Paper 51

©Civil-Comp Press, 2008
Proceedings of the Sixth International Conference
on Engineering Computational Technology,
M. Papadrakakis and B.H.V. Topping, (Editors),
Civil-Comp Press, Stirlingshire, Scotland

Alignment of Surface Triangulations for Approximating Interior Curves

J.M. Escobar[1,2], E. Rodríguez[2], R. Montenegro[2] and G. Montero[2]
[1] Department of Signal and Communications
[2] University Institute for Intelligent Systems and Numerical Applications in Engineering
University of Las Palmas de Gran Canaria, Spain

Abstract

In this work we develop a procedure to deform a given surface triangulation to obtain its exact alignment with interior curves. These curves, defined by splines, can represent internal interfaces between different materials, internal boundaries, etc. An important feature of this procedure is the possibility to adapt a reference mesh to curves that change their shape or their position in the course of an evolutive process. The method moves the nodes of the mesh, maintaining its topology, in order to achieve two objectives: the piecewise approximation of the curve by edges of the mesh, and the optimization of the deformed mesh resulting from the previous process. The overall method, which we will designate as projecting/smoothing, is based on a surface mesh smoothing technique, where the quality improvement of the mesh is obtained by an iterative process in which each node of the mesh is moved to a new position that minimizes a certain objective function. The objective function is derived from the algebraic quality measure *mean ratio* extended to the set of triangles connected to the *free node*. The projecting/smoothing method allows us to track an object moving through the reference mesh without the necessity of remeshing.

Keywords: mesh alignment, moving meshes, mesh adaptation, surface mesh smoothing, node movement, R-adaptivity.

1 Introduction

The numerical simulation of physical problems requires the internal boundaries and discontinuities to be properly represented. Usually, the largest errors are introduced in a neighborhood of such discontinuities. These errors are often greatly reduced if the mesh is aligned with the discontinuities. That is why it is desirable to have a procedure capable of deforming a given triangulation to get its alignment with a curve.

Although there are numerous works dealing with r-adaptivity, that is, mesh adaption allowing only changes in the position of the nodes, only a few of them consider the problem of the exact mesh alignment with interior curves. In fact, the only paper that we have found in the bibliography tackling this question in similar terms, but for quadrilateral grids is [2]. The authors consider the problem of aligning a planar grid to of multiple embedded curves defined by basic segments as straight lines or arcs of circle. A different approach to the problem can be found in [3], where the curve is approximated by a polygonal line included in the surface triangulation, but in this case the segments are not edges of the mesh. The paper [4] present a variant of Ruppert's algorithm for producing a 2-D Delaunay triangulation of a domain containing arbitrary curved inputs. Nevertheless, this algorithm does not allow a dynamical adaption of the mesh without remeshing.

The procedure that we describe in this paper align a given surface triangulation with an arbitrary curve. Usually we have not an analytical representation of the curve. Instead, it is approximately known by a sequence of interpolating data points. We have chosen a parametric cubic spline as interpolating curve due it is C^2 continuous and it has others interesting properties that will be used later. Obviously, the grade of approximation of the curve depends on the element sizes, therefore, a good strategy is to combine the projecting/smoothing technique with a local mesh refinement [5]. Our procedure is specially indicated for evolutionary problems in which the boundaries change their shape or position with time. For example the ones related to fluid-structure interactions involving large displacement (see, for example [6]), or crack modeling. The projecting/smoothing technique could be also applied to domain decomposition, definition of material interfaces, free boundary problems, etc.

The organization of the paper is as follows. In section 2 a rough description of the proposed method is presented. In section 3 we propose an objective function, and the corresponding modification, able to untangle and smooth plane triangulations simultaneously. The projecting/smoothing method is analyzed for plane triangulations in section 4. Its extension to curved surfaces is straightforward derived from the surface mesh smoothing technique proposed in [1]. Section 5 is devoted to applications with a particular mention to the alignment of surface triangulations with contours delimiting relevant orographic features. The paper concludes with a brief discussion of the work and its possible extensions.

2 Description of the Projecting/Smoothing Method

Let C be a curve, and suppose that it is embedded in a surface mesh T_Σ. The basic idea consists of relocating the nodes of T_Σ closest to C in positions just sited in the curve. This operation, which we will refer to as node projection onto the curve, goes on until getting an approximate representation (interpolation) of C by linked edges of T_Σ. A node of T_Σ is considered *projectable* if we can displace it from its initial position to any point of C in such a way the local mesh does not get tangled. This projection implies an enforced alteration of the original positions of the nodes and, in general, has a negative effect on the quality of the triangles close to C. To avoid this drawback, the remaining nodes are also displaced to new positions following the smoothing procedure proposed in [1].

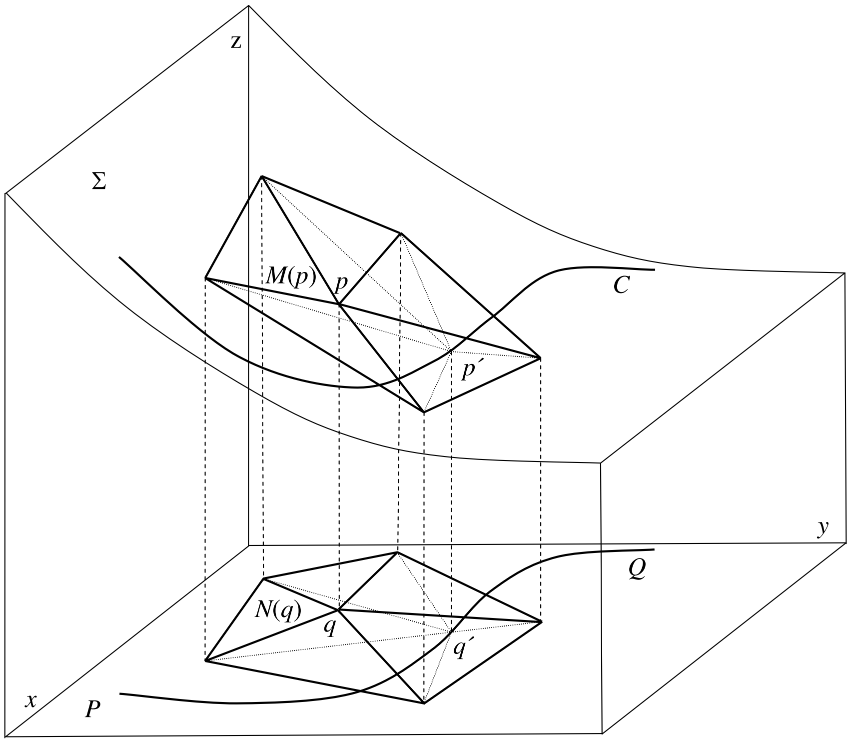


Figure 1: The relocation of node $p \in \Sigma$ is performed in the plane P by projecting q on Q and, consequently, p on C.

For 2-D (or 3-D) meshes the quality improvement can be obtained by an iterative process in which each node of the mesh is moved to a new position that minimizes an objective function derived from certain *algebraic quality measure* of the local mesh [7, 8]. The objective function presents a barrier in the boundary of the *feasible region*. In this context the *feasible region* is the set of points where the free node could be placed to get a *valid* local mesh, that is, without *inverted elements*. This barrier has an important role because it avoids the optimization algorithm to create a tangled mesh when it starts with a valid one, but these objective functions are only directly applicable to plane meshes. We shown in [1] a way to extend the smoothing procedure to curved meshes. The basic idea consist of transforming the original problem on Σ into a two-dimensional one on a plane P. To do this, the local mesh $M(p)$, belonging to T_Σ, is orthogonally projected onto a plane P performing a local mesh $N(q)$, where p is the free node on Σ and q is its orthogonal projection onto P. The plane P is suitably chosen in terms of $M(p)$ in order to get a valid mesh on P (see Fig. 1). Thus, the optimization of $M(p)$ is got by the appropriated optimization of $N(q)$. It involves the construction of *ideal* triangles in $N(q)$ that become near equilateral in $M(p)$.

When Σ is a curved surface each triangle of $M(p)$ is placed on a different plane and it is impossible to define a feasible region. Nevertheless, this region, denoted as $\mathcal{H}_q$, is perfectly defined for $N(q)$ and its associated objective function has a barrier at the boundary of $\mathcal{H}_q$ (see [9]). This is a crucial reason of working on P instead of on Σ.

In the present work the curve C is defined as the image of a curve Q sited on a plane P. Specifically, if we define a plane curve by the parametrization $Q(u) = (x(u), y(u))$ and we consider that $f(x, y)$ is the z coordinate of the underlaying surface (the true surface, if it is known, or the piece-wise linear interpolation, if it is not), then the curve C is given by $C(u) = (x(u), y(u), f(x(u), y(u)))$ (see Fig. 1). This type of parametrization can be straightforward introduced in the new *meccano* method which has been recently developed by the authors [10, 11]. We remark that, although the surface mesh smoothing process can be carried out in different planes chosen in terms of $M(p)$ [1], the particular way in which C is defined demands a unique plane. A general parametric curves $C(u) = (x(u), y(u), z(u))$ could be considered in future works.

The problem of getting a piecewise approximation of C by edges of T_Σ is translated to the plane P. Each node q sited on P is projected onto Q if its corresponding local mesh does not get tangled (see Fig. 1 and 2). Note that, in this work, we consider two kinds of projections: the projection onto a plane and that onto a curve. The task to determine if a node can be projected onto Q and, that being the case, which is its optimal position, is undertaken by an objective function derived from algebraic

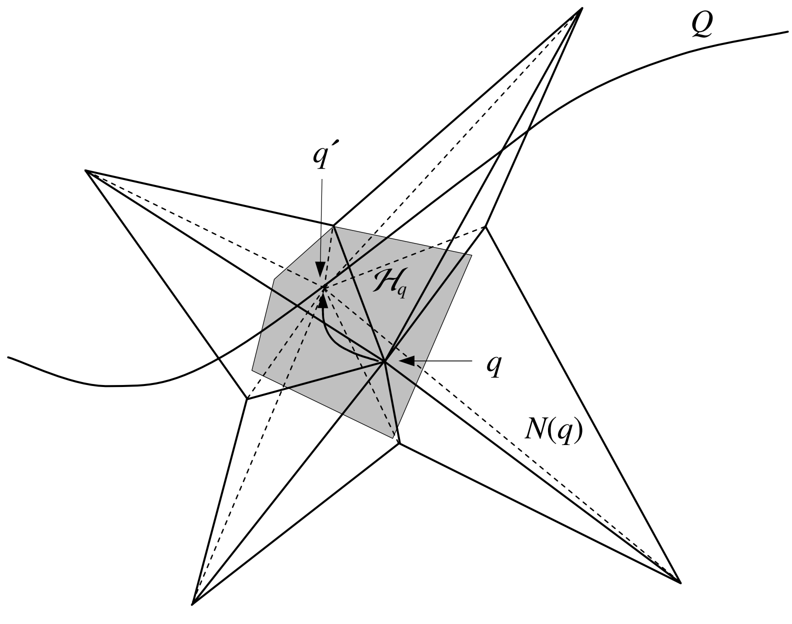


Figure 2: The curve Q intersects the feasible region $\mathcal{H}_q$ (in gray) and, therefore, the node q is projectable, being q' its optimal position on the curve.

quality measures of the local mesh $N(q)$. This objective function incorporates the modifications proposed in [9] in order to deal with tangled meshes.

3 Objective Function for Smoothing and Untangling Plane Triangulations

Firstly, we will focus our attention on finding an objective function to smooth a valid plane triangulation. As it is shown in [8], [12], and [13] we can derive optimization functions from algebraic quality measures of the elements belonging to a local mesh. Let us consider a triangular mesh T_P defined in $\mathbb{R}^2$ and let t be an triangle in the physical space whose vertices are given by $\mathbf{x}_k = (x_k, y_k)^T \in \mathbb{R}^2$, $k = 0, 1, 2$. To start with, we introduce an algebraic quality measure for t. Let t_R be the reference triangle with vertices $\mathbf{u}_0 = (0,0)^T$, $\mathbf{u}_1 = (1,0)^T$, and $\mathbf{u}_2 = (0,1)^T$. If we choose $\mathbf{x}_0$ as the translation vector, the affine map that takes t_R to t is $\mathbf{x} = A\mathbf{u} + \mathbf{x}_0$, where A is the Jacobian matrix of the affine map referenced to node $\mathbf{x}_0$, given by $A = (\mathbf{x}_1 - \mathbf{x}_0, \mathbf{x}_2 - \mathbf{x}_0)$. We will denote this type of affine maps as $t_R \xrightarrow{A} t$. Let now t_I be an *ideal* triangle (equilateral in this case) whose vertices are $\mathbf{w}_k \in \mathbb{R}^2$, $(k = 0, 1, 2)$ and let $W_I = (\mathbf{w}_1 - \mathbf{w}_0, \mathbf{w}_2 - \mathbf{w}_0)$ be the Jacobian matrix, referenced to node $\mathbf{w}_0$, of the affine map $t_R \xrightarrow{W_I} t_I$; then, we define $S = AW_I^{-1}$ as the weighted Jacobian matrix of the affine map $t_I \xrightarrow{S} t$. In the particular case that t_I was the equilateral triangle t_E, the Jacobian matrix $W_I = W_E$ will be defined by $\mathbf{w}_0 = (0,0)^T$, $\mathbf{w}_1 = (1,0)^T$ and $\mathbf{w}_2 = (1/2, \sqrt{3}/2)^T$.

We can use matrix norms, determinant or trace of S to construct algebraic quality measures of t. For example, the Frobenius norm of S, defined by $|S| = \sqrt{tr\left(S^T S\right)}$, is specially indicated because it is easily computable. Thus, it is shown in [7] that $q_\eta = \frac{2\sigma}{|S|^2}$ is an algebraic quality measure of t, where $\sigma = \det(S)$. We use this quality measure to construct an objective function. Let $\mathbf{x} = (x, y)^T$ be the position vector of the free node, and let S_m be the weighted Jacobian matrix of the m-th triangle of a valid local mesh of M triangles. The objective function associated to m-th triangle is $\frac{|S_m|^2}{2\sigma_m}$, and the corresponding objective function for the local mesh is

$$|K_\eta|_n(\mathbf{x}) = \left[\sum_{m=1}^{M} \left(\frac{|S_m|^2}{2\sigma_m}\right)^n (\mathbf{x})\right]^{\frac{1}{n}} \tag{1}$$

being n an integer number, typically $n = 1$ or $n = 2$.

In this context the feasible region is defined as the set of points where the free node must be located to get the local mesh to be valid. More precisely, the feasible region is the interior of the polygonal set $\mathcal{H} = \bigcap_{m=1}^{M} H_m$, where H_m are the half-planes defined by $\sigma_m(\mathbf{x}) \geq 0$. We say that a triangle is *inverted* if $\sigma < 0$. The objective function (1) presents a barrier in the boundary of the feasible region. This barrier avoids the optimization method to create a tangled mesh when it starts with a valid one, but, on the other hand, it prevents the algorithm to untangle it when there are inverted elements. Therefore, this objective function is only appropriate to improve the quality of a valid mesh, not to untangle it. To construct an objective function applicable to deal with tangled meshes we propose to modify it following the criteria developed in [9]. This modification consists of substituting σ in (1) by the positive and increasing function

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

being the parameter $\delta = h(0)$.

In this way, the barrier associated with the singularities of $\left|K_\eta\right|_n(\mathbf{x})$ will be eliminated and the new function will be smooth all over $\mathbb{R}^2$.

The modified objective function is

$$\left|K'_\eta\right|_n(\mathbf{x}) = \left[\sum_{m=1}^{M} \left(\frac{|S_m|^2}{2h(\sigma_m)}\right)^n (\mathbf{x})\right]^{\frac{1}{n}} \tag{3}$$

This new objective function strongly penalizes the negative values of σ, so that, the minimization process of (3) leads to the construction of a local mesh $N(q)$ without inverted triangles, provided it is possible. Note that the minimum of original and modified functions are nearly identical when $\mathcal{H} \neq \emptyset$ and δ tends to zero. With this approach, we can use any standard and efficient unconstrained optimization method to find the minimum of the modified objective function, see for example [14]

4 Alignment with Curves Defined on Plane Triangulations

Node movement provides the mesh the ability to align with an arbitrary curve. Suppose that Q is a curve defined on a 2-D triangulation T_P, our objective is to move some nodes of T_P, projecting them onto Q, to get an interpolation of Q by linked edges of T_P. To achieve this objective we have to decide which nodes of T_P can be projected onto Q without inverting any triangle of its local mesh. More accurately, we say that the free node q is projectable onto Q if there are points of this curve belonging to the feasible region $\mathcal{H}_q$ (see Fig. 2).

In general, if q is projectable, its possible placement on Q is not unique. The projecting/smoothing method must determine if q can be projected onto Q and, if so, which is its optimal position. The last question can be answered by using the objective function (3) subject to the constrained $\mathbf{x} \in Q$. Thus, the problem of finding the optimal position to project the free node on the curve is

$$\text{minimize} \left|K'_\eta\right|_n(\mathbf{x}), \text{ subject to } \mathbf{x} \in Q \tag{4}$$

If $\bar{\mathbf{x}}$ is the minimizing position of (4) and $\sigma(\bar{\mathbf{x}}) > 0$ for all triangle of $N(q)$, we conclude that q is projectable onto Q and $\bar{\mathbf{x}}$ is its optimal position. Otherwise, q is not projectable.

4.1 Curve Definition

The previous criterion allows to determine whether q is projectable onto Q or not, but it involves a high computational cost because it needs to solve the constrained minimization problem (4). Nevertheless, it is clear that most nodes of T_P are not projectable because they are very far from any point of the curve. Therefore, it is convenient to have an efficient method to select those nodes, close to some segment of Q, expected to be projectable.

In many situations of practical interest we have not an analytical representation of Q, but Q is approximately known by a sequence of interpolating data points. Among the options to define an interpolating curve, we have chosen a parametric cubic spline as it has many desired properties: it is a C^2 continuous function, it has a very simple local form, it is minimally oscillating, etc. Moreover, each segment of the spline is a degree 3 Bézier curve that lies within the convex hull of its four defining control points (see, for example [15]). We will use this property in order to know if a given node is close to some segment of Q.

Let $\{P_0, P_1, \ldots, P_m\}$ be a set of interpolating points belonging to $\mathbb{R}^2$ and consider that $\{u_0, u_1, \ldots, u_m\}$ is the corresponding knot vector. The parametric cubic spline

$$Q(u) = (x(u), y(u)), \text{ where } u \in [u_0, u_m] \tag{5}$$

is an interpolating curve that satisfies $Q(u_i) = P_i$ for $i = 0, \ldots, m$ and two additional constraints in order to be fully defined. Usually, these constraints are imposed at the ends of the curve. For example, the conditions $Q''(u_0) = 0$ and $Q''(u_m) = 0$ define a spline known as *natural*.

Every segment of the spline delimited by two consecutive interpolating points is a degree 3 polynomial. Suppose that $Q_i(t) = \mathbf{a}^i + \mathbf{b}^i t + \mathbf{c}^i t^2 + \mathbf{d}^i t^3$, with $\mathbf{a}^i$, $\mathbf{b}^i$, $\mathbf{c}^i$ and $\mathbf{d}^i$ in $\mathbb{R}^2$, is the polynomial of the segment Q_i $(i = 0, 1, \ldots, m-1)$ that runs from P_i to P_{i+1}, being $t \in [0, 1]$ the local parameter. This one is related with the parameter of the entire curve by $t = (u - u_i) / (u_{i+1} - u_i)$

4.2 Procedure to Project Nodes onto the Curve

The Q_i segment also is a degree 3 Bézier curve, given by $Q_i(t) = \sum_{j=0}^{3} \mathbf{u}_j^i B_j^i(t)$ with $t \in [0, 1]$, where $B_j^i(t)$ are the Berstein polynomials and $\mathbf{u}_j^i \in \mathbb{R}^2$ are the control points. The relation between the polynomial coefficients and the control points are given by

$$\begin{pmatrix} \mathbf{u}_0^i \\ \mathbf{u}_1^i \\ \mathbf{u}_2^i \\ \mathbf{u}_3^i \end{pmatrix} = \frac{1}{3} \begin{pmatrix} 3 & 0 & 0 & 0 \\ 3 & 1 & 0 & 0 \\ 3 & 2 & 1 & 0 \\ 3 & 3 & 3 & 3 \end{pmatrix} \begin{pmatrix} \mathbf{a}^i \\ \mathbf{b}^i \\ \mathbf{c}^i \\ \mathbf{d}^i \end{pmatrix} \tag{6}$$

As we said, an interesting property of the Bézier curves establishes that the Q_i segment lies within the convex hull of its control points. If CH denotes the convex hull of a set of points, we have $Q_i \subseteq \text{CH}\,(\mathbf{u}_0^i, \ldots, \mathbf{u}_3^i)$. Note that a necessary (but not sufficient) condition for the node q to be projectable onto Q is that its feasible region $\mathcal{H}_q$ intersects the convex hull of some segment of the curve. In other words, it must exist a segment Q_i such that $\mathcal{H}_q \cap \text{CH}\,(\mathbf{u}_0^i, \ldots, \mathbf{u}_3^i) \neq \emptyset$. This property allows us to know beforehand which nodes are not projectable, because they yield an empty intersection for all segments of the curve. Nevertheless, calculating the set $\mathcal{H}_q$ and, moreover, its intersection with a convex set, is not a trivial problem, so it is more advisable to deal with a simplified version.

Let R_q and R_{Q_i} be the minimal rectangles, with sides parallel to the axes, enclosing the sets $N(q)$ and $\text{CH}\,(\mathbf{u}_0^i, \ldots, \mathbf{u}_3^i)$, respectively. Then, due to $\mathcal{H}_q \subset R_q$, it is clear that q is projectable onto Q_i only if $R_q \cap R_{Q_i} \neq \emptyset$ (see Fig. 3). The computation of this intersection allows us to take a quick decision about if a node is candidate to be projected onto the curve.

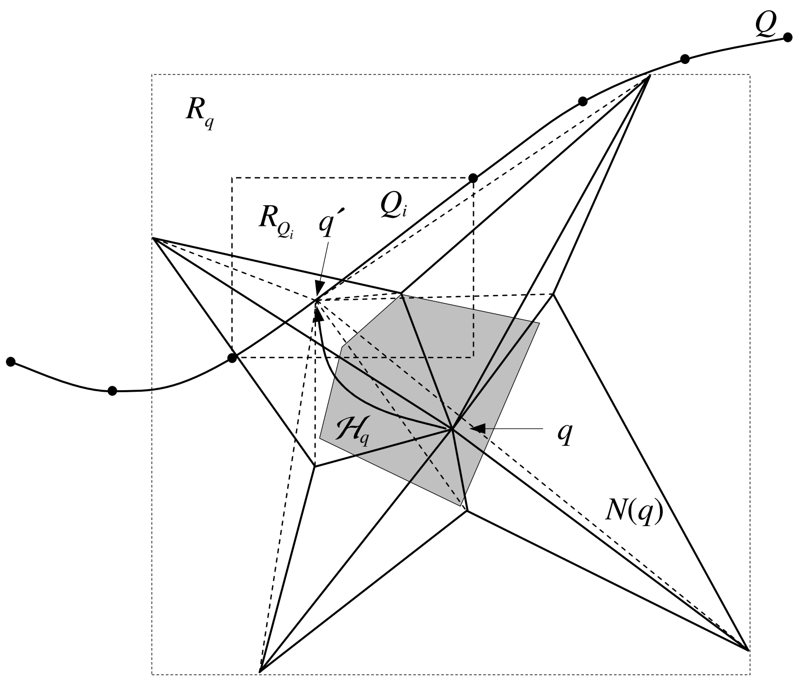


Figure 3: The figure shows the situation in which $R_q \cap R_{Q_i} \neq \emptyset$, but node q is not projectable because the optimal position for the free node, q', is outside the feasible region.

The algorithm to determine if q is projectable onto Q and, if it is so, which is its optimal position, can be summarized as follows. For each segment of the curve analyze $R_q \cap R_{Q_i}$ and, if this set is not empty, solve the minimization problem

$$\text{minimize} \left|K'_\eta\right|_n \left(Q_i\left(t\right)\right), \text{ for } t \in [0,1] \tag{7}$$

Let $\bar{t}$ be the global minimum of (7) and $\bar{\mathbf{x}}_i = Q_i\left(\bar{t}\right)$ the corresponding position of the free node q on the segment Q_i. We say that $\bar{\mathbf{x}}_i$ is an admissible position for the free node if $\sigma_m\left(\bar{\mathbf{x}}_i\right) > 0$ for $m = 1, \ldots, M$. In order to determine the optimal position of the free node, we take $\bar{\mathbf{x}}_{opt}$ as the best admissible position for all segments.

The projection of a free node on Q can give rise to a local mesh with very poor quality. Although this effect is partly palliated after smoothing the remainder nodes, following the procedure described in section 3, it is appropriate to tighten the condition $\sigma_m\left(\bar{\mathbf{x}}_i\right) > 0$ enforcing $\sigma_m\left(\bar{\mathbf{x}}_i\right) > \varepsilon$, with $\varepsilon > 0$ a prescribed tolerance. Nevertheless, this more restrictive condition makes it difficult for the nodes to be projected onto the curve and it could produce situations in which some sections of the curve are not interpolated by edges of T_P. This drawback will be studied in the next subsection but, for that purpose, it needs further clarification.

Up to now, we have accepted that parameter t pertains to the closed interval $[0, 1]$ and, in consequence, the problem (7) admits a global minimum. But, with this consideration, the ends of the consecutive segments are shared and, therefore, a projected point can belong to two segments at the same time. In order to avoid this ambiguity, we will assume that each segment $Q_i\left(t\right)$ is defined for $t \in [0, 1)$, except the last one, that it is for $t \in [0, 1]$ if the curve is open. In this way, each point of the curve belongs to a unique segment.

4.3 Detection and Reconstruction of Discontinuities of the Interpolated Curve

It can happen that, after repositioning all the nodes of the mesh, the piecewise approximation of Q by edges of T_P is not continuous. We can detect this discontinuity if we take into account that the projected nodes are arranged in the curve. Thus, a section of the interpolated curve among two consecutive projected nodes is discontinuous if they are not connected by an edge of T_P.

As the parameter $t \in [0, 1)$ induces an order relation in each segment of the curve and, in turn, each segment is ordered by its subindex, we can say that the node $p \in Q_i$ precedes $p' \in Q_j$ if $i < j$ or, in case of $i = j$, if the corresponding parameters satisfies $t_p < t_{p'}$. A possibility to correct a detected discontinuity in the piecewise approximation of Q is to relax the condition $\sigma_m\left(\bar{\mathbf{x}}_i\right) > \varepsilon$, by decreasing the value of ε. However, there are situations in which, even taken ε equal to zero, there are discontinuities impossible to avoid without removing some of the projected nodes. The Fig. 4(a) shows a scheme of this problem. It can be seen that it is impossible to project the node q (neither r) without tangling the mesh. We propose a solution to this conflict by enforcing the free node q to be projected, even if a tangled mesh is created. The Fig. 4(b) shows how the movement of q produces the tangled triangle abq.

Afterward, the position of q is fixed for subsequent iterations of the projecting/smoothing algorithm, but the surrounding nodes are free to move in search of their optimal positions that untangle the mesh and complete the interpolation of the curve (see 4(c)). The algorithm extracts nodes from the curve if their current positions are not admissible (see the new position of node a in figure 4(c)).

Sometimes the curve represented by splines has sharp features that we want to preserve in the piecewise interpolation. To reach this objective we select, from the interpolating points, a set of prescribed points sited in strategic locations. Once the projecting/smoothing process has finished, the algorithm searches among the nodes projected on Q, which one is the optimal candidate, say q, to be relocated in the position of each prescribed node. If $\mathbf{x}_{pres}$ is the position of certain prescribed point, the node q is chosen, among the nodes projected on Q and close to $\mathbf{x}_{pres}$, as the one that maximizes the quality of $N(q)$ when q is enforced to take the position $\mathbf{x}_{pres}$. Obviously, if $N(q)$ is not valid after the relocation of q, a new iteration of the projecting/smoothing procedure must be done.

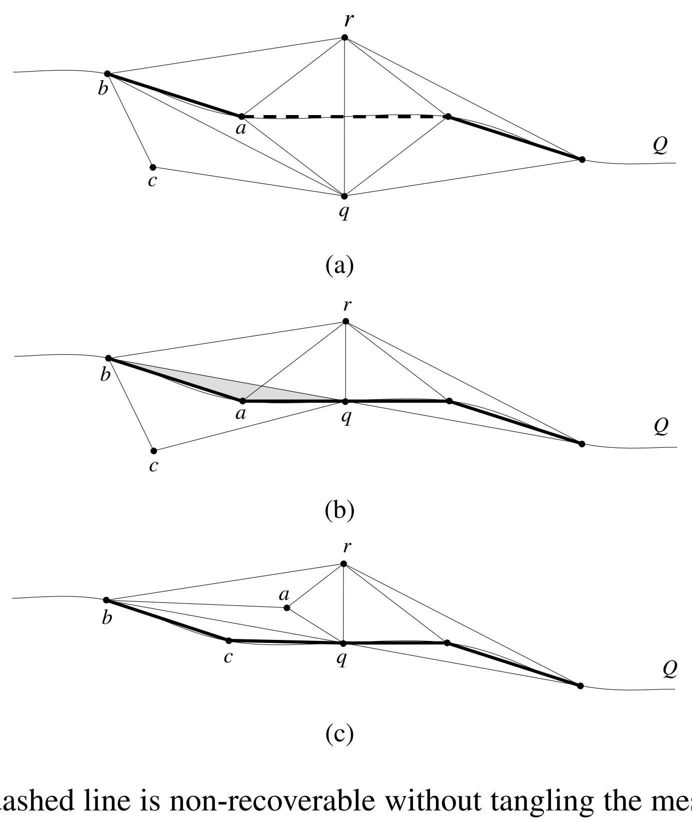


Figure 4: The dashed line is non-recoverable without tangling the mesh (a). The free node q is enforced to be projected (b). The tangled triangle abq is untangled and the node c is also projected (c).

4.4 Extension to Curved Surfaces

We only will point out here that the projecting/smoothing method can be extended to curved surfaces following the surface mesh smoothing technique developed in [1]. As we pointed in section 2, the original problem on Σ is transformed in another one on the plane P. The more significant difference with respect to the former method consists of searching *ideal* triangles in $N(q)$ that become equilateral in $M(p)$.

5 Applications

In this section we present two applications that demonstrate the satisfactory behavior of the projecting/smoothing technique. All the quality measures have been calculated by using the algebraic quality metric based on the condition number proposed in [8].

The first example is a NACA012 profile inserted in an uniform plane mesh with 8364 triangles. The profile, defined by a spline with 36 interpolating points, has been represented with two different angles of attack. We have chosen two prescribed points sited in the leading and trailing edges in order to get a better approximation to the real shape of the profile. The nodes of the triangulation projected on the spline are drawn in bold in order to show up the figure. The number of iterations of the projecting/smoothing algorithm has been 4 for both angles of attack. Note that these 4 iterations was enough to outline the complete contour of the profile in all the cases. All the triangles of the initial mesh are identical, so all of them have the same quality, 0.866. The minimum quality after the projecting/smoothing process was 0.357 and 0.513 for the meshes of the Figs. 5(a) and 5(b), respectively. The average quality was nearly identical for both meshes, 0.875.

The second example corresponds to a surface mesh of a bull, obtained from www-c.inria.fr/gamma/, in which we have inserted the emblem of the Miura's bull breeders, see Fig. 6. In Fig. 7(a) it is shown a detail of the initial mesh and the interpolating points (in bold) used to define the 26 splines composing the emblem. Fig. 7(b) shows the same detail after 5 iterations of the projecting/smoothing process. In this case the points in bold corresponds to the nodes of the mesh projected on the curve. We have used 26 prescribed points in the extremes of the splines in order to keep the sharp angles of the original drawing.

6 Concluding Remarks and Future Research

In this paper we have introduced the projecting/smoothing technique which is able to align a surface triangulation with arbitrary curves without producing, in general, a significant decrease in the minimum quality of the mesh. Indeed, the average quality is increased in many cases as the remainder part of the mesh undergoes a smoothing process.

The technique presented here allows the mesh to align with interior curves which can represent objects moving in a fixed mesh. An important feature of this procedure is the possibility to adapt a reference mesh to curves that change their shape or position in the course of an evolutionary process.

In present work the curves have been defined by splines which interpolating points are given on a plane. In future research, general parametric curves embedded on the surface will be considered. Another more ambitious generalization lies in extending the present method to align a tetrahedral mesh with interior surfaces.

Acknowledgments

This work has been supported by the *Secretaría de Estado de Universidades e Investigación* of the *Ministerio de Educación y Ciencia* of the Spanish Government and FEDER, grant contract: CGL2004-06171-C03-02/CLI.

References

[1] J.M. Escobar, G. Montero, R. Montenegro and E. Rodríguez, An algebraic method for smoothing surface triangulations on a local parametric space, Int. J. Numer. Methods. Engng. 66 (2006) 740-760.

[2] J.M. Hyman, S. Li, P.M. Knupp and M. Shashkov, An algorithm to align a quadrilateral grid with internal boundaries, Journal of Computational Physics 163 (2000) 133-149.

[3] G.P. Bonneau and S. Hahmann, Smooth polylines on polygon meshes, in G. Brunnett, B. Hamann, H. Mueller (eds.): Geometric Modeling for Scientific Visualization, Springer, pp. 69-84, (2003).

[4] S.E. Pav and N.J. Walkington, Delaunay refinement by corner looping, in: Proc. 14th International Meshing Roundtable, Springer, Berlin, 2005, pp. 165-181.

[5] J.M. González-Yuste, R. Montenegro, J.M. Escobar, G. Montero, and E. Rodríguez, Local refinement of 3-D triangulations using object-oriented methods, Advances in Engineering Software 35 (2004) 693702.

[6] K. Stein, T.E. Tezduyar and R. Benney, Automatic mesh update with the solid-extension mesh moving technique, Comput. Methods Appl. Mech. Engng. 193 (2004) 2019-2032.

[7] P.M. Knupp, Algebraic mesh quality metrics, SIAM J. Sci. Comput. 23 (2001) 193-218.

[8] L.A. Freitag and P.M. Knupp, Tetrahedral mesh improvement via optimization of the element condition number, Int. J. Numer. Methods Engng. 53 (2002) 1377-1391.

[9] J.M. Escobar, E. Rodríguez, R. Montenegro, G. Montero and J.M. González-Yuste, Simultaneous untangling and smoothing of tetrahedral meshes, Comp. Meth. Appl. Mech. Eng. 192 (2003) 2775-2787.

[10] J.M. Cascón, R. Montenegro, J.M. Escobar, E. Rodríguez and G. Montero, A new *Meccano* technique for adaptive 3-D triangulations, in: Proc. 16th International Meshing Roundtable, Springer, Berlin, 2007, pp. 103-120.

[11] R. Montenegro, J.M. Cascón, J.M. Escobar, E. Rodríguez and G. Montero, An automatic strategy for adaptive tetrahedral mesh generation, App. Num. Math. (In press) (2008).

[12] P.M. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part I-A framework for surface mesh optimization, Int. J. Num. Meth. Eng. 48 (2000) 401-420.

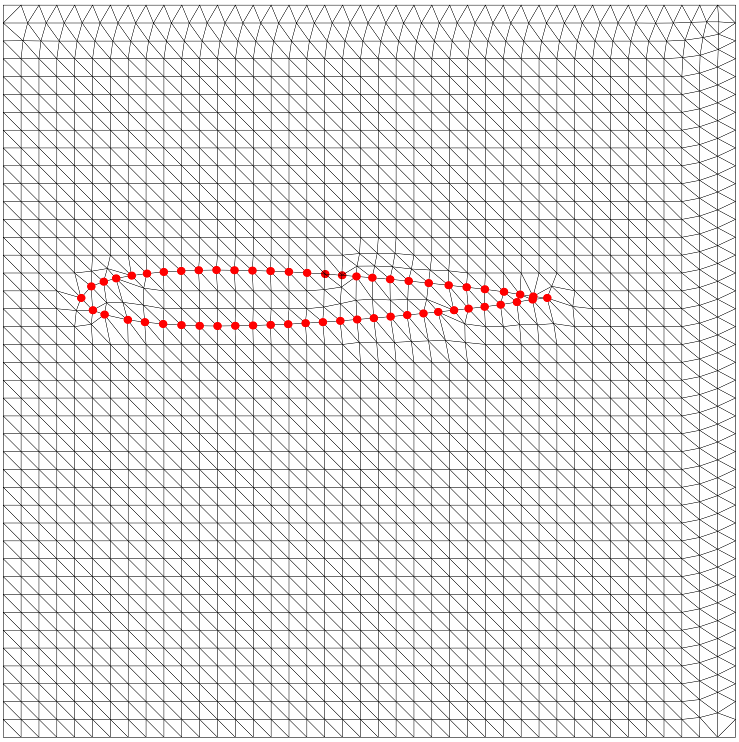
[13] P.M. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. Part II-A frame work for volume mesh optimization and the condition number of the Jacobian matrix, Int. J. Numer. Methods Engng. 48 (2000) 1165-1185.

[14] M.S. Bazaraa, H.D. Sherali and C.M. Shetty, Nonlinear programing: Theory and algorithms (John Wiley and Sons, Inc., New York, 1993).

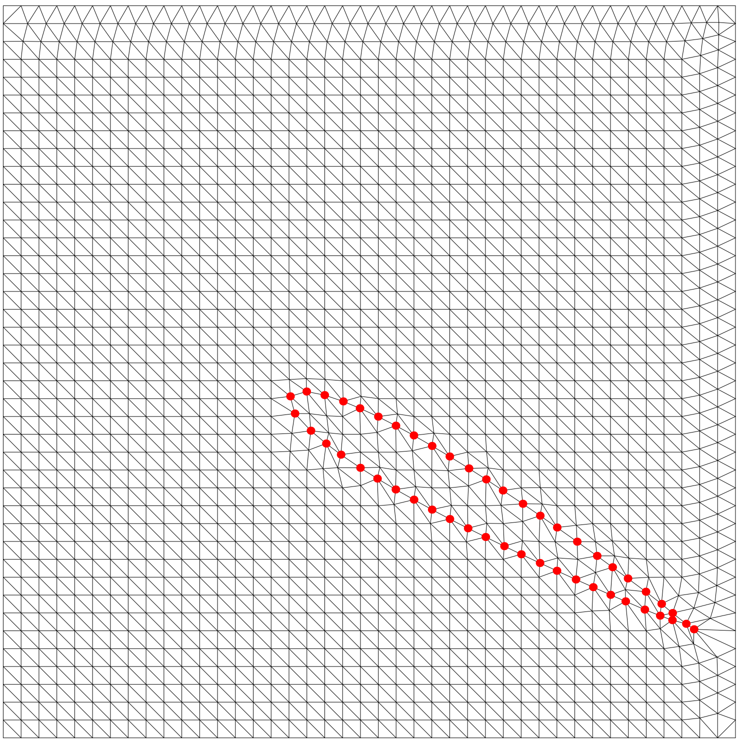
[15] H.R. Bartels, J.C. Beatty and B.A. Barsky, An introduction to Splines for use in computer graphics & geometric modeling, (Morgan Kaufmann, Los Altos, CA, 1987).

[16] R. Montenegro, G. Montero, J.M. Escobar, E. Rodríguez and J.M. González-Yuste, Tetrahedral mesh generation for environmental problems over complex terrain, Lect. Notes Comp. Sci. 232 (2002) 335-344.

[17] G. Montero, R. Montenegro and J.M. Escobar, A 3-D Diagnostic model for wind field adjustment, J. Wind Engng. Ind. Aer. 74-76 (1998) 249-261.



(a)



(b)

Figure 5: NACA012 profile inserted in an uniform mesh with $0°$ angle of attack (a) and $30°$ (b).

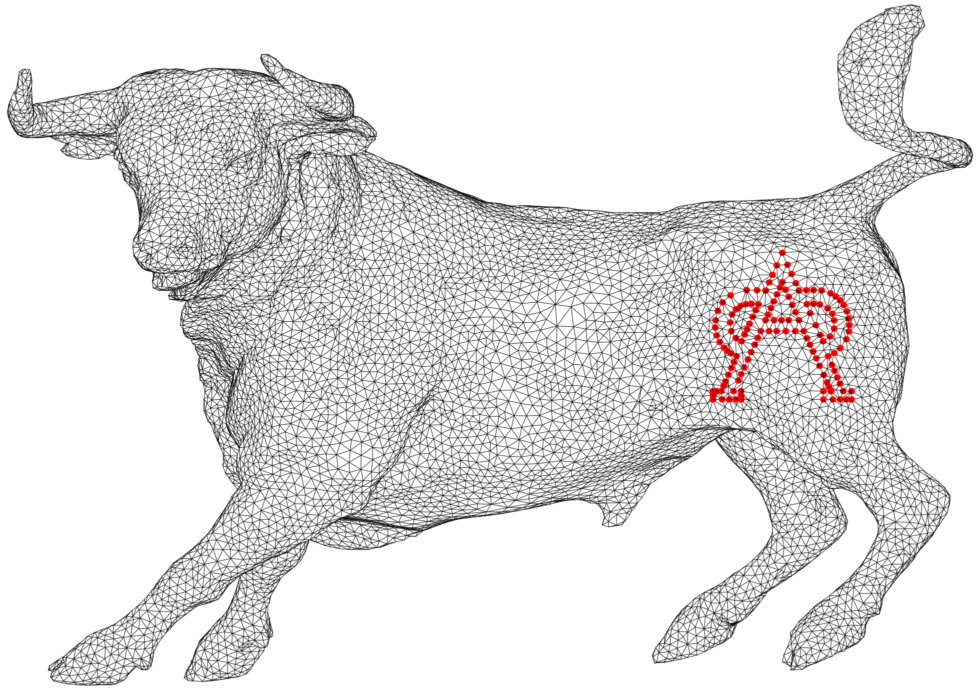
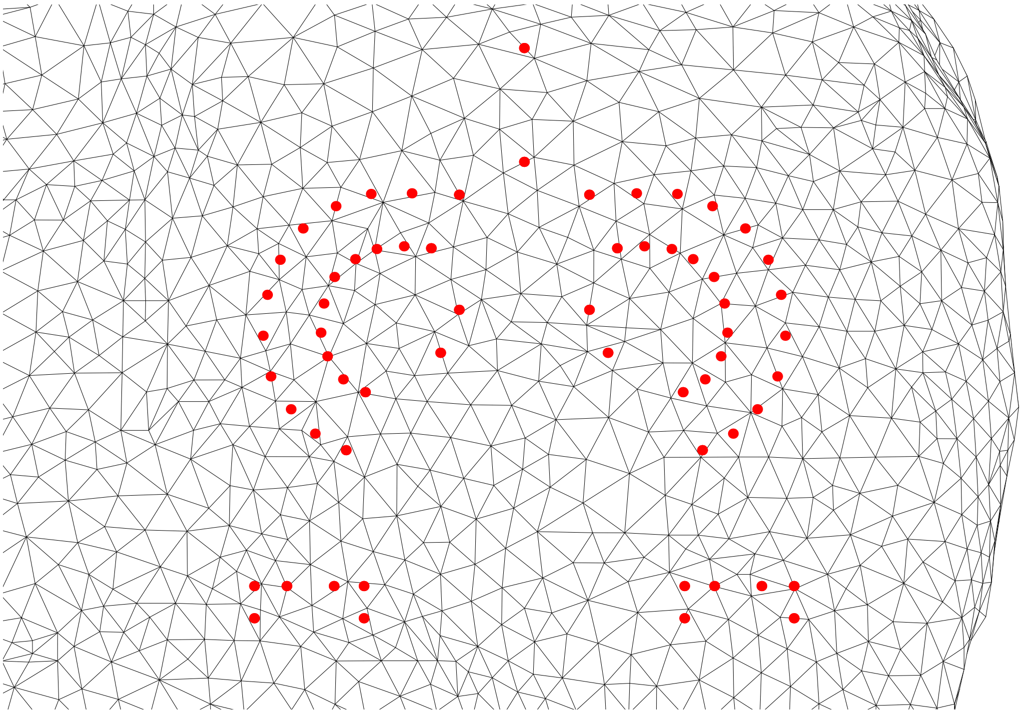
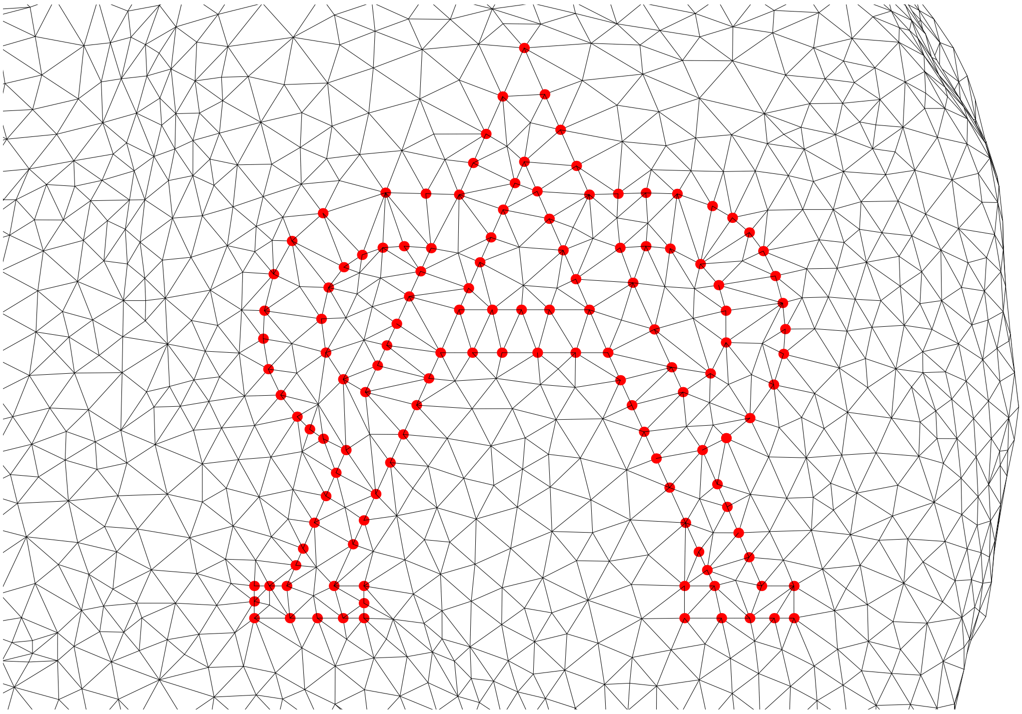


Figure 6: Mesh of a bull with an emblem inserted in its back.



(a)



(b)

Figure 7: Detail of the initial mesh including the interpolating points that define the splines (a). The same detail, remarking the nodes of the mesh projected on the curve, after 5 iterations of the projecting/smoothing process (b).