

An Improved Algorithm to Smooth Graded Quadrilateral Meshes Preserving the Prescribed Element Size

J. Sarrate
A. Huerta

An improved algorithm to smooth graded quadrilateral meshes preserving the prescribed element size *

Josep Sarrate and Antonio Huerta[†]

*Departament de Matemàtica Aplicada III
E.T.S. de Ingenieros de Caminos, Canales y Puertos
Universitat Politècnica de Catalunya
Campus Nord, E-08034 Barcelona, Spain*

March 29, 2000

Abstract

In the generation of quadrilateral unstructured meshes, special attention is focused to the shape of the elements. This is because it is well known that the distortion of the elements and the accuracy of the analysis are closely related. However, in adaptive schemes it is also essential that the new generated mesh meets the prescribed element sizes in order to obtain a solution with the desired precision. In 1982 Giuliani developed a robust rezoning algorithm based on geometrical criteria.¹ It gives proven results in a smooth element size distribution, but elements do not verify the prescribed element size when sharp distributions appear. This paper presents a modification of the Giuliani method that generates non-distorted elements while preserving the element size. Similar to the original method, this modification can be extended to three-dimensional cases.

Keywords: Finite elements, mesh generation, mesh smoothing techniques, unstructured meshes, quadrilateral elements.

*Contract grant sponsor: Ministerio de Educación y Cultura; Contract grant number: TAP98-0421

[†]*e-mail:* antonio.huerta@upc.es, *web:* <http://www.upc.es/ma3/lacan.html>

1 Introduction

The efficiency of a h -adaptive strategies relies in two ingredients. First, computing an accurate bound of the error, from which the desired element sizes are deduced. Second, generating a new grid with well shaped elements of the prescribed size. It is important to note that the verification of the element size plays a basic role in this kind of processes, because it is assumed that the error of the finite element solution is proportional to $h^{f(p)}$, where h is the characteristic element size, p is the degree of the interpolation polynomial and f is a some positive function of p , see² for a detailed discussion on remeshing techniques.

Triangular elements are extensively used in h -adaptive techniques. However, in several plasticity applications as well as in some incompressible fluid formulations quadrangular elements are preferred.

Quadrilateral mesh generation algorithms^{3-7,9,10} initially yield meshes with very distorted elements. Therefore, mesh quality enhancement procedures are needed in order to improve the overall mesh quality. There are two basic ways to meet this goal. The first one, often called *make-up* techniques, is focused in the improvement of the mesh topology. The second one, called *mesh smoothing*, improves the shape of the elements by modifying the position of the inner nodes once the topology is fixed. Concerning the final element size of the mesh, the former plays a minor role because, in general, the smoothing algorithm is able to distribute elements inside the domain. Therefore, special attention has to be focused on the smoothing algorithm.

Nowadays, there exists a wide range of smoothing algorithms. For instance,⁸ extend the scope of the variational methods, widely used for structured grids, to nonstructured triangular meshes. Other commonly used smoothing technique for unstructured meshes is the so-called Laplacian method,^{11,12} which computes the new nodal position solving the Laplace equation. This technique has an important drawback: in non-convex domains, nodes may run outside of it. Techniques to preclude such a pitfall either increase the computational cost enormously or introduce new terms in the formulation that are particular for each geometry. Giuliani¹ developed a new rezoning algorithm based on geometrical criteria. This method modifies the position of every node in order to minimize a geometric-oriented average distortion of elements meeting on it. These modifications are done with an explicit iterative procedure. In this case, nodes cannot depart from the domain because this is an unstable position in terms of distortion and

squeeze.

h -adaptive techniques^{13,14} first compute a solution on a given coarse mesh. Then, a new element size distribution is computed from a local measure of the estimated error. Therefore, it is crucial that the mesh generator preserves the prescribed element size. In this sense, it is essential that the smoothing algorithm also maintains the size. Giuliani method gives proven results for smooth element size distributions, but it yields unsatisfactory meshes when sharp distributions appear. This is due to the fact that zones with high density tend to loose elements after several remeshing iterations at advanced stages of the analysis. The cause of this problem may be found at the heart of the rezoning principle (see¹ for details).

Therefore, it is important to develop a smoothing algorithm that obtains well shaped elements while the prescribed element size is maintained. This is the goal of the present paper.

2 Distortion metric

A basic point for mesh smoothing techniques is how to quantify the mesh quality. The distortion metric developed by Oddy *et al.*¹⁵ is used in this work. It accounts for both shearing and stretching effects. Moreover, it is not affected by rigid body motions and is independent of the element size. It is computed from the Jacobian of the iso-parametric mapping, \mathbf{J} , as

$$D = \sum_{i=1}^2 \sum_{j=1}^2 C_{ij}^2 - \frac{1}{2} \left(\sum_{k=1}^2 C_{kk} \right), \quad (1)$$

where

$$C_{ij} = \frac{1}{|\mathbf{J}|} \sum_{k=1}^2 J_{ki} J_{kj}. \quad (2)$$

Notice that four-noded linear isoparametric elements are assumed. According to previous analysis,^{15,16} the distortion metric is evaluated at the nodes in an element and the highest value is chosen to represent the quality of the element. If $|\mathbf{J}|$ becomes null or negative in an element, then D is set to an arbitrarily large positive value.

In order to visualize the distortion measure defined in equation (1), Figure 1.a and 1.b show the distortion values corresponding to the effect of shearing and stretching an square element respectively. Note that D is zero for a square element.

3 Improved algorithm

In the original Giuliani method, an influence domain is defined for every node P_i of the unstructured quadrilateral mesh. This domain is defined by the set of triangles obtained by joining all nodes connected to node P_i via the element sides (dashed area in figure 2.a). The distortion of each triangle is defined in terms of its height, h , the average height in the influence domain, \bar{h} , the distance between vertex P_i and the midperpendicular of its opposite side, d , and the average length of the opposite edges to point P_i in the influence domain, \bar{b} (Figure 2.b shows a graphical interpretation of the basic elements of a generic triangle). This distortion reads

$$D = \left(\frac{h - \bar{h}}{\bar{h}} \right)^2 + \left(\frac{2d}{\bar{b}} \right)^2. \quad (3)$$

The new position of P_i is found by minimizing the sum of distortions in the influence domain. This is iteratively repeated for all the nodes in a Gauss-Seidel like procedure, until convergence is achieved (see¹ for details on the implementation of the algorithm). The redistribution of nodal density mentioned above is due to the presence of the mean height \bar{h} in the expression of the distortion, which tends to equalize the size of all the triangles to the mean size in each influence domain.

In order to overcome this problem an improved algorithm is developed. The basic idea is to replace \bar{h} by the theoretical height of a triangle rectangle and isosceles on P_i . This height is $b/2$ when the length of the opposite side to P_i is b (Figure 2.b). Note that if each triangle in the influence domain were rectangle and isosceles on P_i , then the mesh would be structured. A straightforward implementation of this modification is to substitute \bar{h} by $b/2$ in the original algorithm.

In order to stop the iterative procedure the following criterion is used: the maximum relative displacement must be less than a given tolerance. That is, for each iteration and node P_i the following values are computed: the displacement, δ_i , of P_i during this iteration and the shortest element edge, ℓ_i , in contact with P_i . The smoothing algorithm is stopped when:

$$Q \equiv \max_{i=1, \dots, np} \left\{ \frac{\delta_i}{\ell_i} \right\} \leq T_{ol} \quad (4)$$

where np is the total number of nodes and T_{ol} is the prescribed tolerance. Usually, T_{ol} is prescribed as $0.5 \cdot 10^{-n}$ where n is the number of significant

digits desired for the position of a node. Note that in practical computations 1 or 2 significant digits are enough for a reasonable description of the mesh.

Numerical experiments show that the modified algorithm has a higher rate of convergence than the original algorithm. For instance, figure 3 plots the logarithm of Q versus the number of iterations for both algorithms and the mesh presented in the third example of next section. The prescribed tolerance in this case is $T_{ol} = 0.5 \cdot 10^{-3}$. Note that the modified algorithm generates a significant reduction of Q during the initial iterations. For instance, if one significant digit is desired for the new position of a node ($T_{ol} = 0.5 \cdot 10^{-1}$), which is a reasonable value for practical purposes, only 16 iterations are needed with the modified algorithm, whereas the original algorithm needs 92 iterations. Moreover, the ratio of convergence of the modified algorithm is 1.34 faster than the original one. Similar behavior has been detected when both algorithms were applied to other meshes.

This modification can be extended to three-dimensional meshes. In these cases, the influence domain is composed by the set of tetrahedrons obtained by joining all nodes connected to node P_i via the elements sides. The expression of the distortion metric (3) is still valid, being h the height of the tetrahedron and \bar{h} the mean height in the influence domain (see¹ for details). The idea is now to use the height each tetrahedron would have if it were rectangle and ‘isosceles’ on P_i (see figure 4). It is straightforward to see that this height can be expressed in terms of the area of the opposite side, A , as $h = \sqrt{(2A)/(3\sqrt{3})}$.

4 Numerical examples

In order to show the performance of the improved algorithm, four examples are presented in this section. The objective of the first one is to show that the robustness of the method is still preserved. Figure 5.a presents a very distorted initial mesh. It is smoothed using the improved algorithm, and the optimal mesh for this case is obtained in few iterations (see figure 5.b). A L-shape domain with a inner node placed outside of the domain is presented in figure 5.c. The smoothed mesh is presented in figure 5.d. Notice that, even in non-convex domain cases, the modified algorithm still places nodes inside the domain.

The second one is a simple comparison between the original and modified algorithm. The original domain is a unit square meshed into four quadrilat-

erals: two squares of sizes 0.25 and 0.75 and two rectangles (see figure 6.a). Although the two square elements have no distortion ($D = 0$), the mesh can be improved because the two rectangle elements are stretched ($D = 3.55$), see figure 6.b. The smoothed mesh obtained using the original algorithm is shown in figure 6.c and the associated distortion is presented in figures 6.d. The distortion range is $[0.84, 3.55]$ and the mean value of the distortion is $\bar{D} = 2.46$. Figures 6.e and 6.f show the smoothed mesh and the distortion of its elements when the modified algorithm is used. The distortion range is now $[0.88 \cdot 10^{-2}, 3.55]$. It can be appreciated how the original algorithm tends to equalize the element size, while the modified one keeps the density of the original mesh as far as possible. Note that, although the same maximum value is obtained for the distortion measure (1), smaller values are also obtained and the mean value of the distortion is now $\bar{D} = 1.91$.

The goal of the third example is to show that the modified algorithm preserves the element size in the regions where small values are prescribed. In this case the domain is a square of length 5. It is discretized using the a mesh generator algorithm previously developed.^{9,10} A constant element side $h = 0.01$ is prescribed on the bottom while $h = 1$ on the upper side. The prescribed element size value over the domain is computed as linear interpolation of the prescribed values on the lower and upper sides. Figure 7.a shows the mesh before any smoothing technique is applied. It can be observed that highly distorted elements appear. The smoothed meshes using the original and the modified algorithm are presented in figures 7.b and 7.c respectively. The tolerance used to stop the smoothing algorithms is $T_{ol} = 0.5 \cdot 10^{-2}$. Note that the modified algorithm maintains the prescribed small values of the element size on the base. This feature is highlighted in figures 7.d and 7.e, where a detail near the bottom–right corner of the obtained meshes using the original and modified algorithm are presented. In order to compare the quality of the final meshes, the distributions of the element distortion metric, equation (1), for both algorithms are presented in figure 8.a. Note that less distorted elements are generated with the modified algorithm (there is a difference of two orders of magnitude in the number of elements with small values of the distortion measure between both algorithms). Moreover, the mean value of the element distortion using the modified algorithm is $\bar{D} = 1.13$ whereas using the original algorithm it is $\bar{D} = 2.87$. This is because no stretched elements are generated near the bottom side. In order to measure if the prescribed element size is verified, the characteristic element size is computed as the square root of the element area. Then, it is compared with the character-

istic element size obtained from the prescribed values. Figure 7.b shows the distribution of the relative error of the characteristic element size obtained using both algorithms. This relative error is defined as the absolute value of $(\textit{obtained size} / \textit{prescribed size}) - 1$. Note that the modified algorithm tends to generate more elements that meet the prescribed values (there is a difference of one order of magnitude in the number of elements with small relative error). Moreover, if the original algorithm is used, the mean value of the obtained relative error is $\bar{r} = 0.44$ and its maximum ($r_{max} = 1.78$) is located at the bottom of the square (precisely the region of computational interest). On the other hand, if the modified algorithm is used, the mean value of the relative error is reduced to $\bar{r} = 0.14$ and its maximum ($r_{max} = 0.82$) is located outside the region of computational interest. Note that a smooth variation in the element size is obtained notwithstanding the remarkable element size gradient in both cases.

In the fourth example an application of the modified algorithm to adaptive computations is presented. A plane strain adaptive analysis of a dam is conducted using the new remeshing strategy developed by Díez and Huerta.² Starting from a initial mesh (see figure 9.a) two meshes are obtained for two different values of the acceptability criterion: $\eta_L = 5\%$ and $\eta_L = 3\%$ (figures 9.b and 9.c respectively) Note that regular and well shaped elements are generated even in a small region where a high gradient of the element size is prescribed.

5 Conclusions

A modification of the smoothing algorithm developed by Giuliani¹ is presented in this paper. It is proved that the new algorithm generates well shaped elements. Moreover, when sharp distributions of the element size are prescribed, in contrast with the original algorithm, it is able to maintain the prescribed element size, specially in the regions where small values are specified. This property is its basic characteristic and makes it reliable when dealing with adaptive techniques.

References

- [1] S. Giuliani, ‘An algorithm for continuous rezoning of the hydrodynamic

- grid in Arbitrary Lagrangian-Eulerian codes’, *Nuclear Engineering and Design*, **72**, No. 2, 205–212 (1982).
- [2] P. Díez and A. Huerta ‘A unified approach to remeshing strategies for finite element h -adaptivity’, *Computers Methods in Applied Mechanics and Engineering*, **176**, No. 1–4, 215–229 (1999).
- [3] J.A. Talbert and A.R. Parkinson, ‘Development of an automatic two-dimensional finite element mesh generator using quadrilateral elements and Bézier curve boundary definition’, *International Journal for Numerical Methods in Engineering*, **29**, 1551–1567 (1990).
- [4] T.D. Blacker and M.B. Stephenson, ‘Paving: a new approach to automated quadrilateral mesh generation’, *International Journal for Numerical Methods in Engineering*, **32**, 811–847 (1991).
- [5] J.Z. Zhu, O.C. Zienkiewicz, E. Hinton and J. Wu, ‘A new approach to the development of automatic quadrilateral mesh generation’, *International Journal for Numerical Methods in Engineering*, **32**, 849–866 (1991).
- [6] E. Rank, M. Schweingruber and M. Sommer, ‘Adaptive mesh generation and transformation of triangular to quadrilateral meshes’, *Communications in Numerical Methods in Engineering*, **9**, 121–129 (1993).
- [7] C.K. Lee and S.H. Lo, ‘A new scheme for the generation of a graded quadrilateral mesh’, *Computers and Structures*, **52**, No. 5, 847–857 (1994).
- [8] J. Cabello, R. Löhner and O.P. Jacquotte, ‘A Variational method for the optimization of directionally stretched elements generated by the advancing front method (AFM)’, *Proceedings of the 3rd. Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* Barcelona, Spain, June 3–7, North Holland 521–532 (1991).
- [9] J. Sarrate, Modelización numérica de la interacción fluido-sólido rígido: desarrollo de algoritmos, generación de mallas y adaptividad, Doctoral Thesis, Universitat Politècnica de Catalunya, Barcelona, (1996).
- [10] J. Sarrate and A. Huerta, ‘Efficient Unstructured Quadrilateral Mesh Generation’ *International Journal for Numerical Methods in Engineering*, (accepted for publication) .

- [11] L.R. Herrmann, ‘Laplacian–isoparametric grid generation scheme’, *Journal of the Engineering Mechanical Division ASCE*, **102**, 749–756 (1976).
- [12] R. Löhner and C. Yang, ‘Improved ALE mesh velocities for moving bodies’ *Communications in Numerical Methods in Engineering*, **12**, 599–608 (1996).
- [13] A. Huerta, A. Rodríguez–Ferran, P. Díez and J. Sarrate ‘Adaptive finite element strategies based on error assessment’, *International Journal for Numerical Methods in Engineering*, **46**, 1803–1818 (1999).
- [14] P. Díez, M. Arroyo and A. Huerta, ‘Adaptivity based on error estimation for viscoplastic softening materials’, *Mechanics of Cohesive and Frictional Materials*, **5**, 87-112 (2000).
- [15] A. Oddy, J. Goldak, M. McDill and M. Bibby, ‘A distortion metric for iso–parametric finite elements’, *CSME* **12**, No. 4, 213–217 (1988).
- [16] S. Hyun and L.E. Lindgren, ‘Mesh smoothing techniques for graded elements’, *Simulation of Materials Processing: Theory and Applications*, Huétink and Baaijens Eds., Rotterdam, 109–114, (1998).

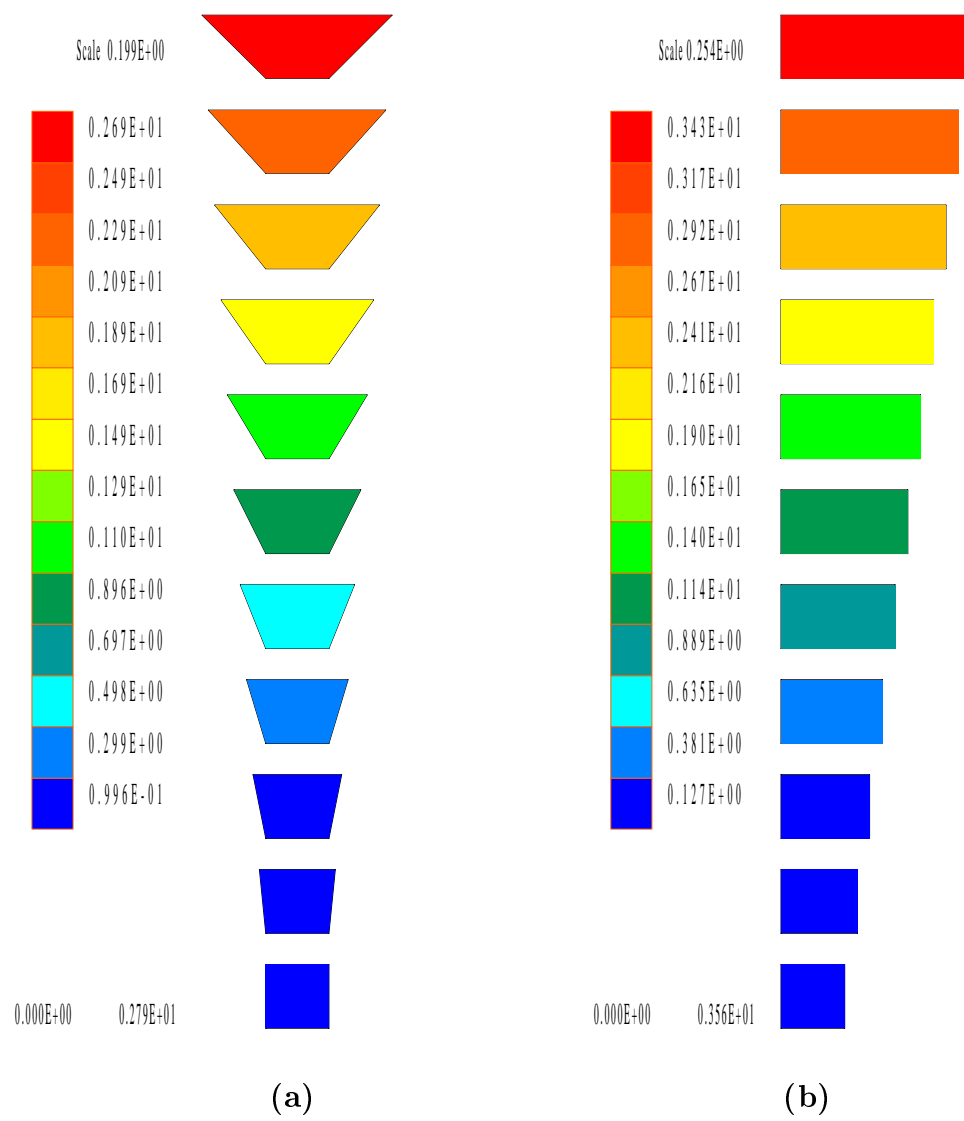


Figure 1: Values of the distortion metric D due to: **(a)** the shearing of a square element, **(b)** the stretching of a square element.

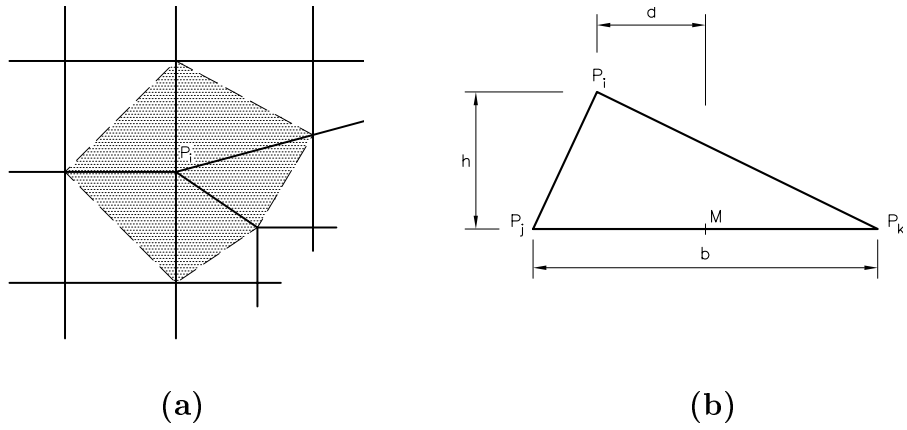


Figure 2: (a), Representation of the influence domain (shaded) of node P_i (b), Basic elements of a generic triangle

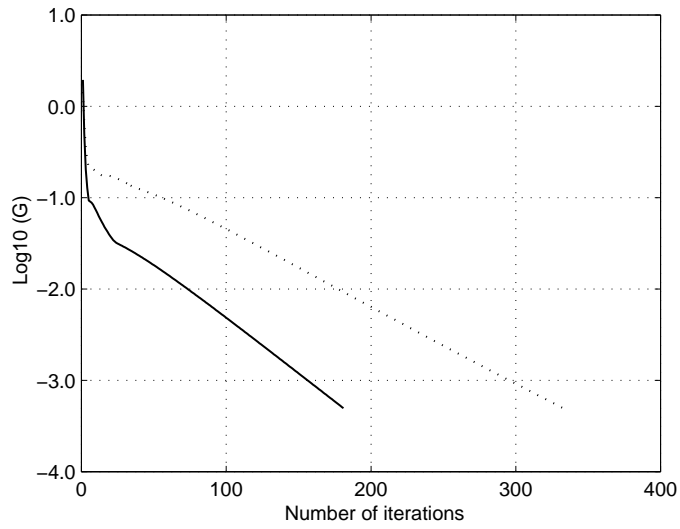


Figure 3: Comparison of the rate of convergence of the smoothing algorithms: original algorithm (dot line) and modified algorithm (solid line).

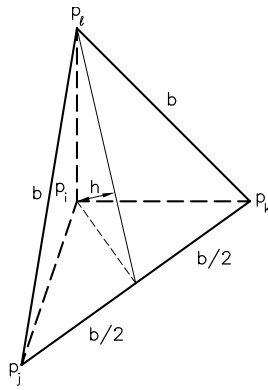
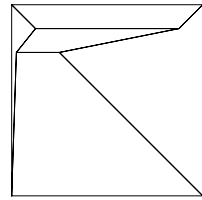
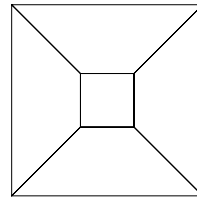


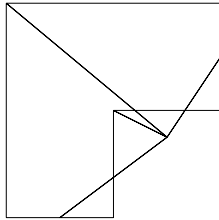
Figure 4: Basic elements of a tetrahedron in the influence domain of node P_i



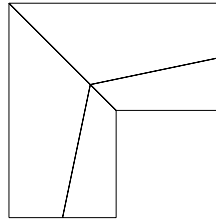
(a)



(b)



(c)



(d)

Figure 5: Robustness of the rezoning algorithm: (a) initial mesh for a square domain, (b) final mesh for a square domain, (c) initial mesh for a L-shape domain, (d) final mesh for a L-shape domain.

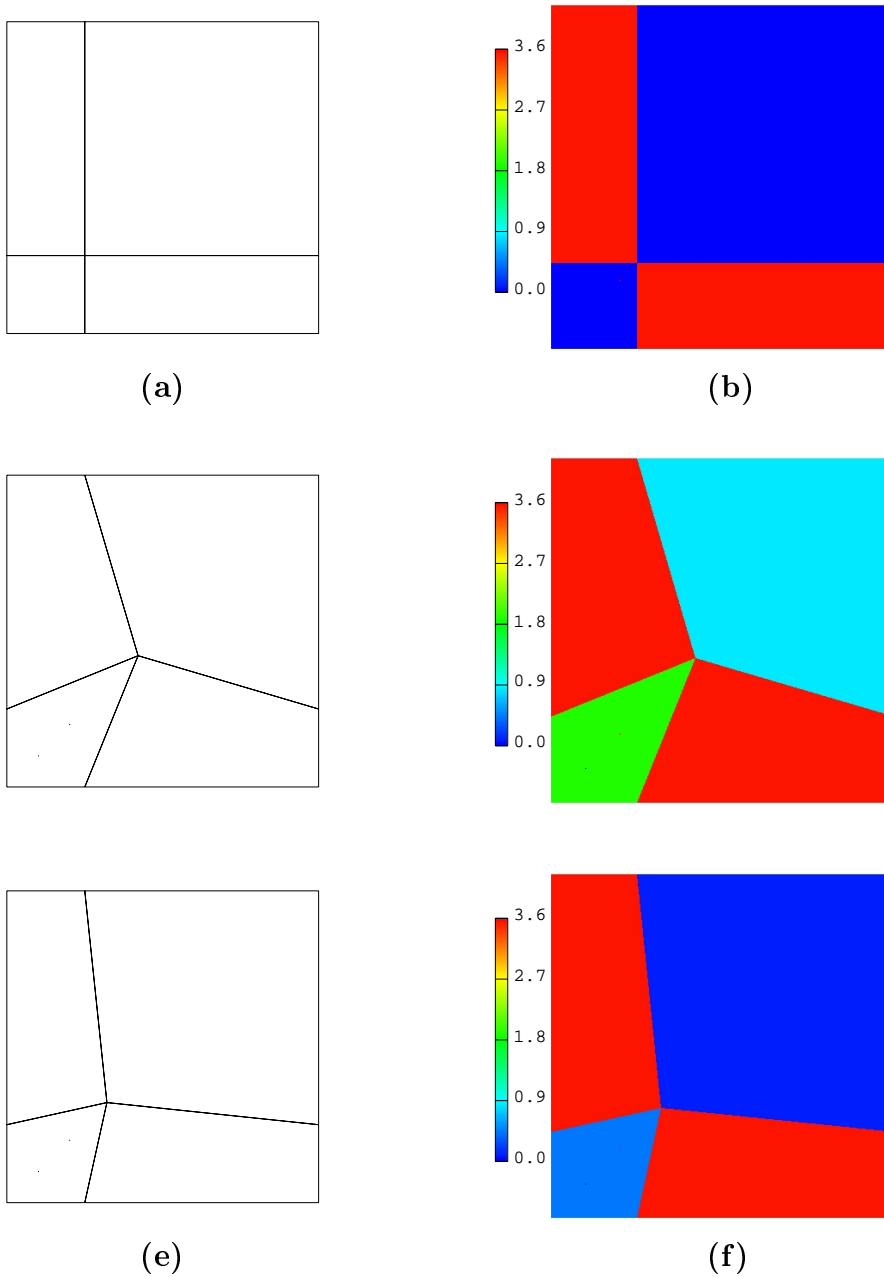
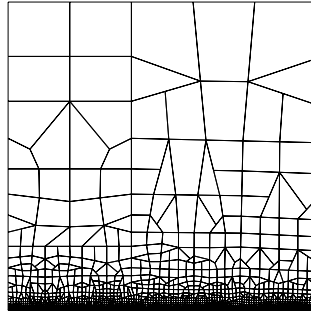
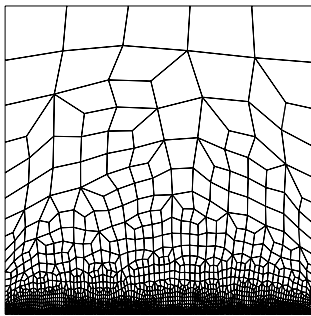


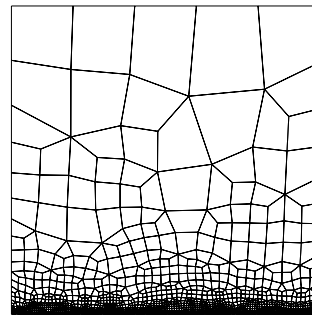
Figure 6: Comparison of the obtained results with the original and the modified smoothing algorithm: **(a)** initial mesh, **(b)** distortion distribution over the initial mesh, **(c)** smoothed mesh using the original algorithm, **(d)** distortion distribution obtained using the original algorithm, **(e)** smoothed mesh using the modified algorithm, **(f)** distortion distribution obtained using the modified algorithm.



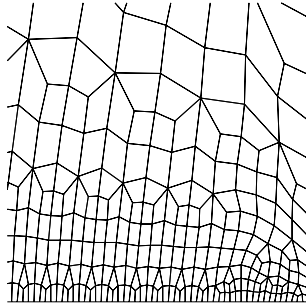
(a)



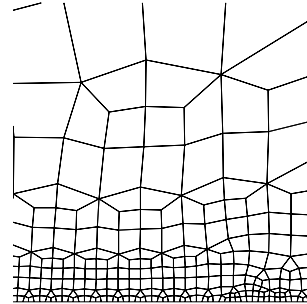
(b)



(c)

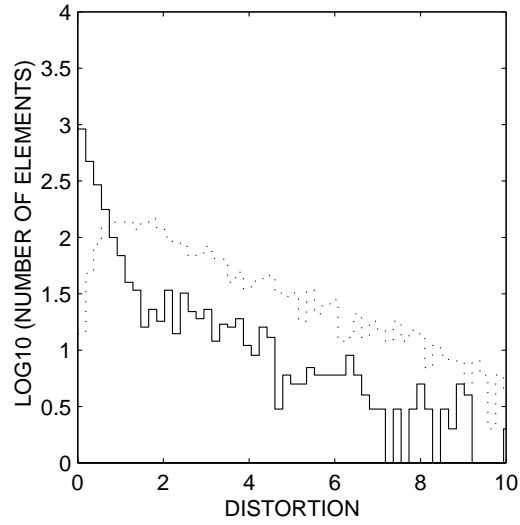


(d)

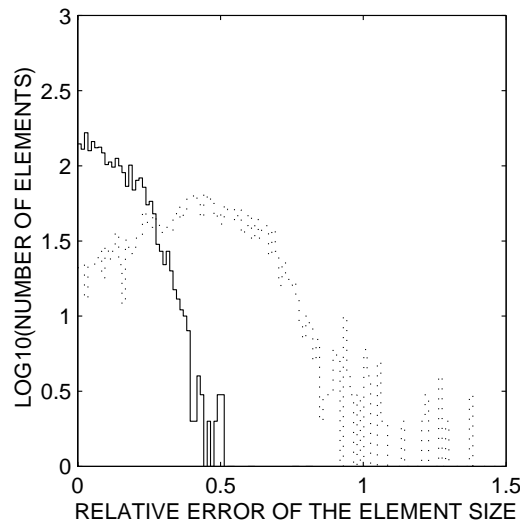


(e)

Figure 7: Comparison of the obtained results with the original and the modified smoothing algorithm: (a) initial mesh, (b) smoothed mesh using the original algorithm, (c) smoothed mesh using the modified algorithm, (d) detail of the obtained mesh using the original algorithm, (e) detail of the obtained mesh using the modified algorithm.

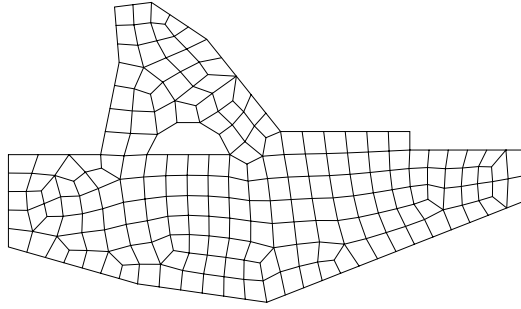


(a)

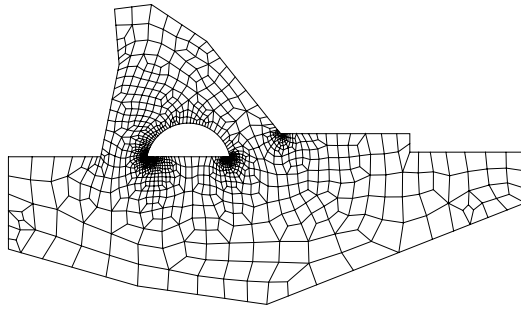


(b)

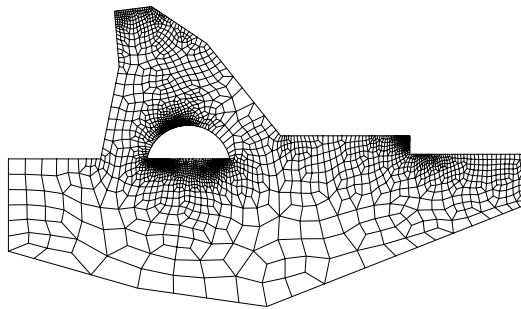
Figure 8: Comparison of the obtained results with the original and the modified smoothing algorithm: **(a)** distribution of the element distortion using the original algorithm (dotted line) and the modified algorithm (solid line), **(b)** distribution of the relative error of the characteristic element size using the original algorithm (dotted line) and the modified algorithm (solid line).



(a)



(b)



(c)

Figure 9: Application of the modified algorithm to adaptive computations: (a) initial mesh, (b) obtained mesh with $\eta_L = 3\%$, (c) obtained mesh with $\eta_L = 5\%$.