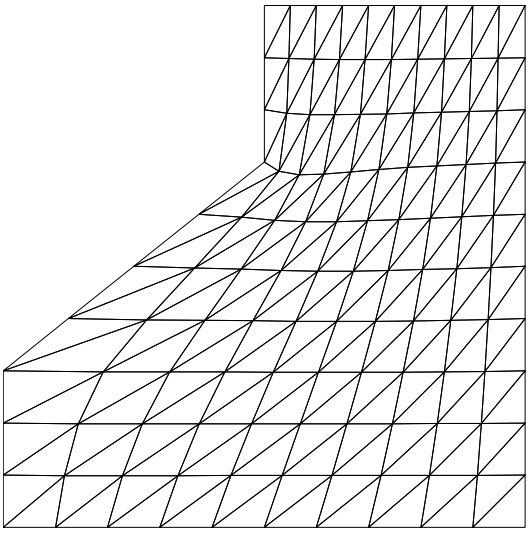

Figure 9. Test 1 - 50 % def. - Final mesh.

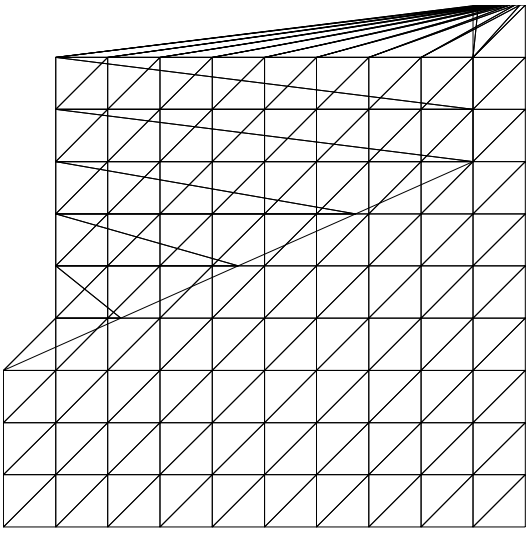

Figure 10. Test 1 - 90 % def. - Initial mesh.

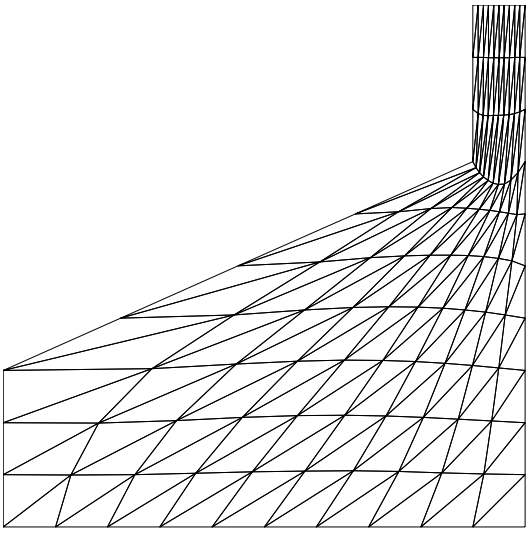

Figure 11. Test 1 - 90 % def. - Final mesh.

5.3. Test 3

This test is the 3D extension of the test presented in subsection 5.1. Figure 23 shows the domain for different deformations. The top face of the cube is moved in the vertical direction. During the deformation, this face is transformed in two planes at different heights joined by a truncated cone with upper radius r and lower radius R.

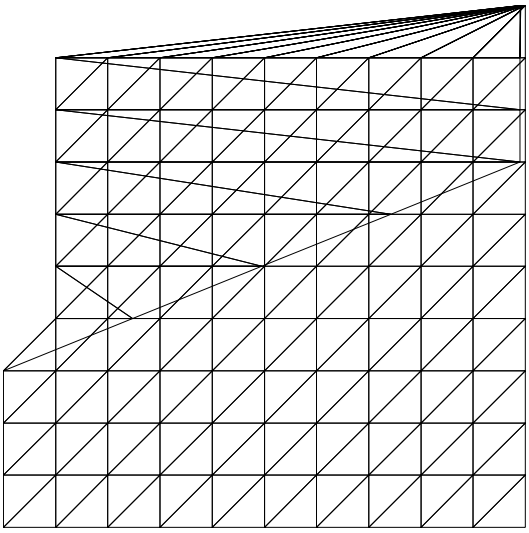

Figure 12. Test 1 - 99 % def. - Initial mesh.

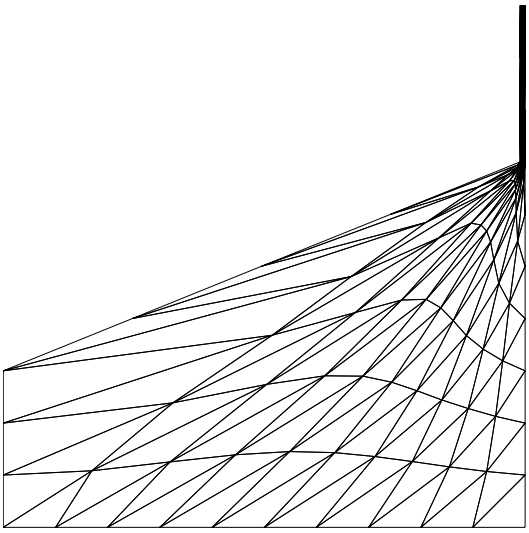

Figure 13. Test 1 - 99 % def. - Final mesh.

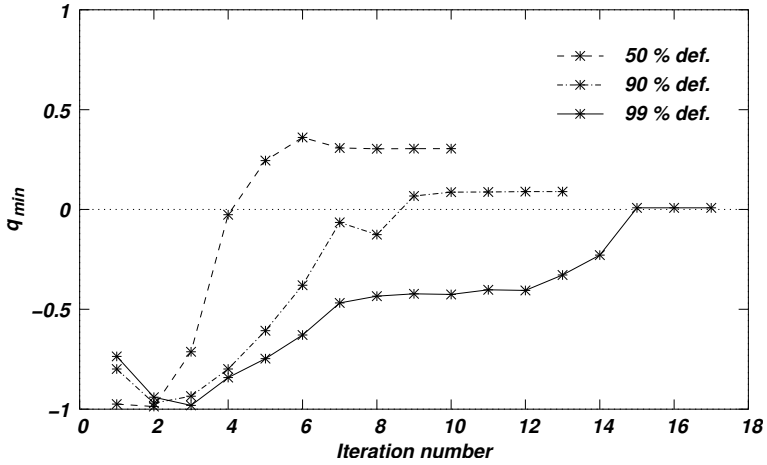

Figure 14. Test 1 - Evolution of q_{min}.

A mesh with 1080 tetrahedral elements and 343 nodes was used. This test was solved for $50\%, 80\%$ and 87% of relative deformation. Figures 24 to 30 show the results achieved.

5.4. Test 4

This test was taken from [31] and consists in a unit cube with 625 tetrahedras and 216 nodes. The invalid initial mesh was obtained transforming the cube to another cube of 10 units of side length changing the coordinates of some nodes in the following way: the internal nodes

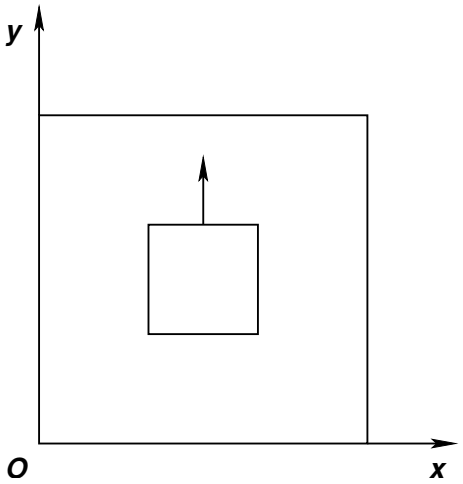

Figure 15. Test 2 - Domain.

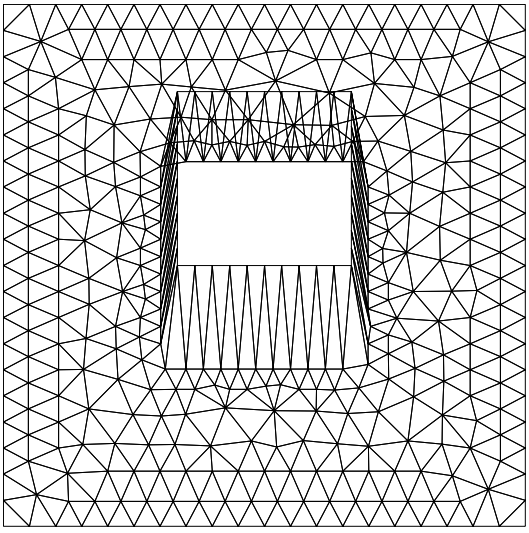

Figure 16. Test 2 - 50 % def. - Initial mesh.

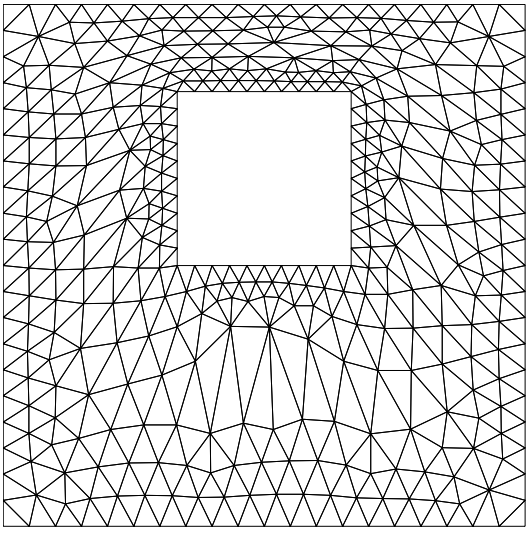

Figure 17. Test 2 - 50 % def. - Final mesh.

are kept fixed, those nodes lying on the edges are relocated on the edges of the new cube and those nodes lying on the faces of the original cube are projected on the new faces respectively. (see figure 31). In figure 32 the final mesh obtained is shown.

In figure 33 the minimum quality as a function of the number of iterations is presented.

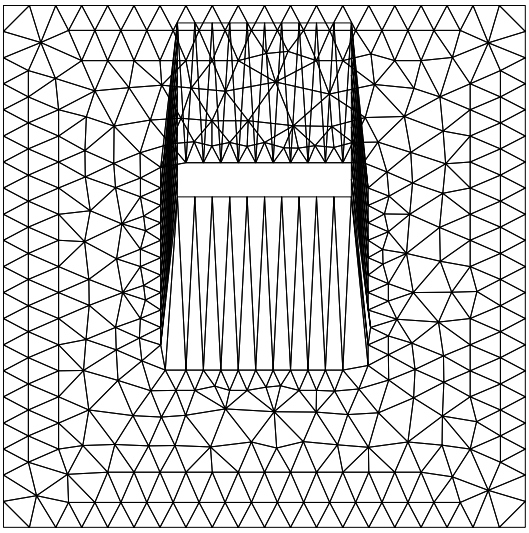

Figure 18. Test 2 - 90 % def. - Initial mesh.

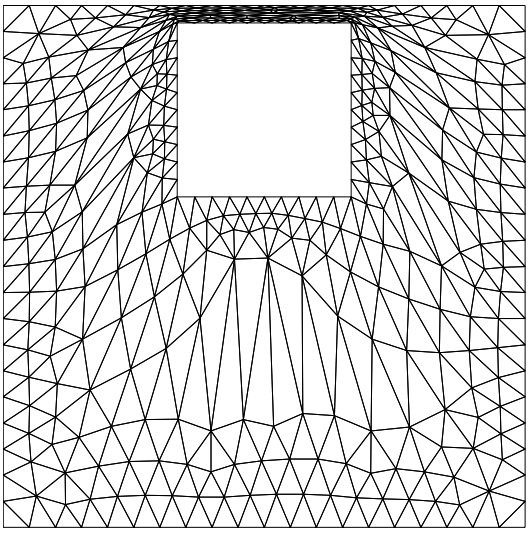

Figure 19. Test 2 - 90 % def. - Final mesh.

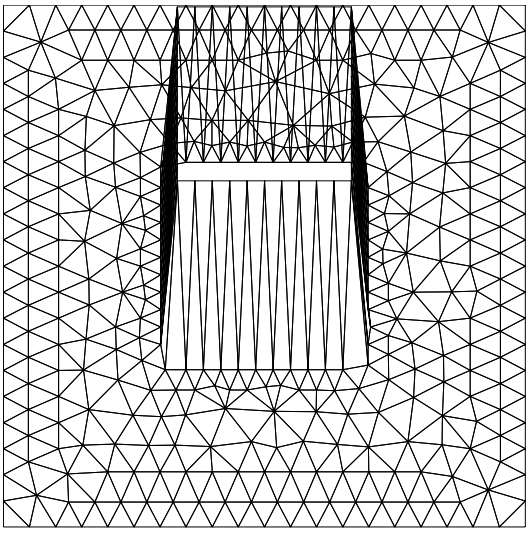

Figure 20. Test 2 - 99 % def. - Initial mesh.

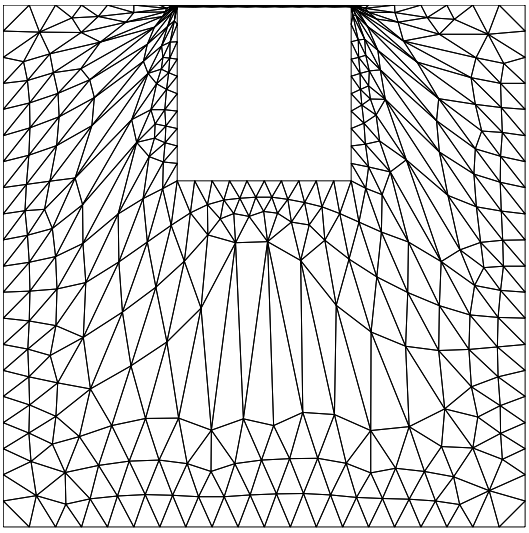

Figure 21. Test 2 - 99 % def. - Final mesh.

5.5. Mesh generation

As it was previously mentioned, the strategy of simultaneous untangling and smoothing may serve as an algorithm for mesh generation. Here, a 2D test example is presented to give an idea about its potentiality. This case was extracted from [33] and it consists to find 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

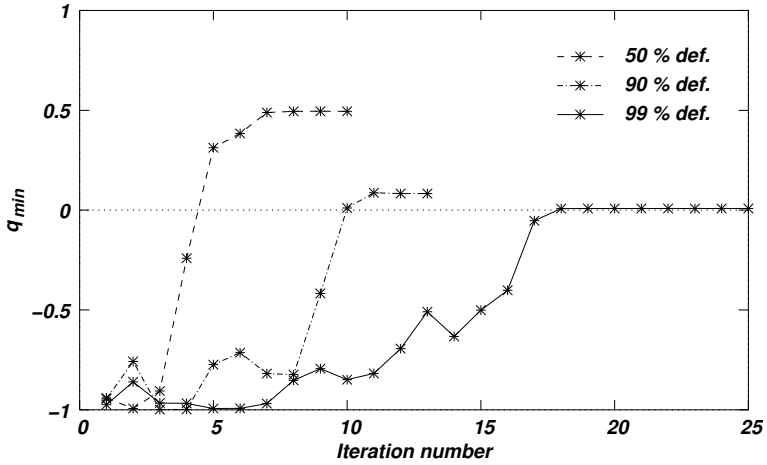

Figure 22. Test 2 - Evolution of q_{min}.

objective, and initial mesh composed by 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 is applied like in mesh dynamics problems. The achieved results are shown in figures 34 and 35.

5.6. Supersonic flow past a highly flexible structure

The above presented strategy was also applied to a FSI test [34]. This test consists of a two dimensional supersonic flow impinging on a flexible plate of a finite thickness that undergoes large deformations. A 65900 triangular elements and 36000 nodes mesh was used to cover the whole domain. Initially, the mesh kinematics obtained from a typical FSI solution was employed. Some deformed meshes are shown in figures 36 to 39. The respective pressure field at each time are presented in figures 40 to 43. Applying the basic smoothing strategy, the computational cost due to the mesh dynamics part was 25 % of the total CPU time. Because of the large mesh deformation, the time step size of the simulation was limited by the CMD strategy in order to guarantee the validity of the mesh for all the time interval. In this way, the CFD code had to reduce the Courant number to assure that the mesh did not collapse, increasing the computational cost.

Being the target to assess the simultaneous untangling and smoothing technique presented when it is applied to this FSI problem, the same sequence of deformed meshes obtained with the original technique was taken. Starting with the mesh at the time step t_1 a new mesh for the structure deformation for $t_2 > t_1$ was computed. The robustness of this method was assessed according to the maximum time step size able to be used to produce good quality meshes. In this way it is possible to remove the time step restriction due to the CMD leaving this size only imposed by physical and numerical features of the real problems involved.

With the proposed strategy a Courant number 100 times greater than the original one imposed by the CFD criterion can be used. The results achieved may be observed in figures 44

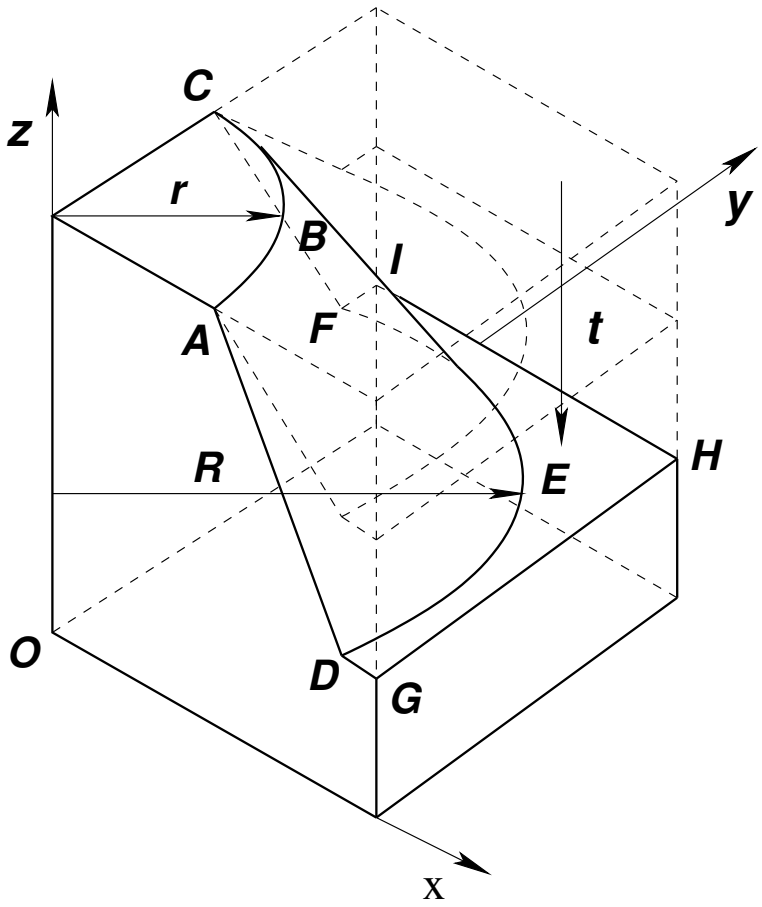

Figure 23. Test 3 - Problem definition.

and 45. This figures show the minimum quality and the number of iterations to get a converged mesh as a function of time.

The computational cost of this strategy was not measured directly. However, an estimation of it may be viewed from figure 45. Each iteration within the nonlinear loop has a cost slightly greater than that obtained for the original smoothing technique. This little extra cost is due to the line search and the evaluation of the element quality at each iteration. The new strategy takes between 1 to 16 iterations to get the final mesh compared with 100 iterations used to obtain a similar mesh with the original method. This comparison goes in favour of the simultaneous untangling and smoothing technique over the original standard smoothing.

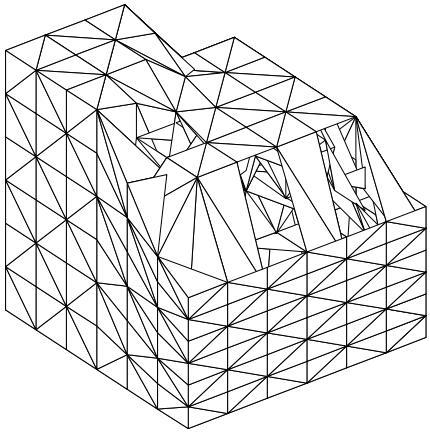

Figure 24. Test 3 - 50 % def. - Initial mesh.

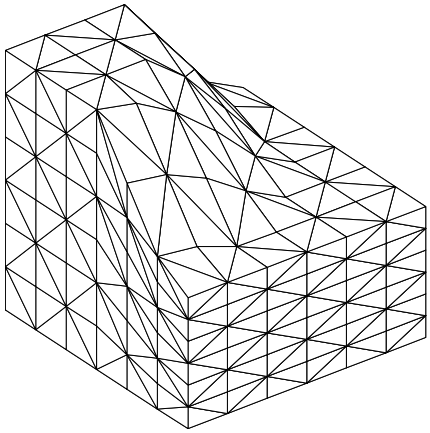

Figure 25. Test 3 - 50 % def. - Final mesh.

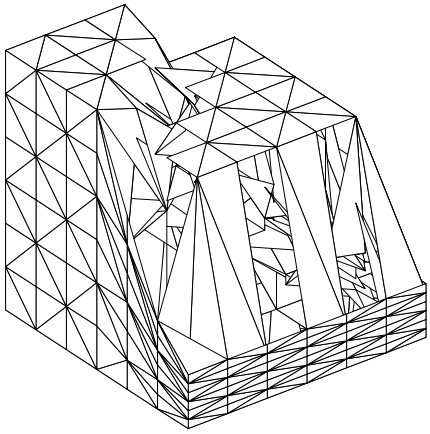

Figure 26. Test 3 - 80 % def. - Initial mesh.

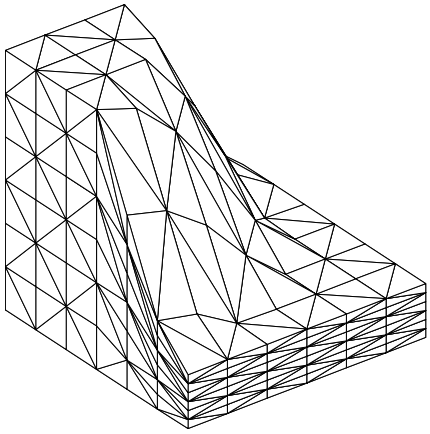

Figure 27. Test 3 - 80 % def. - Final mesh.

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 several tests in both 2D and 3D with medium, large and extra large deformation it may be concluded 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 where one does not

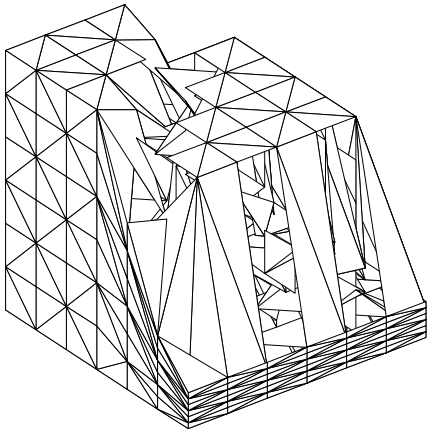

Figure 28. Test 3 - 87 % def. - Initial mesh.

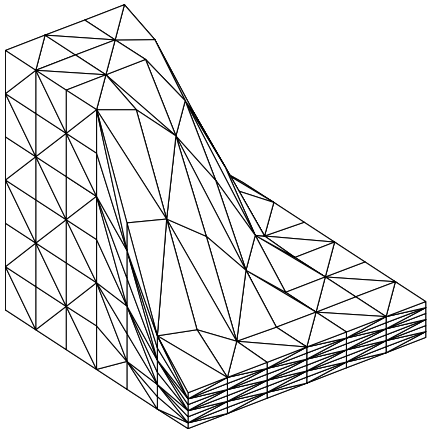

Figure 29. Test 3 - 87 % def. - Final mesh.

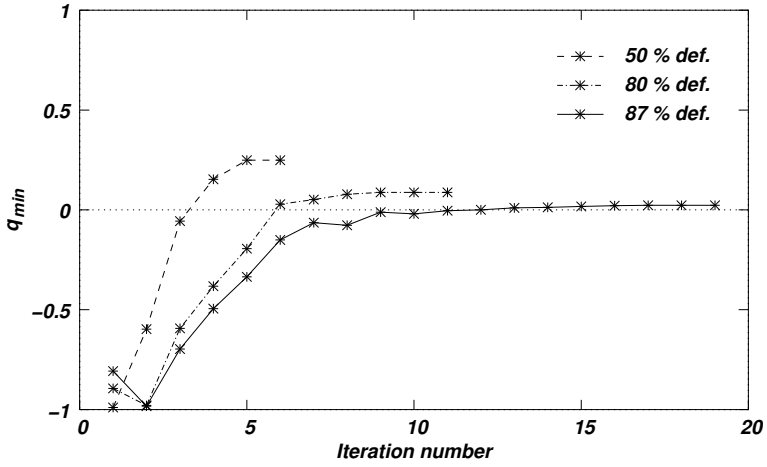

Figure 30. Test 3 - Evolution of q_{min}.

expect that this stage of the whole procedure be more restrictive than any of the others two. However, in several applications the refinement imposes to reduce the time step size to the mesh dynamics in order to avoid the element inversion. Enhancing 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 more expensive than in the original mesh smoothing strategy. But, taking into account that this new procedure does not alter the time step size in FSI problems, in general it makes the global computational cost cheaper and more robust. As a side effect this proposed technique

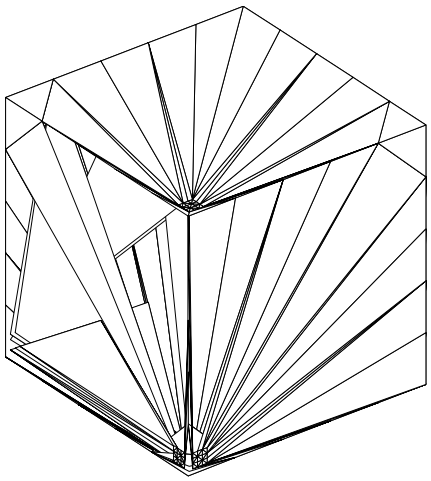

Figure 31. Test 4 - Initial mesh.

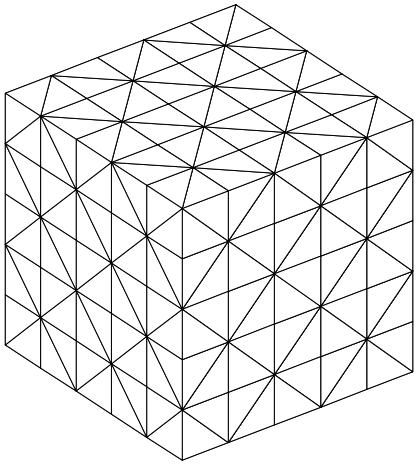

Figure 32. Test 4 - Final mesh.

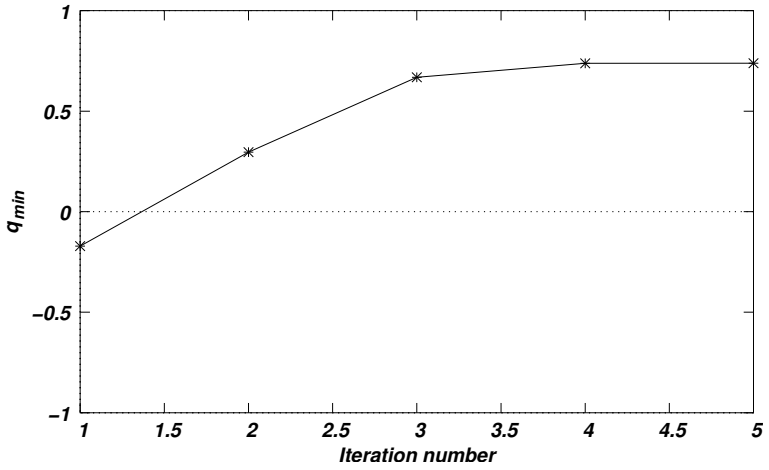

Figure 33. Test 4 - Evolution of q_{min}.

was successfully applied to mesh generation of a simple 2D domain. In future works it is expected to prove this capability to others 3D mesh in more complex domains, and also to extend the functional to get nearly conformal meshes to be applied in airfoil meshing. From the mathematical point of view this methodology deserves more work.

REFERENCES

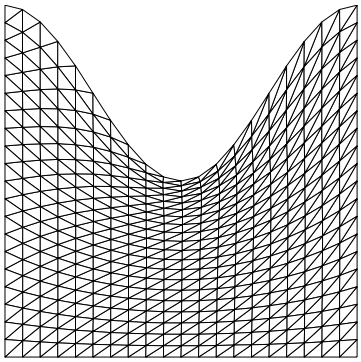

Figure 34. Final mesh.

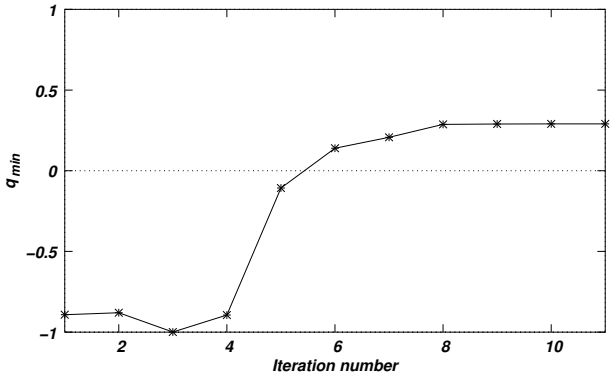

Figure 35. Evolution of q_{min}.

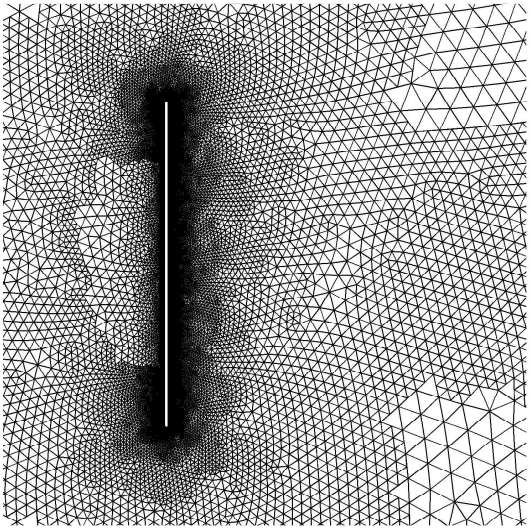

Figure 36. FSI Problem - Initial mesh.

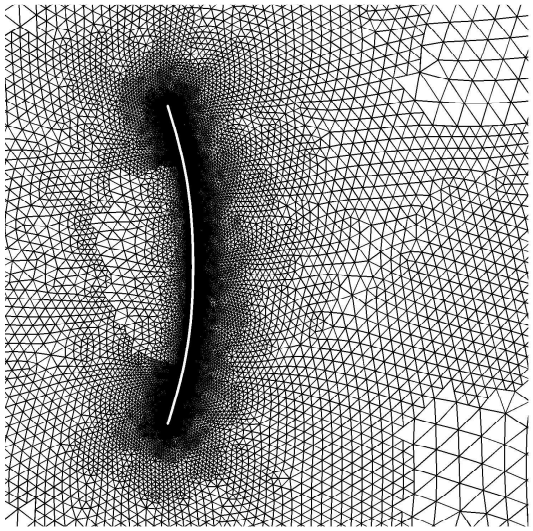

Figure 37. FSI Problem - Deformed mesh at $t = 3.75 \times 10^{-4}$.

1. 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.
2. 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.
3. 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.
4. Donea J, Giuliani S, Halleux JP. An arbitraty, Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering* 1982; **33**: 689–700.
5. 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.

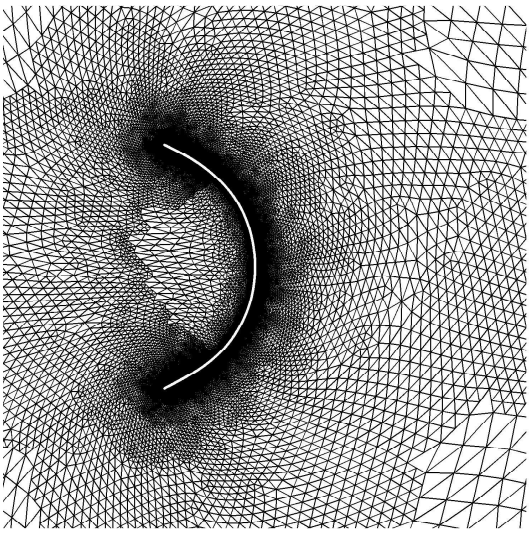

Figure 38. FSI Problem - Deformed mesh at $t = 7.1 \times 10^{-4}$.

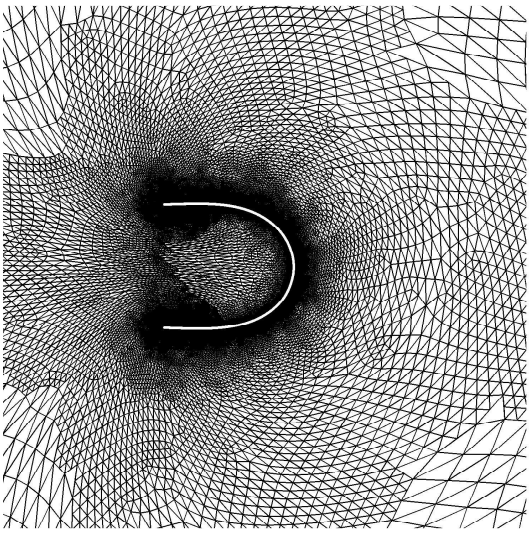

Figure 39. FSI Problem - Deformed mesh at $t = 1.15 \times 10^{-3}$.

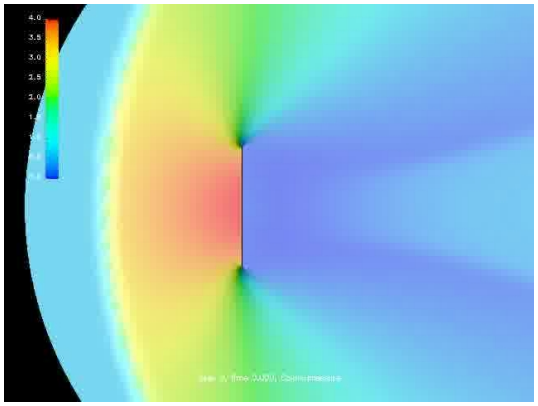

Figure 40. FSI Problem - Pressure at $t = 0$.

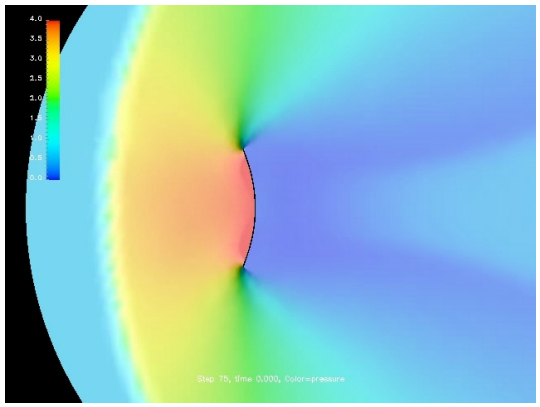

Figure 41. FSI Problem - Pressure at $t = 3.75 \times 10^{-4}$.

6. 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.
7. 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.
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. 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.
10. Stein K, Tezduyar T, Benney R. Automatic mesh update with the solid-extension mesh moving technique. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**: 2019–2032.

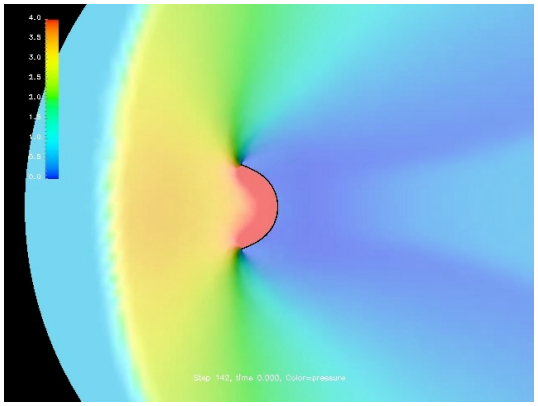

Figure 42. FSI Problem - Pressure at $t = 7.1 \times 10^{-4}$.

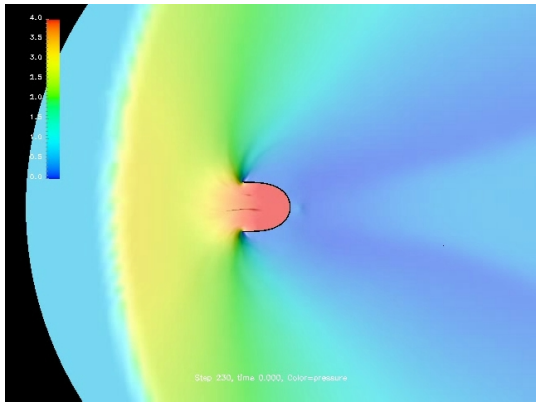

Figure 43. FSI Problem - Pressure at $t = 1.15 \times 10^{-3}$.

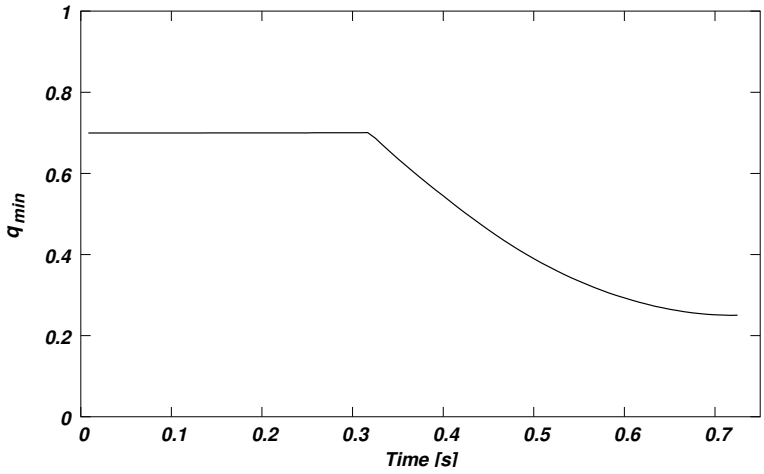

Figure 44. FSI Problem - Evolution of q_{min} as a function of time.

11. 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.
12. Blom FJ. Considerations on the spring analogy. *Numerical Methods in Fluids* 2000; **32**(6): 647–668.
13. 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.
14. Kjellgren P, Hyvärinen J. An arbitrary Lagrangean-Eulerian finite element method. *Computational Mechanics* 1998; **21**: 81–90.
15. Löhner R, Yang C. Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering* 1996; **12**(10): 599–608.
16. 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.
17. 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.

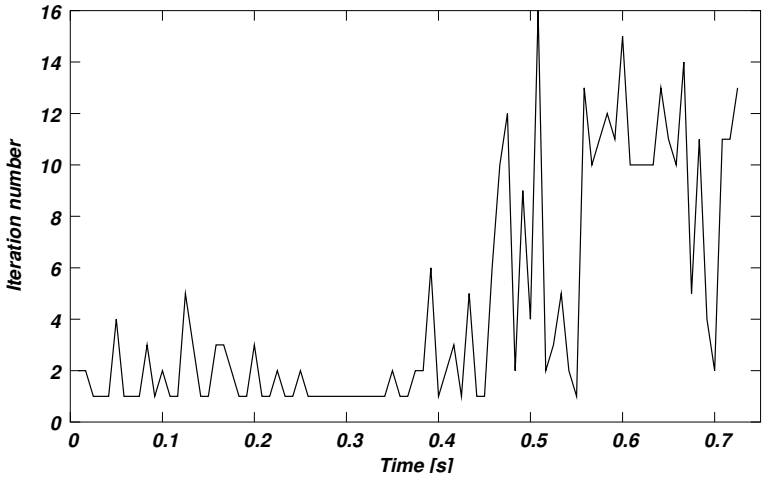

Figure 45. FSI Problem - Iteration number as a function of time.

18. Winslow AM. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics* 1997; **135**(2): 128–138.
19. Knupp PM. Winslow smoothing on two-dimensional unstructured meshes. *Engineering with Computers* 1999; **15**(3): 263–268.
20. Amenta N, Bern M, Eppstein D. Optimal point placement for mesh smoothing. *Journal of Algorithms* 1999; **30**(2): 302–322.
21. 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.
22. 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 Seventh International Meshing Roundtable*, Dearborn, MI, 1998; pages 479–494.
23. Parthasarathy VN, Kodiyalam S. A constrained optimization approach to finite element mesh smoothing. *Finite Elements in Analysis and Design* 1991; **9**: 309–320.
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. Freitag LA, Knupp PM. Tetrahedral element shape optimization via the jacobian determinant and condition number. *Proceedings of the 8th International Meshing Roundtable*, Sandia National Laboratories, 1999; pages 247–258.
26. López EJ, Nigro NM, Storti MA, Toth JA. A minimal element distortion strategy for computational mesh dynamics. *International Journal for Numerical Methods in Engineering* 2006; (accepted).
27. Knupp P. Hexahedral mesh untangling and algebraic mesh quality metrics. *Proceedings of 9th International Meshing Roundtable* 2000; pages 173–183.
28. Freitag LA, Plassmann P. Local optimization-based simplicial mesh untangling and improvement. *International Journal for Numerical Methods in Engineering* 2000; **49**(1): 109–125.
29. Delanaye M, Hirsch Ch, Kovalev K. Untangling and optimization of unstructured hexahedral meshes. *Computational Mathematics and Mathematical Physics* 2003; **43**(6): 807–814.
30. 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.
31. 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.
32. Papalambros PY, Wilde DJ. *Principles of Optimal Design. Modeling and Computation.* Cambridge University Press, 1988.
33. 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.
34. Tam D, Radovitzky R, Samtaney R. An algorithm for modelling the interaction of a flexible rod with a two-dimensional high-speed flow. *International Journal for Numerical Methods in Engineering* 2005; **64**(8): 1057–1077.