CHALMERS
FINITE ELEMENT CENTER

PREPRINT 2002–19

A variable diffusion method for mesh smoothing

Joakim Hermansson and Peter Hansbo

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2002

CHALMERS FINITE ELEMENT CENTER

Preprint 2002–19

A variable diffusion method for mesh smoothing

Joakim Hermansson and Peter Hansbo

CHALMERS

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, December 2002

A variable diffusion method for mesh smoothing
Joakim Hermansson and Peter Hansbo
NO 2002–19
ISSN 1404–4382

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 1000
Fax: +46 (0)31 772 3595
www.phi.chalmers.se

Printed in Sweden
Chalmers University of Technology
Göteborg, Sweden 2002

A VARIABLE DIFFUSION METHOD FOR MESH SMOOTHING

JOAKIM HERMANSSON AND PETER HANSBO

Abstract. In this note we suggest a new approach to mesh smoothing, based on a combination of Laplacian smoothing and Winslow's method. Using our approach, we can avoid mesh folding and maintain/produce element stretch. We also make a comparison with a method due to Giuliani and show that the latter method cannot ensure that folding is avoided.

1. Introduction

Mesh smoothing techniques are used, e.g., in mesh generation processes and in arbitrary Lagrangian–Eulerian (ALE) problems. The main purpose in mesh smoothing is from a given mesh configuration, with an algorithm, automatically move the nodes to new positions and thereby generate a mesh with as good elements as the algorithm can carry out. In ALE applications the computational domain can undergo large deformations, and in order to represent the physical domain and thus the physical problem the mesh has to change shape in the same way as the domain. Here we have two possibilities: create a new mesh or move the nodes with a mesh smoothing algorithm using the boundary nodes as prescribed variables. If feasible, the latter variant is preferable since the computational cost is lower.

There are several mesh smoothing algorithms available, perhaps the simplest being the Laplacian smoother, see [1, 3]. Its simplicity makes it attractive for mesh smoothing purposes, but unfortunately it has the drawback that elements can become folded when the domain boundaries are concave. The problem of folding is less severe in mesh smoothing than in mesh generation, since the points in the mesh are already distributed according to geometrical constraints. Nevertheless, there is usually a marked crowding of nodes in the vicinity of concave boundaries that is highly undesirable. A method that attempts to avoid this problem, based on minimization of geometrical measures (such as the distortion and the squeeze of the elements) was proposed by Giuliani [2]. However, we have also found that the elements can become folded in Giuliani's method in some cases (cf. below).

Date: December 16, 2002.

Key words and phrases. Mesh smoothing, Giuliani smoothing, Winslow smoothing, Laplacian smoothing.

Joakim Hermansson, Department of Applied Mechanics, Chalmers University of Technology, S–412 96 Göteborg, Sweden, *email*: herman@am.chalmers.se

Peter Hansbo, Department of Applied Mechanics, Chalmers University of Technology, S–412 96 Göteborg, Sweden, *email*: hansbo@am.chalmers.se.

A common problem with the mentioned methods is that they cannot maintain element stretches during the smoothing process (if such are present) or produce stretched elements in their original formulation (for earlier examples of such methods, see, e.g., Shashkov and Knupp [7] and Anderson [8]). Our purpose in this paper is twofold: we wish to find simple modifications of the existing Laplacian and Guiliani smoothing methods such that

(1) the reulting methods maintain stretch, and
(2) avoid crowding of points (or folding) near concave boundaries.

For these purposes, we introduce a new smoothing algorithm, which is based on combining the the linear Laplacian smoother and (a modification of) the nonlinear, but related, Winslow's method [1]. A parameter $0 \leq p \leq 1$ has to be chosen by the user, and the two extremes $p = 0$ and $p = 1$ correspond to Laplacian and a modified Winslow smoothing, respectively. We will also incorporate ideas from Reference [3], in the proposed approach and in Giuliani's method, of how to maintain, or produce, element stretches during the mesh smoothing process. The primary application in this work is mesh smoothing on existing meshes, for instance mesh smoothing arising in arbitrary Lagrangian-Eulerian problems or in cases where it is desirable to control the element stretches. We emphasize that this problem is considerably easier than the full mesh generation problem, where typically a logically rectangular domain is mapped to the physical domain. Thus, for example, the original Laplacian smoothing is far from robust enough for mesh generation purposes but can almost always be used to smooth an existing mesh without risk of folding.

2. THE VARIABLE DIFFUSION SMOOTHING METHOD

2.1. **Auxiliary plane.** We begin by recalling the ideas of how to produce or maintain stretched elements from [3]. To incorporate stretched elements in mesh smoothing an auxiliary (χ_1, χ_2)-plane is introduced. In this plane, the element is supposed to be unstretched. The smoothing algorithm is then expressed in its original form in the auxiliary coordinates $\boldsymbol{\chi}$. Solving for $\boldsymbol{\chi}$ we will get the mesh configuration in the auxiliary plane, according to the chosen mesh smoother, but, since we are interested to control the element stretches, we do not actually solve for $\boldsymbol{\chi}$. Instead we incorporate the Jacobian of transformation, i.e., the deformation gradient, in the smoother to map the element geometry from the (χ_1, χ_2)-plane to the (x_1, x_2)-plane, i.e., our physical or final plane where the element may be stretched. Depending on the smoothing algorithm this can be done explicitly or implicitly. The explicit approach is applicable for mesh smoothers which are expressed in position coordinates, while the implicit approach is suitable for mesh smoothers which are expressed as differential equations. In the latter the Jacobians of transformation are invoked in the weak form of the differential equation and are predefined by the user. The Jacobian describes the mapping from the (χ_1, χ_2)-plane to the (x_1, x_2)-plane for an element, and here we have chosen it to contain the element's stretches and its corresponding directions. If the Jacobians are chosen to be the identity matrix the auxiliary and the physical plane will coincide.

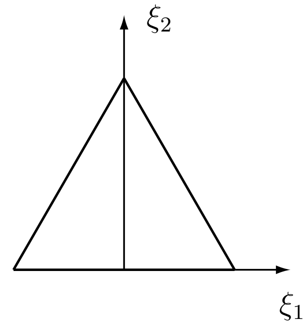

FIGURE 1. The equilateral reference triangle.

2.2. **Winslow and modified Winslow methods.** We start by establishing the equation which defines Winslow smoothing for the auxiliary coordinates $\boldsymbol{\chi}(\boldsymbol{\xi})$. The desired element stretches will after that be built in the map from the auxiliary (χ_1, χ_2)-plane to the physical (x_1, x_2)-plane, following Reference [3], in the weak form of the Winslow equation.

Winslow smoothing is based on solving the problem

$$\alpha \frac{\partial^2 \boldsymbol{\chi}}{\partial \xi_1^2} - 2\beta \frac{\partial^2 \boldsymbol{\chi}}{\partial \xi_1 \partial \xi_2} + \gamma \frac{\partial^2 \boldsymbol{\chi}}{\partial \xi_2^2} = \mathbf{0}, \tag{2.1}$$

where

$$\begin{aligned} \alpha &= \left(\frac{\partial \chi_1}{\partial \xi_2}\right)^2 + \left(\frac{\partial \chi_2}{\partial \xi_2}\right)^2, \\ \beta &= \frac{\partial \chi_1}{\partial \xi_1}\frac{\partial \chi_1}{\partial \xi_2} + \frac{\partial \chi_2}{\partial \xi_1}\frac{\partial \chi_2}{\partial \xi_2}, \\ \gamma &= \left(\frac{\partial \chi_1}{\partial \xi_1}\right)^2 + \left(\frac{\partial \chi_2}{\partial \xi_1}\right)^2, \end{aligned}$$

cf. [1, 4]. We can rewrite problem (2.1) as

$$\nabla_\xi \cdot \boldsymbol{A} \nabla_\xi \chi_i + \boldsymbol{b} \cdot \nabla_\xi \chi_i = 0, \quad i = 1, 2, \tag{2.2}$$

where

$$\boldsymbol{A} := \begin{bmatrix} \alpha & -\beta \\ -\beta & \gamma \end{bmatrix}, \quad \boldsymbol{b} := \left(\frac{\partial \beta}{\partial \xi_2} - \frac{\partial \alpha}{\partial \xi_1}, \frac{\partial \beta}{\partial \xi_1} - \frac{\partial \gamma}{\partial \xi_2}\right).$$

Writing (2.2) in variational form, we seek χ_i such that

$$\int_{\Omega_\xi} \left(\nabla_\xi v_i \cdot \boldsymbol{A} \nabla_\xi \chi_i + v_i \, \boldsymbol{b} \cdot \nabla_\xi \chi_i\right) d\Omega_\xi = 0 \tag{2.3}$$

for all admissible v_i, element-wise on the reference domain, see Figure 1. Since we here deal with meshes built up from triangles the reference element is an equilateral triangle because of its isotropic properties. If the mesh was built up from quadrilateral elements the reference element would be a square. The location and orientation of the reference coordinate system does not matter since the differential equation is (Galilean) invariant.

We now define a new method based on the following numerical experience:

(1) The $\boldsymbol{b}-$term, which makes the method unsymmetric, does not affect the crowding of nodes near concave boundaries.
(2) The modified Winslow method as defined by (2.3) without the $\boldsymbol{b}$–term tends to generate elements of equal size, counteracting local refinement.

With these findings in mind, we propose the following approach: seek χ_i such that

$$\int_{\Omega_\xi} \nabla_\xi v_i \cdot \boldsymbol{A}^p \, \nabla_\xi \chi_i \, d\Omega_\xi = 0 \tag{2.4}$$

for all admissible v_i. Here $0 \leq p \leq 1$ is a free parameter to be chosen by the user. We note that $p = 0$ corresponds to Laplacian smoothing (cf. [3]), and $p = 1$ is a modified Winslow smoothing.

Remark. The omission of $\boldsymbol{b}$ will be of marked importance in the case of *mesh generation*. We will show the effect of including (simplistically) vs. omitting $\boldsymbol{b}$ in the numerical examples.

Remark. Note that $\boldsymbol{A} = \boldsymbol{A}(\nabla_\xi \chi_i)$ can be computed elementwise on a given finite element mesh. $\boldsymbol{A}^p$ can then be computed by finding the eigenvalues and eigenvectors to $\boldsymbol{A}$.

2.3. **Handling stretched meshes.** The relation (2.4) can be used as a smoother if the element sizes are supposed to be equal-sized. To extend the variable diffusion method to stretched meshes, we must also use the map $\boldsymbol{\chi} \to \boldsymbol{x}$, which is given analytically beforehand and represents the stretch. Defining

$$\nabla_\xi \boldsymbol{\chi} := \begin{bmatrix} \dfrac{\partial \chi_1}{\partial \xi_1} \\ \dfrac{\partial \chi_1}{\partial \xi_2} \\ \dfrac{\partial \chi_2}{\partial \xi_1} \\ \dfrac{\partial \chi_2}{\partial \xi_2} \end{bmatrix},$$

we can apply the chain rule to find

$$\nabla_\xi \boldsymbol{\chi} = \boldsymbol{C} \nabla_\xi \boldsymbol{x},$$

where

$$\boldsymbol{C} := \begin{bmatrix} \dfrac{\partial \chi_1}{\partial x_1} & 0 & \dfrac{\partial \chi_1}{\partial x_2} & 0 \\ 0 & \dfrac{\partial \chi_1}{\partial x_1} & 0 & \dfrac{\partial \chi_1}{\partial x_2} \\ \dfrac{\partial \chi_2}{\partial x_1} & 0 & \dfrac{\partial \chi_2}{\partial x_2} & 0 \\ 0 & \dfrac{\partial \chi_2}{\partial x_1} & 0 & \dfrac{\partial \chi_2}{\partial x_2} \end{bmatrix}. \tag{2.5}$$

Thus the final version of (2.4) is to seek $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ such that

$$\int_{\Omega_\xi} \nabla_\xi \boldsymbol{v} \cdot \begin{bmatrix} \boldsymbol{A}^p & 0 \\ 0 & \boldsymbol{A}^p \end{bmatrix} \boldsymbol{C} \nabla_\xi \boldsymbol{x} \, d\Omega_\xi = 0$$

for all $\boldsymbol{v} = \boldsymbol{v}(\boldsymbol{\xi})$.

Now, the geometry on each element is approximated as

$$\boldsymbol{x} \approx \boldsymbol{\varphi}^{\mathrm{T}}(\xi_1, \xi_2)\mathbf{X},$$

where $\boldsymbol{\varphi}$ contains the local shape functions, expressed in the local coordinates (ξ_1, ξ_2), and is given by

$$\boldsymbol{\varphi}^{\mathrm{T}}(\xi_1, \xi_2) = \begin{bmatrix} \varphi_1 & 0 & \varphi_2 & 0 & \cdots & \varphi_n & 0 \\ 0 & \varphi_1 & 0 & \varphi_2 & \cdots & 0 & \varphi_n \end{bmatrix}.$$

Furthermore, $\mathbf{X}^{\mathrm{T}} = [X_1^1, X_2^1, X_1^2, X_2^2, \cdots, X_1^n, X_2^n]$ is the vector containing the new element nodal coordinates and n is the number of element nodes. Thus, $\boldsymbol{A}$ and J involve the two maps $\boldsymbol{\xi} \to \boldsymbol{\chi} \to \boldsymbol{x}$ and can be computed using the element approximation and the analytically given map $\boldsymbol{\chi} \to \boldsymbol{x}$ (which also directly defines $\boldsymbol{C}$).

The nonlinear boundary value problem is solved using the foregoing iteration's solution to update $\boldsymbol{A}$ on each element. The components in the matrix $\boldsymbol{C}$, see (2.5), are still unknown and can be computed from the inverse of the Jacobian. The Jacobian, which defines the mapping $\boldsymbol{\chi} \to \boldsymbol{x}$, is defined as

$$J_{ij} = \frac{\partial x_i}{\partial \chi_j}.$$

This map will be user defined and constructed using the principal element stretches and its principal directions as given, see [3]. The Jacobian (also referred to as the material deformation gradient) is constructed as

$$\boldsymbol{J} = \boldsymbol{Q}^{\mathrm{T}} \begin{bmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{bmatrix} \boldsymbol{Q},$$

where Λ_1 and Λ_2 are the principal stretches and $\boldsymbol{Q} = [\boldsymbol{n}_1, \boldsymbol{n}_2]$ contains the corresponding principal directions. The stretch is defined, e.g., in one dimension as the current length divided by the original length. In order to compute the matrix $\boldsymbol{C}$ we need the inverse of the Jacobian, which can be computed as

$$\boldsymbol{J}^{-1} = \frac{\partial \chi_i}{\partial x_j} = \boldsymbol{Q} \begin{bmatrix} \dfrac{1}{\Lambda_1} & 0 \\ 0 & \dfrac{1}{\Lambda_2} \end{bmatrix} \boldsymbol{Q}^{\mathrm{T}}. \tag{2.6}$$

All tools which are needed to smooth with the variable diffusion smoothing method with stretched elements are now established. Numerical examples are given in Section 4.

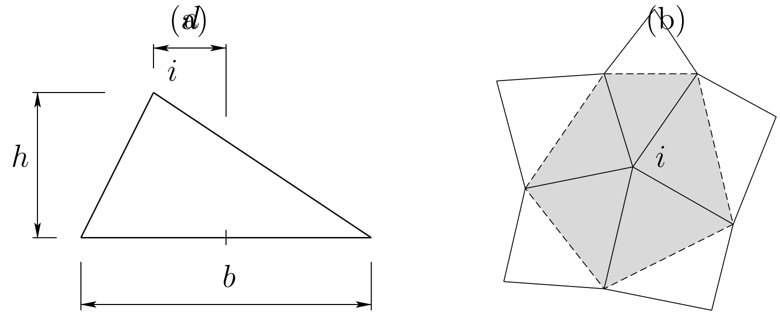

FIGURE 2. (a) A generic triangle. (b) Influence domain on a quadrilateral patch.

3. GIULIANI SMOOTHING OF STRETCHED ELEMENTS

The Giuliani smoothing method is based on minimization of geometrical measures such as the distortion and the squeeze. The smoothing procedure is conducted for each interior node i (or a sliding boundary node) at each time for all movable nodes on the mesh. For each movable node i there is a belonging influence domain, which is defined as the set of triangles which connect node i with its neighboring nodes, see the shadowed area in Figure 1b. The influence domain can be built up from quadrilateral or triangle elements (in two dimensions). The idea is to iteratively minimize the distortion and the squeeze for all influence domains on the mesh. The function to minimize on each influence domain is defined as

$$E = \sum_{N} \left(\frac{h - \bar{h}}{\bar{h}} \right)^2 + \left(\frac{2d}{\bar{b}} \right)^2 ,$$

where N is the number of elements on the influence domain, h the height, d the distortion, b the base, see Figure 1a, and the superscribed bar denotes the average value on the influence domain. The resulting scheme which minimizes the distortion and the squeeze can be seen in Reference [2]. To incorporate the idea in [3] of using a predefined Jacobian of transformation, we choose to transform the node coordinates on each influence domain to the auxiliary plane (χ_1, χ_2) in order to smooth the mesh. The displacements of node i are then transformed back to the physical plane and the node coordinates can be updated. The node coordinates are transformed to the auxiliary plane using the inverse of the Jacobian, see (2.6), as

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{J}\, \mathrm{d}\boldsymbol{\chi} \quad \Rightarrow \quad \boldsymbol{\chi} = \boldsymbol{J}^{-1}\boldsymbol{x} + \boldsymbol{D},$$

where the integration constant can be set to $\boldsymbol{D} = \boldsymbol{0}$. This because the smoothing algorithm only uses differences. To avoid folded elements during the transformation we choose to transform each influence domain using a local coordinate system located at node i. The background for this is that if one of the nodes on the influence domain is located in an area which is defined to be stretched and the other nodes are not, and the domain is located 'far away' from the global origin, elements can become folded. This can also occur if neighboring nodes are located in different zones with a large difference in the stretch. For

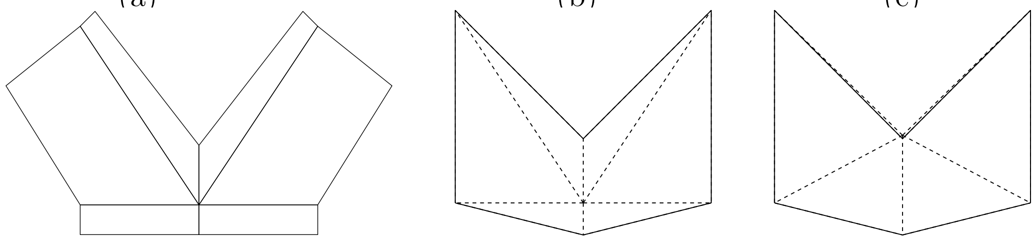

FIGURE 3. A quadrilateral patch smoothed with Giuliani's method. (a) The (failed) patch. (b) The influence domain. (c) The new node position.

the original smoothing scheme without stretched elements, Sarrate and Huerta [5] claims that a node cannot depart from its influence domain since it is an unstable position in terms of distortion and squeeze. We cannot find that this holds for an arbitrary element configuration; for instance, the node in Figure 3b will depart from the influence domain. The new node position is seen in Figure 3c.

4. NUMERICAL EXAMPLES

In the first two examples we investigate the proposed variable diffusion smoothing method's ability to withstand folded elements. We start to examine the element configuration where the Giuliani smoother failed, see Figure 3 or 4a. For $p = 0$, i.e., Laplacian smoothing, the node departs from the domain, see Figure 4c, and for $p = 0.1$ the node stays in the domain, see Figure 4d. As can be seen i Figure 4 the elements become more equal-sized as p increases.

In the second example we examine the method's ability to withstand folded/bad elements in problems where the domain undergoes large deformations, e.g., fluid–structure interaction, and where the nodes are moved using a smoothing algorithm instead of remeshing. Here we choose to displace the top boundary on a unit square, see Figure 5. The Giuliani smoother produces folded elements close to the top boundary, see Figure 5b, which also the variable diffusion smoother does for $p \lesssim 0.8$. A good result is achieved for $p = 0.9$, and for $p = 1$ the solution does not converge due to extreme stretching of the elements.

The third example treats the methods' abilities to smooth a mesh including a curved boundary with local element refinements, see Figure 6a. The Giuliani smoother does not preserve the element sizes close to the curved boundary, which was also reported in [5]. The variable diffusion smoother preserves the element sizes for small p, let say $p < 0.1$. Again, the solution did not converge for $p = 1$.

In the fourth example, we produce a stretched layer on the unit square for $\frac{1}{3} < x < \frac{2}{3}$ with the predefined stretches $\Lambda_1 = 0.5$ and $\Lambda_2 = 1.0$, see Figure 7. We note that all methods are able to produce the desired stretch.

Finally, we show the effect of including the $\boldsymbol{b}$–term in the smoothing scheme. We have made a simplistic implementation on bilinear quadrilaterals, where we have directly computed $\boldsymbol{b}$ elementwise (it will not be zero because of the $x\,y$–term in the approximation).

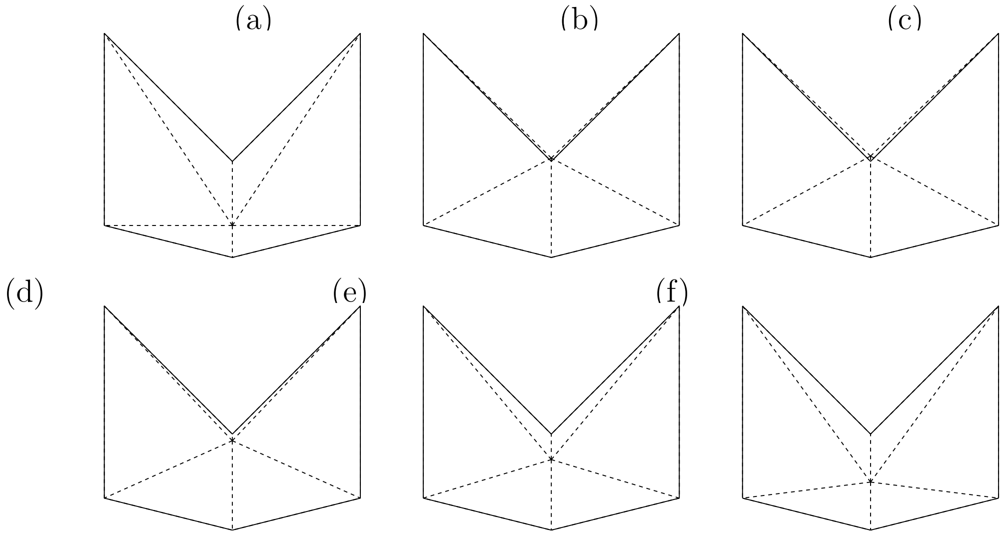

FIGURE 4. A comparison between the proposed variable diffusion method, with $\Lambda_1 = \Lambda_2 = 1$, for different p and the Giuliani method, on the mesh configuration which failed for the latter method, see Figure 2. (a) The original configuration. (b) The mesh smoothed with the Giuliani smoother, and (c) $p = 0$ (Laplacian), (d) $p = 0.1$, (e) $p = 0.5$ and (f) $p = 1$ (modified Winslow).

In Figure 8 we show the resulting meshes for 20×20 elements. Note that the modified Winslow scheme (without $\boldsymbol{b}$) tends to produce more equal-sized elements.

5. CONCLUDING REMARKS

A mesh smoothing approach has been proposed, which is based on a combination of Laplacian smoothing, see [3], and a modified Winslow's method, see [1]. A parameter $0 \leq p \leq 1$, where $p = 0$ is Laplacian smoothing and $p = 1$ is a modified Winslow's method, has to be chosen by the user. By this approach the features of the Laplacian smoother and the modified Winslow's method can be utilized in a better way by a good choice of the parameter p, where the choice of p depends on the problem. Further, it is well known that the original Winslow formulation always result in an unfolded transformation on the continuum level (under certain restrictions on the boundary map), see Knupp and Steinberg [6]. This is not necessarily the case with our modified version. As noted earlier, the problem of folding is however considerably less severe for mesh smoothing than for mesh generation.

Based on our limited set of numerical examples we suggest that a mesh with a concave boundary or a mesh where the elements are wished to be equal-sized needs a larger p, and a mesh where local refinements are present needs a small p. During the simulations it turned out that for $p = 1$ the solution did not converge for our choice of examples, except for the failed patch example. In most cases there is no need to choose a large p to avoid folded elements at concave meshes, a small p is sufficient.

(c)

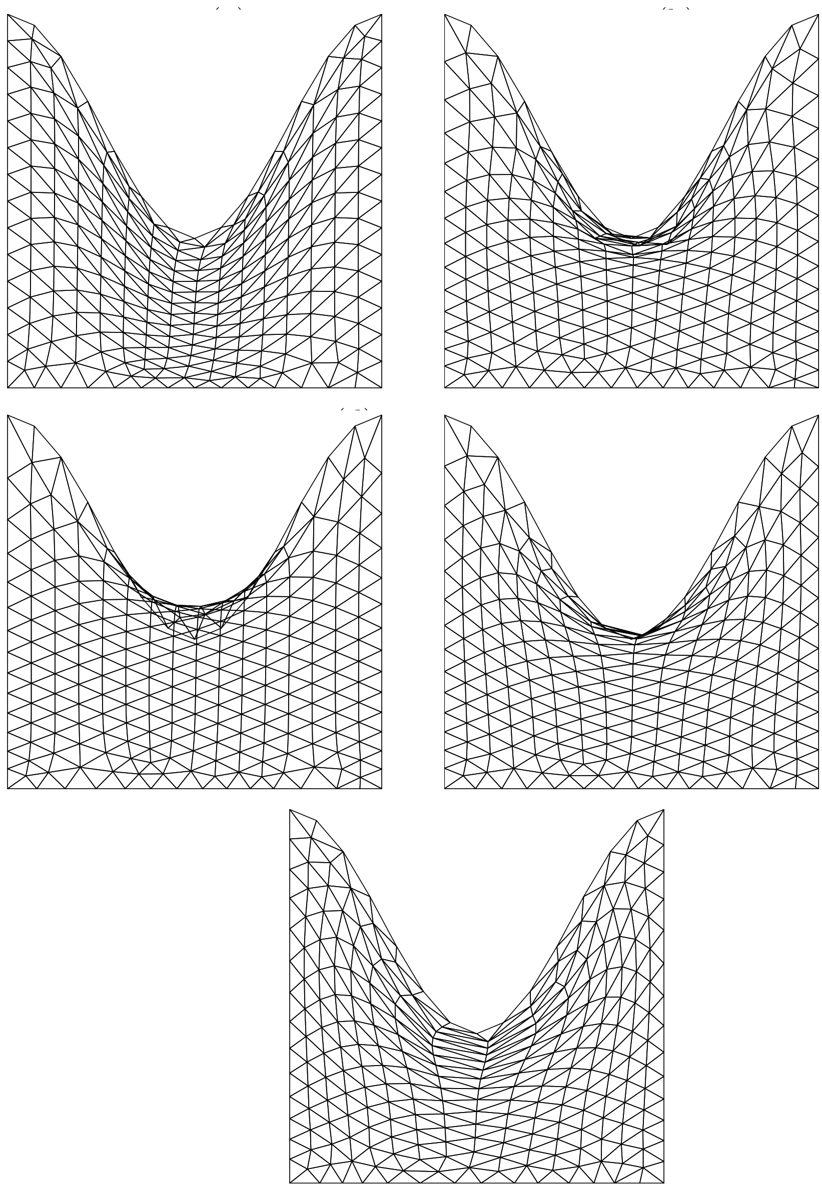

FIGURE 5. (a) The original mesh. The mesh smoothed with $\Lambda_1 = \Lambda_2 = 1$ using (b) Giuliani's method, the proposed variable diffusion method with p chosen as (c) $p = 0.0$ (Laplacian), (d) $p = 0.5$, and (e) $p = 0.9$.

Further, the idea in [3] of how to produce, or maintain, stretched elements during smoothing processes has been incorporated in Giuliani's method, see [2], and in the proposed variable diffusion smoother. During the numerical simulation it turned out that Giuliani's method cannot guarantee that the elements not become folded for all element configurations.

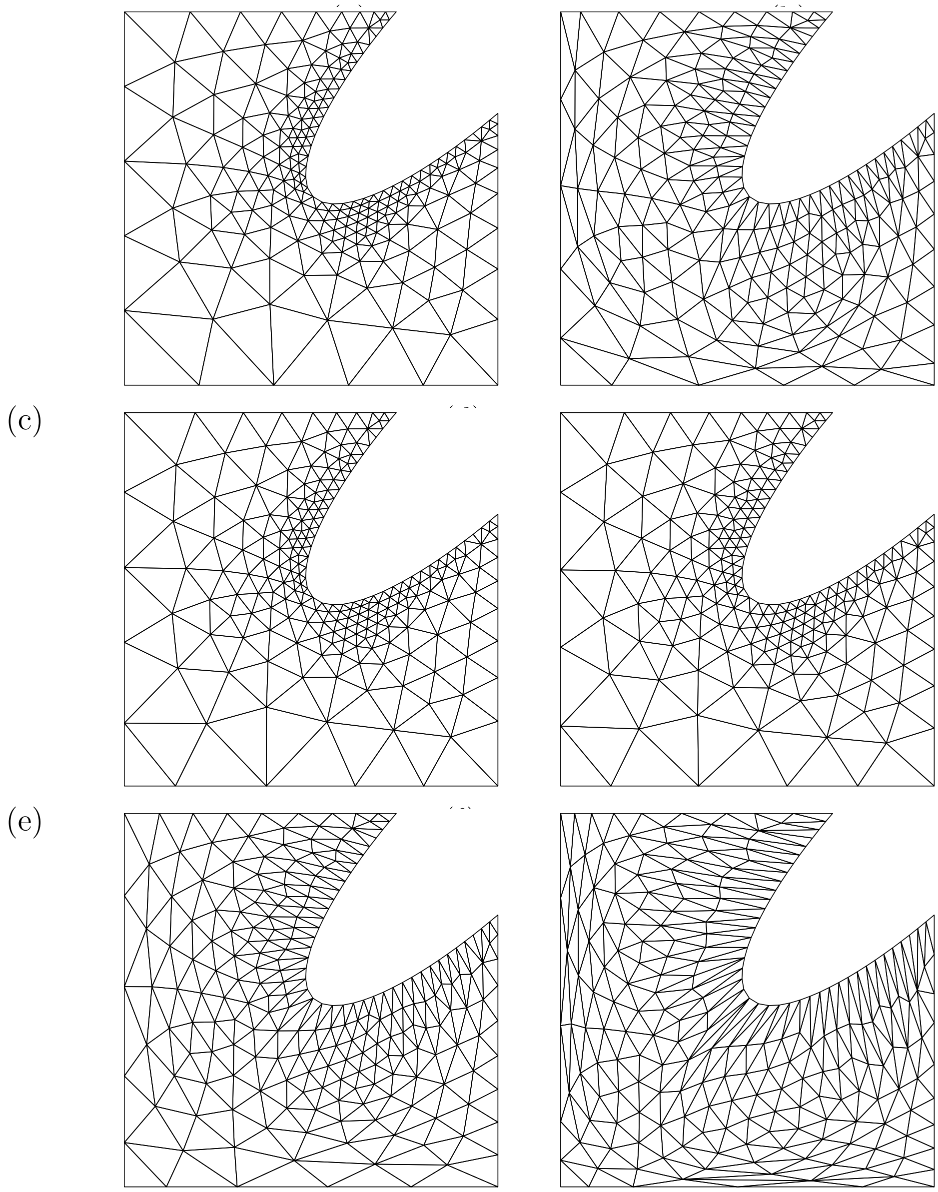

FIGURE 6. A mesh with local element refinements at a curved boundary. The stretches are chosen to be $\Lambda_1 = \Lambda_2 = 1$. (a) The original mesh. (b) The mesh smoothed with the Giuliani smoother, (c) $p = 0.0$ (Laplacian), (d) $p = 0.1$, (e) $p = 0.5$ and (f) $p = 0.9$.

Acknowledgement. This work was supported by the Swedish Foundation for Strategic Research through the Integral Vehicle Structure (IVS) graduate school.

(c)

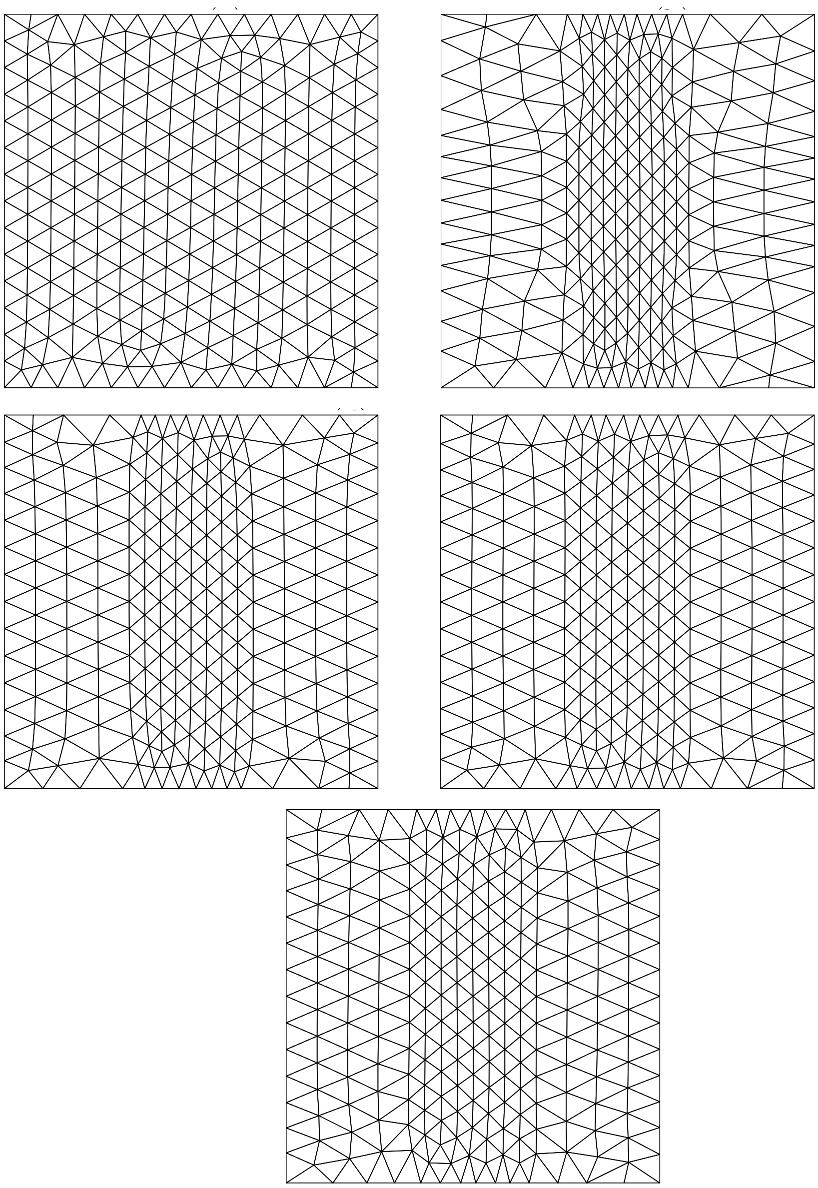

FIGURE 7. A mesh with a stretched layer in the interval $\frac{1}{3} < x < \frac{2}{3}$. The stretches were chosen as $\Lambda_1 = 0.5$ and $\Lambda_2 = 1.0$. (a) The original mesh. The stretched layer produced with (b) Giuliani's method, (c) $p = 0.0$ (Laplacian), (d) $p = 0.5$, and (e) $p = 0.9$.

References

[1] A. Winslow. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *Journal of Computational Physics* 1967; **2**:149–172.
[2] S. Giuliani. An algorithm for continuous rezoning of thehydrodynamic grid in arbitrary Lagrangian-Eulerian computer codes. *Nuclear Engineering and Design* 1982; **72**:205–212.

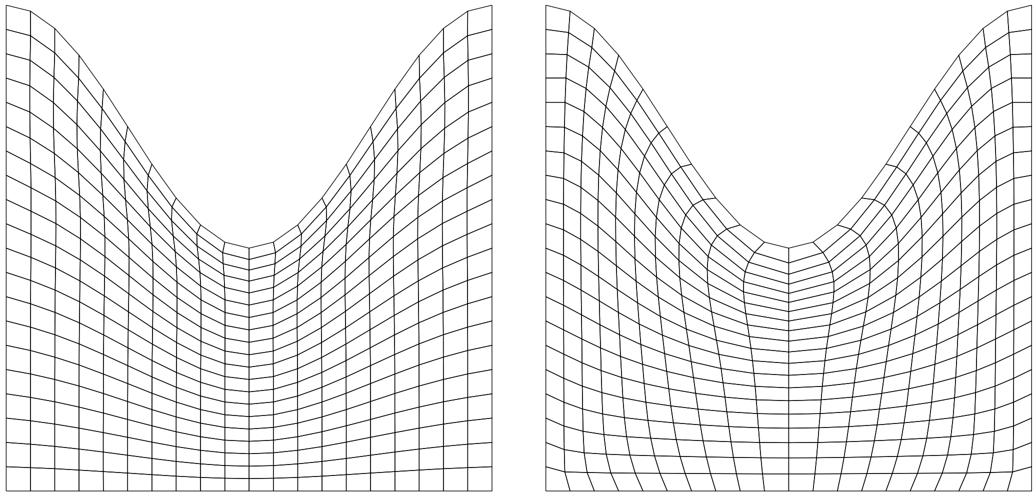

FIGURE 8. Winslow-type smoothing with approximate $\boldsymbol{b}$ (left) and without $\boldsymbol{b}$ (right).

[3] P. Hansbo. Generalized Laplacian smoothing of unstructured grids. *Communications in Numerical Methods in Engineering* 1995;**11**:455–464.

[4] P. M. Knupp. Winslow Smoothing on Two-Dimensional Unstructured Meshes. *Engineering with Computers* 1999; **15**:263–268.

[5] J. Sarrate, A. Huerta. An improved algorithm to smooth graded quadrilateral meshes preserving the prescribed element size. *Communications in Numerical Methods in Engineering* 2001; **17**:89–99.

[6] P. Knupp, S. Steinberg. *Fundamentals of Grid Generation.* CRC Press, Boca Raton, 1993.

[7] M. Shashkov, P. Knupp. Optimization-based reference matrix rezone strategies for arbitrary Lagrangian-Eulerian methods on unstructured Meshes, pp. 167–176, 10th International Meshing Roundtable, Newport Beach CA, 2001.

[8] D. A. Anderson. Grid cell volume control with an adaptive grid generator. *Applied Mathematics and Computation* 1990; **35**:209–217.

Chalmers Finite Element Center Preprints

2001–01 *A simple nonconforming bilinear element for the elasticity problem*
Peter Hansbo and Mats G. Larson

2001–02 *The $\mathcal{LL}^*$ finite element method and multigrid for the magnetostatic problem*
Rickard Bergström, Mats G. Larson, and Klas Samuelsson

2001–03 *The Fokker-Planck operator as an asymptotic limit in anisotropic media*
Mohammad Asadzadeh

2001–04 *A posteriori error estimation of functionals in elliptic problems: experiments*
Mats G. Larson and A. Jonas Niklasson

2001–05 *A note on energy conservation for Hamiltonian systems using continuous time finite elements*
Peter Hansbo

2001–06 *Stationary level set method for modelling sharp interfaces in groundwater flow*
Nahidh Sharif and Nils-Erik Wiberg

2001–07 *Integration methods for the calculation of the magnetostatic field due to coils*
Marzia Fontana

2001–08 *Adaptive finite element computation of 3D magnetostatic problems in potential formulation*
Marzia Fontana

2001–09 *Multi-adaptive galerkin methods for ODEs I: theory & algorithms*
Anders Logg

2001–10 *Multi-adaptive galerkin methods for ODEs II: applications*
Anders Logg

2001–11 *Energy norm a posteriori error estimation for discontinuous Galerkin methods*
Roland Becker, Peter Hansbo, and Mats G. Larson

2001–12 *Analysis of a family of discontinuous Galerkin methods for elliptic problems: the one dimensional case*
Mats G. Larson and A. Jonas Niklasson

2001–13 *Analysis of a nonsymmetric discontinuous Galerkin method for elliptic problems: stability and energy error estimates*
Mats G. Larson and A. Jonas Niklasson

2001–14 *A hybrid method for the wave equation*
Larisa Beilina, Klas Samuelsson and Krister Åhlander

2001–15 *A finite element method for domain decomposition with non-matching grids*
Roland Becker, Peter Hansbo and Rolf Stenberg

2001–16 *Application of stable FEM-FDTD hybrid to scattering problems*
Thomas Rylander and Anders Bondeson

2001–17 *Eddy current computations using adaptive grids and edge elements*
Y. Q. Liu, A. Bondeson, R. Bergström, C. Johnson, M. G. Larson, and K. Samuelsson

2001–18 *Adaptive finite element methods for incompressible fluid flow*
Johan Hoffman and Claes Johnson

2001–19 *Dynamic subgrid modeling for time dependent convection–diffusion–reaction equations with fractal solutions*
Johan Hoffman

2001–20 *Topics in adaptive computational methods for differential equations*
Claes Johnson, Johan Hoffman and Anders Logg

2001–21 *An unfitted finite element method for elliptic interface problems*
Anita Hansbo and Peter Hansbo

2001–22 *A P^2–continuous, P^1–discontinuous finite element method for the Mindlin-Reissner plate model*
Peter Hansbo and Mats G. Larson

2002–01 *Approximation of time derivatives for parabolic equations in Banach space: constant time steps*
Yubin Yan

2002–02 *Approximation of time derivatives for parabolic equations in Banach space: variable time steps*
Yubin Yan

2002–03 *Stability of explicit-implicit hybrid time-stepping schemes for Maxwell's equations*
Thomas Rylander and Anders Bondeson

2002–04 *A computational study of transition to turbulence in shear flow*
Johan Hoffman and Claes Johnson

2002–05 *Adaptive hybrid FEM/FDM methods for inverse scattering problems*
Larisa Beilina

2002–06 *DOLFIN - Dynamic Object oriented Library for FINite element computation*
Johan Hoffman and Anders Logg

2002–07 *Explicit time-stepping for stiff ODEs*
Kenneth Eriksson, Claes Johnson and Anders Logg

2002–08 *Adaptive finite element methods for turbulent flow*
Johan Hoffman

2002–09 *Adaptive multiscale computational modeling of complex incompressible fluid flow*
Johan Hoffman and Claes Johnson

2002–10 *Least-squares finite element method with applications in electromagnetics*
Rickard Bergström

2002–11 *Discontinuous/continuous least-squares finite element methods for Elliptic Problems*
Rickard Bergström and Mats G. Larson

2002–12 *Discontinuous least-squares finite element methods for the Div-Curl problem*
Rickard Bergström and Mats G. Larson

2002–13 *Object oriented implementation of a general finite element code*
Rickard Bergström

2002-14 *On adaptive strategies and error control in fracture mechanics*
Per Heintz and Klas Samuelsson

2002-15 *A unified stabilized method for Stokes' and Darcy's equations*
Erik Burman and Peter Hansbo

2002-16 *A finite element method on composite grids based on Nitsche's method*
Anita Hansbo, Peter Hansbo and Mats G. Larson

2002-17 *Edge stabilization for Galerkin approximations of convection–diffusion problems*
Erik Burman and Peter Hansbo

2002-18 *Adaptive strategies and error control for computing material forces in fracture mechanics.*
Per Heintz, Fredrik Larsson, Peter Hansbo and Kenneth Runesson

2002-19 *A variable diffusion method for mesh smoothing.*
J. Hermansson and P. Hansbo

These preprints can be obtained from

`www.phi.chalmers.se/preprints`