INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, VOL. 41, 915–934 (1998)

ADAPTIVE TRIANGULAR–QUADRILATERAL MESH GENERATION

HOUMAN BOROUCHAKI [1*] AND PASCAL J. FREY [2]

[1] *Université de Technologie de Troyes, BP 2060, 10010 Troyes Cedex, France*
[2] *INRIA, Gamma Project, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France*

ABSTRACT

In this paper, we begin by recalling an adaptive mesh generation method governed by isotropic and anisotropic discrete metric maps, by means of the generation of a unit mesh with respect to a Riemannian structure. We propose then an automatic triangular to quadrilateral mesh conversion scheme, which generalizes the standard case to the anisotropic context. In addition, we introduce an optimal vertex smoothing procedure. Application test examples, in particular a CFD test, are given to demonstrate the efficiency of the proposed method. © 1998 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Engng., **41**, 915–934 (1998)

KEY WORDS: delaunay triangulation; anisotropic mesh generation; adaptive meshing; quadrilateral mesh; Riemannian structure

1. INTRODUCTION

Mesh adaption offers the advantages of improving the accuracy of the computed solution in the context of the finite element methods and reducing the number of elements required to capture the behaviour of the underlying physical phenomenon.[1–4] This procedure leads to refine or derefine the mesh according to the variation of the gradient of the solution. This information is translated into a size map associated with the initial mesh vertices, which indicates at each vertex the desired size of its surrounding elements. An extension of the latter indicates also the desired sizes not defined in the usual metric but defined in a general Riemannian metric thus making the treatment of anisotropic cases possible. The problem we face is to construct a triangular or quadrilateral mesh which respects a given Riemannian metric map.

Basically, two different approaches can be envisaged to address the adaptive triangular mesh generation problem of a given domain:

(i) mesh optimization using refinement and derefinement tools[5–7] and
(ii) mesh reconstruction of the whole domain.[8]

For both cases, numerous algorithms have been proposed which appear to give satisfactory results, in particular when isotropic specifications are required.

* Correspondence to: H. Borouchaki, Gamma Project, INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France. E-mail: houman.borouchaki@inria.fr.

CCC 0029–5981/98/050915–20$17.50

Received 16 August 1996
Revised 16 April 1997

There exists mainly two approaches (direct and indirect) concerning the generation of quadrilateral meshes for domains of arbitrary shape. We briefly review these two classes of methods.

1.1. Direct approaches

Among the direct approaches, essentially two methods have been proposed, based on

(i) the domain decomposition followed by quadrilateral subdomain filling by means of an algebraic method[9, 10] and
(ii) quadrilateral paving techniques.[11]

The first method is domain decomposition sensitive and relies on the quasi-convex nature of the resulting sub-domains. The domain decomposition algorithms usually require the local or global knowledge of the domain. In particular in this last case, the skeleton fully defines and allows an accurate decomposition of the domain. The skeleton extraction of a domain of $\mathbb{R}^2$ is an almost solved problem. The second method consists in paving the domain from the boundary to the interior and managing the front collisions. By nature, the performances of this method are closely related to the boundary discretization.

In the case where a constant isotropic metric field is specified, these two classes of methods are likely to lead to the same results. Actually, when using the second method, the advancing-front shock approaches the skeleton. On the other hand, if a generalized metric map is specified, the second method is more likely to respect the field. An improvement of the first method might be to redefine the skeleton with respect to the specified metric map, which is equivalent to supply the domain with a Riemannian structure. This latter is a theoretically difficult problem.

1.2. Indirect approaches

Given a triangular mesh of the domain, this approach aims at combining triangles into quadrilaterals[12–17] and leads to two related merging processes. The triangle merging procedure

1. is driven by the quadrilateral quality and may lead to mixed triangular–quadrilateral meshes, or
2. starts from the boundary and moves to the interior of the domain, ensuring an even number of vertices when two fronts collide and results in pure quadrilateral meshes (if the boundary discretization has an even number of vertices).

The first method is conceptually very simple, whereas the second method requires a topological classification of the front collisions and eventually adds vertices to the initial mesh.

We propose an indirect-based method of the first approach. The main contribution of this paper is to extend the triangle merging procedure to the case where a generalized metric map is specified. In addition, we introduce a new mesh optimization technique based on vertex smoothing. In Section 2, we recall the unit triangular mesh concept in order to construct the triangular mesh governed by a generalized metric map. The adaption scheme is given in Section 3. In Section 4, we describe the triangular to quadrilateral mesh conversion algorithm. Mesh optimization using vertex smoothing is detailed in Section 5. Some numerical results, academic tests and a CFD example, are discussed in Section 6 and future works and extension are given in the last section.

2. UNIT TRIANGULAR MESH

Let Ω be a domain of $\mathbb{R}^2$, supplied with a Riemannian structure (see Reference [18] for a survey on Riemannian differential geometry). We will discuss the problem of meshing Ω in such a way that every edge in the resulting mesh is of unit length size. This mesh is called the *unit mesh* of the domain Ω. The Riemannian structure is used to govern the mesh generation process. In practice, this structure is defined, using an interpolation, from a discrete field of metrics associated with the vertices of a given mesh of Ω (a background mesh). The metric at a mesh vertex P is defined by a symmetric positive definite 2×2 matrix, denoted as $\mathscr{M}_2$ and given by

$$\mathscr{M}_2(P) = \begin{pmatrix} a(P) & b(P) \\ b(P) & c(P) \end{pmatrix} \tag{1}$$

where $a(P) > 0$ and $a(P)c(P) - b^2(P) > 0$. The aim is to obtain a unit mesh with respect to the field $\mathscr{M}_2$, i.e. such that every mesh edge $[PX]$ connected to P satisfies at best the relationship (cf. Figure 1):

$${}^t\overrightarrow{PX}\,\mathscr{M}_2(P)\,\overrightarrow{PX} = 1 \tag{2}$$

The structure defined by the background mesh and the discrete field of metrics is called the *control space*. The meshing process of Ω, with respect to the control space, includes the following two stages:

(1) the generation of the unit mesh of the boundary of Ω and
(2) the generation of the unit mesh of Ω using the unit mesh of its boundary as input.

Afterwards, we briefly recall the main features of the unit meshing algorithm. For more details see References 8 and 19.

2.1. Unit mesh of the boundary

We assume that a mathematical model defines the geometry of the domain. For example, this model can be derived from a sufficiently fine initial boundary discretization. Thus, a unit mesh of the boundary with respect to a given Riemannian structure is a discretization of this model, obtained by subdividing it with *unit arc length* segments with respect to the control space. To this end, we can approximate the model with a polygonal segment, which is later subdivided into unit length segments.

2.2. Unit mesh of the domain

The unit mesh of the boundary of Ω provides a set of *constrained edges* having as endpoints a set of points, denoted as $\mathscr{S}(\Omega)$. At first, an initial constrained mesh of Ω is generated, whose sole vertices are the members of $\mathscr{S}(\Omega)$, respecting the constrained edges. A new mesh is then obtained by adding iteratively field points inside this mesh, and optimized so as to obtain the unit mesh of the domain. The mesh is initialized by the initial constrained mesh. At each iteration, the internal mesh edges are analysed and the internal field points

(i) are generated along the edges so as to subdivide them into unit length segments (new points must not be too close to the existing vertices) and

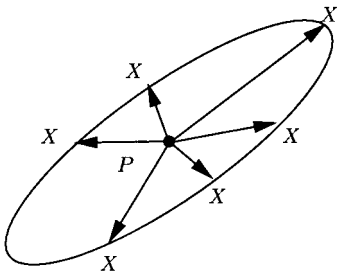

Figure 1. Unit circle, anisotropic case

(ii) are inserted into the current mesh via the *constrained* Delaunay kernel applied in a Riemannian context.

This process is repeated as long as the current mesh is modified. Afterwards, we review the generalized constrained Delaunay kernel, the length of a segment computation and the optimization procedure.

Generalized constrained Delaunay kernel. The classical constrained Delaunay kernel is a procedure resulting in the insertion of one internal point in a (Delaunay) triangulation, based on a *proximity criterion*. Formally speaking, the constrained Delaunay kernel can be written as[20]

$$T = T - C(P) + B(P) \tag{3}$$

where $C(P)$ is the cavity associated with point P and $B(P)$ is the triangulation of $C(P)$ enclosing P as a vertex, T denoting the current Delaunay mesh. The cavity is constructed using a proximity criterion, written as

$$\{K, K \in T, P \in \mathrm{Disc}(K) \quad \text{and} \quad P \text{ visible from any vertex of } K\} \tag{4}$$

where $\mathrm{Disc}(K)$ is the circumdisc of K.

The generalization of this procedure consists in redefining the cavity $C(P)$ in a Riemannian context.[21] Therefore, we define first the *Delaunay measure* $\alpha_{\mathscr{M}_2}$ associated with the pair (P,K), with respect to a given metric $\mathscr{M}_2$

$$\alpha_{\mathscr{M}_2}(P,K) = \left[\frac{d(O_K,P)}{r_K}\right]_{\mathscr{M}_2} \tag{5}$$

where O_K (resp. r_K) is the centre (resp. radius) of the circumdisc of K and $[*]_{\mathscr{M}_2}$ indicates that the quantity $*$ is evaluated in the euclidean space characterized by the metric $\mathscr{M}_2$. The usual proximity criterion, $P \in Disc(K)$, is expressed as $\alpha_{\mathscr{I}_2}(P,K) < 1$, where $\mathscr{I}_2$ is the identity metric. The cavity $C(P)$ is then redefined as

$$C(P) = C_1(P) \cup C_2(P) \tag{6}$$

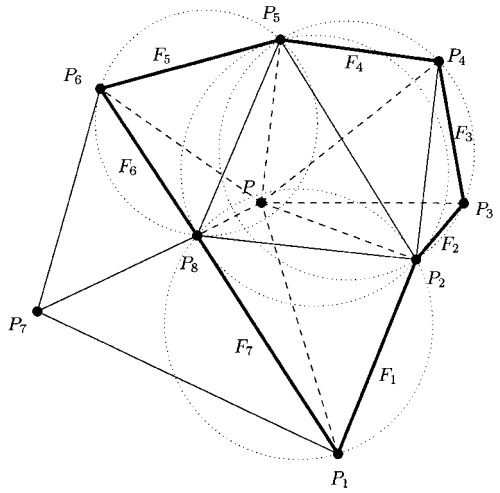

Figure 2. Delaunay kernel

with

$$C_1(P) = \{K,\ K \in T,\ K \text{ including } P\} \tag{7}$$

$$C_2(P) = \{K,\ K \in T,\ \exists K' \in C(P),\quad K \text{ adjacent to } K'$$

$$\alpha_{\mathcal{M}(P)}(P,K) \prod_V \alpha_{\mathcal{M}(V)}(P,K) < 1,\quad V \text{ vertex of } K \tag{8}$$

$$P \text{ visible from the vertices of } K\}$$

Hence, the $C(P)$ is constructed by adjacency from the elements of $C_1(P)$. With this definition, we can deduce that the generalized cavity is star-shaped with respect to P and the triangulation of $B(P)$ is then valid. Figure 2 shows $C_1(P) = [P_2P_5P_8]$ and $C_2(P) = [P_1P_2P_8] \cup [P_2P_4P_5] \cup [P_2P_3P_4] \cup [P_5P_6P_8]$ in the isotropic case.

Length of a segment. The length of a segment $[PQ] = (P + t\overrightarrow{PQ})_{0 \leqslant t \leqslant 1}$ of Ω is given by

$$l([PQ]) = \int_0^1 \sqrt{{}^t\overrightarrow{PQ}\,\mathcal{M}_2(P + t\overrightarrow{PQ})\overrightarrow{PQ}}\,\mathrm{d}t \tag{9}$$

where $\mathcal{M}_2(P + t\overrightarrow{PQ})$ is the metric at point $P + t\overrightarrow{PQ}$ of the segment $[PQ]$, this metric being well defined through the control space.

Optimization. This procedure aims at improving the mesh edge lengths, by means of

(i) edge swapping, if the alternative edges are closer to the unit length and
(ii) optimal vertex smoothing, which consists in moving P, 'step-by-step', to a position where all incident edges are close to the unit length size.

3. ADAPTION SCHEME

If the control space, consisting of the initial mesh and the relative discrete metric map, does not match exactly the ideal continuous field of metrics, the discrete map is enriched to govern the meshing process more accurately. The whole procedure is then rerun. Indeed, the unit mesh generation scheme can be easily extended to construct a loop of adaption. To this end, the control space at each iteration i is defined by the mesh at iteration $i-1$ and a metric map specified at all vertices of this mesh. The adaption scheme is given as follows:

(i) Data of F_{geom}, the initial discretization of the boundary of Ω serving to construct a smooth geometry $Model_{\text{geom}}$ of the domain Ω.
(ii) Initial discretization F_0 of $Model_{\text{geom}}$ according to a size map H_0.
(iii) Construction of the initial mesh T_0 using F_0 and H_0 as data.
(iv) Adaption loop (starting at $i=1$)
  (1) data of a metric map H_i on T_{i-1}.
  (2) discretization F_i of model $Model_{\text{geom}}$ governed by the control space (T_{i-1}, H_i).
  (3) adapted mesh T_i using F_i and the control space (T_{i-1}, H_i).
  (4) iteration $i=i+1$, if required.

The above scheme is iterated until an almost satisfactory mesh T_i is obtained with respect to (T_i, H_{i+1}). In other words, the edges in T_i have a length almost equal to unity in the control space. In this case, the mesh T_i is said to be compatible with the metric maps H_i or H_{i+1}.

4. CONVERSION TO UNIT QUADRILATERAL MESH

Given a unit mesh and an associated compatible metric map, the main idea of the conversion to unit quadrilateral mesh consists in pairing the triangles to form quadrilateral elements by controlling the quality. Obviously, this well-known strategy cannot always produce a pure quadrilateral mesh and may lead to a mixed mesh, composed of both triangular and quadrilateral elements. The proposed method generalizes the merging process to the case of a domain supplied with a Riemannian structure and tends to create optimal quadrilaterals via the specified structure. Another goal can be to minimize the number of *isolated island triangles*.

Therefore, we introduce several preliminary definitions, prior to define the quadrilateral quality and we propose an algorithm suitable for the quadrilateral meshing process with respect to the given objectives.

4.1. Preliminary definitions

We review the notion of the scalar product, the angle measure and the angle quality with respect to a given metric and introduce the "generalized" quality for a quadrilateral.

4.1.1. Scalar product with respect to a metric. The scalar product of two given vectors $\mathbf{u}$ and $\mathbf{v}$ in the euclidean space characterized by a metric $\mathcal{M}_2(X)$ is given by

$$[\langle \mathbf{u}, \mathbf{v} \rangle]_{\mathcal{M}_2(X)} = {}^t\mathbf{u}\,\mathcal{M}_2(X)\mathbf{v} \tag{10}$$

and the norm of a vector $\mathbf{w}$ by

$$[\|\mathbf{w}\|]_{\mathcal{M}_2(X)} = \sqrt{{}^t\mathbf{w}\mathcal{M}_2(X)\mathbf{w}} \tag{11}$$

4.1.2. Angle measure with respect to a metric. Let ABC be three points such that $\overrightarrow{BC} \times \overrightarrow{BA} > 0$ where $\times$ denotes the cross product in the usual euclidean metric. The angle measure in radians of the angle between the two vectors $\overrightarrow{BC}$ and $\overrightarrow{BA}$ with respect to a metric $\mathcal{M}_2(X)$ is given by

$$[\theta_B]_{\mathcal{M}_2(X)} = [\widehat{(\overrightarrow{BC}, \overrightarrow{BA})}]_{\mathcal{M}_2(X)} = \arccos\left(\frac{[\langle \overrightarrow{BC}, \overrightarrow{BA}\rangle]_{\mathcal{M}_2(X)}}{[\|\overrightarrow{BC}\|]_{\mathcal{M}_2(X)}[\|\overrightarrow{BA}\|]_{\mathcal{M}_2(X)}}\right) \tag{12}$$

Remark 1. In the case where $\overrightarrow{BC} \times \overrightarrow{BA} < 0$ the angle is then given by

$$[\theta_B]_{\mathcal{M}_2(X)} = 2\pi - [\widehat{(\overrightarrow{BA}, \overrightarrow{BC})}]_{\mathcal{M}_2(X)} \tag{13}$$

4.1.3. Quality of a quadrilateral element. Let $\mathcal{Q} = [ABCD]$ be a quadrilateral element where A, B, C, D are given in counterclockwise order. The quality of $\mathcal{Q}$ can be defined from the measure of the angle (in $[0, 2\pi]$) qualities of $\mathcal{Q}$. Indeed, this measure is sufficient to capture a high aspect ratio element if it is coupled with the edge length control. For instance, a long rectangle will not be optimal with respect to an isotropic metric. This latter is optimal for an angle value of $\pi/2$ radians and decreases as much as its difference with this value increases. Moreover, the angle quality is equal to 0 for an angle measure in $[\pi, 2\pi]$. By normalizing the optimal value to 1 and by considering, for instance, a linear variation of the quality, it is possible to define an angle quality function ϕ with respect to a given metric $\mathcal{M}_2(X)$, as (cf. Figure 3)

$$\phi([\theta]_{\mathcal{M}_2(X)}) = \begin{cases} \dfrac{2[\theta]_{\mathcal{M}_2(X)}}{\pi} & \text{if } 0 \leqslant [\theta]_{\mathcal{M}_2(X)} < \pi/2 \\ 2 - \dfrac{2[\theta]_{\mathcal{M}_2(X)}}{\pi} & \text{if } \dfrac{\pi}{2} \leqslant [\theta]_{\mathcal{M}_2(X)} < \pi \\ 0 & \text{if } \pi \leqslant [\theta]_{\mathcal{M}_2(X)} \end{cases} \tag{14}$$

where $[\theta]_{\mathcal{M}_2(X)}$, $0 \leqslant [\theta]_{\mathcal{M}_2(X)} \leqslant 2\pi$, represents the measure of an angle (in radians) in the metric $\mathcal{M}_2(X)$.

The quality $q(\mathcal{Q})$ of $\mathcal{Q}$ is then defined as the minimum value of the angle quality function, computed with respect to the metrics associated to its vertices. Hence, we have:

$$q(\mathcal{Q}) = \min_{(X,Y)\in\{A,B,C,D\}} \phi([\theta_Y]_{\mathcal{M}_2(X)}) \tag{15}$$

Remark 2. The quality of a quadrilateral element in the Riemannian case requires the evaluation of 16 quality measures in the euclidean case. A simplified version which involves 4 euclidean quality measures consists of defining the quality as

$$q(\mathcal{Q}) = \min_{Y\in\{A,B,C,D\}} \phi([\theta_Y]_{\mathcal{M}_2(Y)}) \tag{16}$$

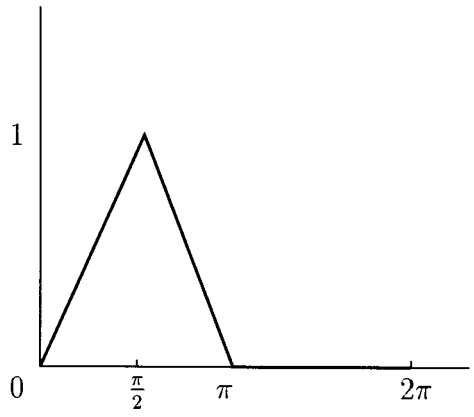

Figure 3. Angle quality

4.2. Scheme of the method

It is possible to govern the triangle combination procedure by using the usual approach described hereafter. Isolated island triangles may be present in the final mesh. To obtain a pure quadrilateral mesh, we review a standard triangle-removal procedure.

4.2.1. Triangle merging procedure. Let $\mathcal{T}$ be the initial unit triangular mesh. Each pair of adjacent triangles is likely to form a quadrilateral element and is identified by the common edge. A simplified measure of the corresponding quadrilateral quality is associated with each edge (cf. Figure 3). Let $[ABC]$ and $[ACD]$ be two adjacent triangles, sharing the edge $a=[AC]$. If $\theta_A=(\widehat{\overrightarrow{AB},\overrightarrow{AD}})$ and $\theta_C=(\widehat{\overrightarrow{CD},\overrightarrow{CB}})$, the quality of a can then be defined as (cf. Figure 4)

$$q(a)=\min([\theta_A]_{\mathcal{M}_2(A)},[\theta_A]_{\mathcal{M}_2(C)},[\theta_C]_{\mathcal{M}_2(A)},[\theta_C]_{\mathcal{M}_2(C)}) \tag{17}$$

The edge list of $\mathcal{T}$ is sorted in the decreasing order according to the quality, the corresponding quadrilaterals are constructed and their edges are removed from the list. The merging process is thus governed by the quality measure, even if the number of isolated island triangles is not minimized. It is possible to reduce the number of triangles by governing the combination of triangles by adjacency, from the existing quadrilaterals. To this end, it is sufficient to redefine the edge quality $q^*(a)$ of edge a as

$$q^*(a)=q(a)-q([AB])-q([BC])-q([CD])-q([DA]) \tag{18}$$

Remark 3. With this new definition, the edge list must be updated after each quadrilateral creation. This operation can be efficiently implemented using a heap structure. In addition, two parameters, $q_t(=0{\cdot}1)$ and $q_d(=-0{\cdot}5)$, are used in order to control the triangles merging:

(i) if $q(a)<q_t$ then the edge a is not taken into account,
(ii) if the edge a is a boundary edge or if it is an edge of a newly created quadrilateral then set $q^*(a)=q_d$.

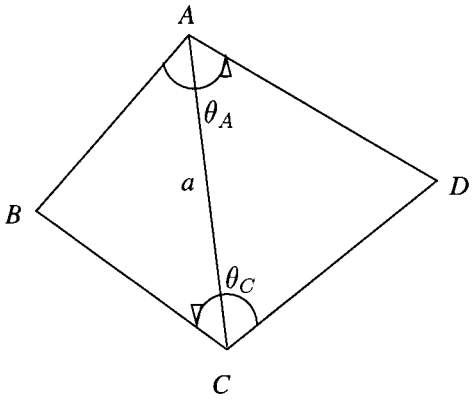

Figure 4. Edge quality

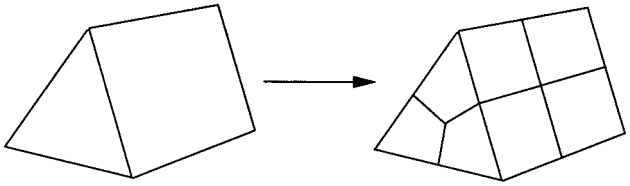

Figure 5. Mesh refinement

The merging process by adjacency has an undesirable side effect, the mesh quality is globally degraded.

4.2.2. Conversion of isolated island triangles to quadrilaterals. Each isolated island triangle is split into three quadrilaterals according to the usual scheme, adding three vertices in the middle of the edges and a vertex at the barycenter of the element. To preserve the mesh conformity, a similar process is also applied to all quadrilaterals, thus propagating a uniform refinement.

Remark 4. Mesh refinement (Figure 5) has the disadvantage of producing a mesh that is no longer compatible with the specified metric map. It is possible to retrieve the compatibility by governing the initial triangular mesh creation process using the metric map $\{\frac{1}{4}\mathcal{M}_2(P)\}$, where $\{\mathcal{M}_2(P)\}$ is the initial field. In the isotropic case, this is equivalent to multiply the specified vertex size by two.

5. MESH OPTIMIZATION USING VERTEX SMOOTHING

This last stage is required because the initial mesh $\mathcal{T}$ is optimal with respect to the triangle quality instead of the quadrilateral quality and because of the introduction of the extra vertices during the conversion of the isolated island triangles. Usually, this step consists in moving the mesh vertices in order to optimize the edge lengths. In the case of a triangular mesh, this scheme leads to a satisfactory result. However, the algorithm cannot be applied to quadrilateral meshes. Actually, a

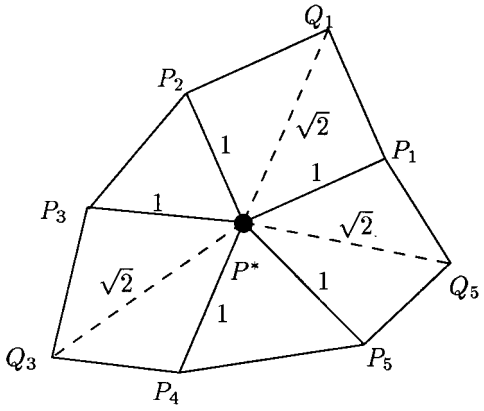

Figure 6. Optimal position of point P

diamond-shaped element is optimal with respect to the edge length criteria. Therefore, we propose a modified technique which consists in using the diagonals of the quadrilaterals.

Let P be a mesh vertex, n (resp. m) be the number of quadrilaterals $\{[PP_iQ_iP_{i+1}]\}$ (resp. triangles $\{[PP_iP_{i+1}]\}$) sharing P, $[PP_i]$ are the edges and $[PQ_i]$ the diagonals of elements. We define the points P_i^* and Q_i^* such that

(i) $\overrightarrow{P_iP_i^*}$ is colinear to $\overrightarrow{P_iP}$, the length of $[P_iP_i^*]$ is equal to the unit length (in the control space) and P_i and P_i^* are on the same side of P. Point P_i^* is given by the approximate formula:

$$P_i^* \approx P_i + \frac{\overrightarrow{P_iP}}{l([P_iP])} \tag{19}$$

(ii) $\overrightarrow{Q_iQ_i^*}$ is colinear to $\overrightarrow{Q_iP}$, the length of the segment $[Q_iQ_i^*]$ is equal to $\sqrt{2}$ and Q_i and Q_i^* are on the same side of P. Point Q_i^* is given by the approximate formula:

$$Q_i^* \approx Q_i + \sqrt{2}\frac{\overrightarrow{Q_iP}}{l([Q_iP])} \tag{20}$$

The smoothing procedure consists in moving the point P step-by-step toward the point P^*, barycentre of the points P_i^* and Q_i^*, if the quality of the worst element of $\{[PP_iQ_iP_{i+1}]\}$ is improved. The new position of P is given by

$$P + \omega \overrightarrow{PP^*} \tag{21}$$

where ω is a relaxation parameter. Figure 6 shows the optimal position of point P.

Remark 5. The length of each segment is evaluated via the integral of the equation (9). It is sufficient to apply a Simpson formula with three points (the two endpoints and the middle point) to evaluate the integral.

If the final mesh is not satisfactory with respect to the normalized edge lengths, additional quality optimization procedures based on local topological modifications can be used.[13]

6. NUMERICAL RESULTS

In this section, two class of examples are given, concerning two academic tests and a CFD problem. In the academic tests, the metric map used to govern the mesh generation is user-specified. In the CFD example, the metric map is given by an *a posteriori* error estimate based on the hessian of the solution.

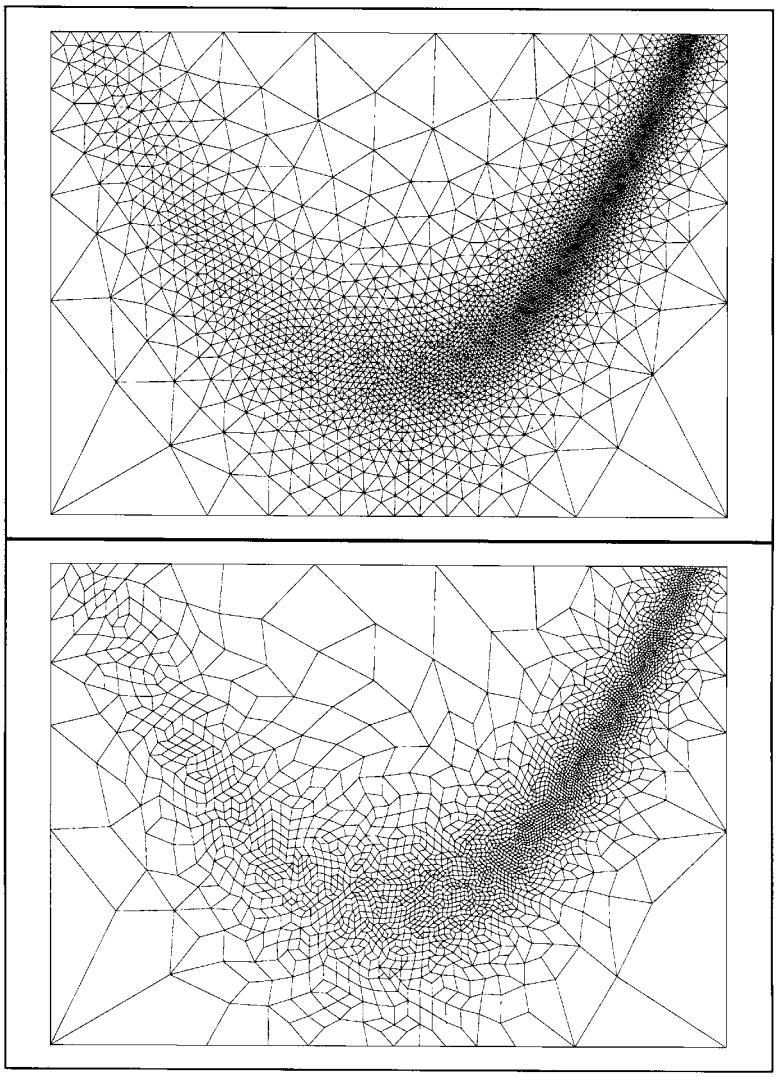

Figure 7. Triangular adaptive mesh and mixed triangular–quadrilateral mesh after triangle merging

6.1. Academic test examples

Two academic test examples are given.

6.1.1. Example 1. In the first test (by Lewis et al.[14]), the domain is the area of a rectangle defined by

$$-1{\cdot}25 \leqslant x \leqslant 1{\cdot}25, \quad -0{\cdot}5 \leqslant y \leqslant 1{\cdot}25 \tag{22}$$

supplied with an isotropic metric map indicating at each (x, y) of the domain the desired sizes $h(x, y)$:

$$h(x, y) = 0{\cdot}01[1 + f(x, y)] \tag{23}$$

where

$$f(x, y) = 30(y - x^2)^2 + (1 - x)^2 \tag{24}$$

The unit mesh with respect to the isotropic map is shown in Figure 7 (top). This mesh contains 3500 vertices and 6933 triangles. The normalized edge lengths range from 0·57 to 1·56, with 95 per cent between 0·75 and 1·33. Figure 7 (bottom) shows the mixed mesh after the triangle merging which contains 3417 (97 per cent) quadrilaterals and 99 (3 per cent) triangles. The corresponding mean quality is 0·63. Figure 8 (top) illustrates the mixed mesh after the refinement procedure. The final mesh after optimization, Figure 8 (bottom), contains 14 023 vertices and 13 965 quadrilaterals, the worst quality is 0·26 (2 elements) and the mean quality is 0·75, 91 per cent of the vertices are of degree 4 and the minimal and maximal values are 3 and 6, respectively. The normalized edge lengths of this mesh range from 0·4 (9 edges) to 1·7 and 90 per cent between 0·75 and 1·33. The whole process has taken 15 s of CPU time on a HP735/99 MHz machine.

6.1.2. Example 2. In the second test, the domain is the area of a unit square defined by

$$0 \leqslant x \leqslant 1, \quad 0 \leqslant y \leqslant 1 \tag{25}$$

without the area of a disk of radius 0·15 and centre $(0{\cdot}75, 0{\cdot}75)$. We considered two metric maps, isotropic and anisotropic. The isotropic map requires a size of 0·003 along the medial axis of the domain and a linear size grow-up from the medial axis to the interior of the domain. The anisotropic map requires stretched elements along the medial axis and a linear variation of the metrics in order to have isotropic elements from the medial axis to the interior of the domain. The corresponding meshes (as in Example 1) for the isotropic map are given in Figure 9. After the merging process, 96 per cent of the elements are quadrilaterals. The minimal quality of the final mesh is 0·3 and the mean quality is 0·73. The mesh contains 46 438 edges and the corresponding normalized edge lengths range between 0·4 (16 edges) and 2·0 (7 edges), 87 per cent between 0·75 and 1·33. The total speed-up of the quadrilateral mesh generation is 1400 quadrilaterals per second.

The meshes for the anisotropic map are given in Figure 10. After the merging process, 97 per cent of the elements are quadrilaterals. The minimal quality of the final mesh is 0·1 and the

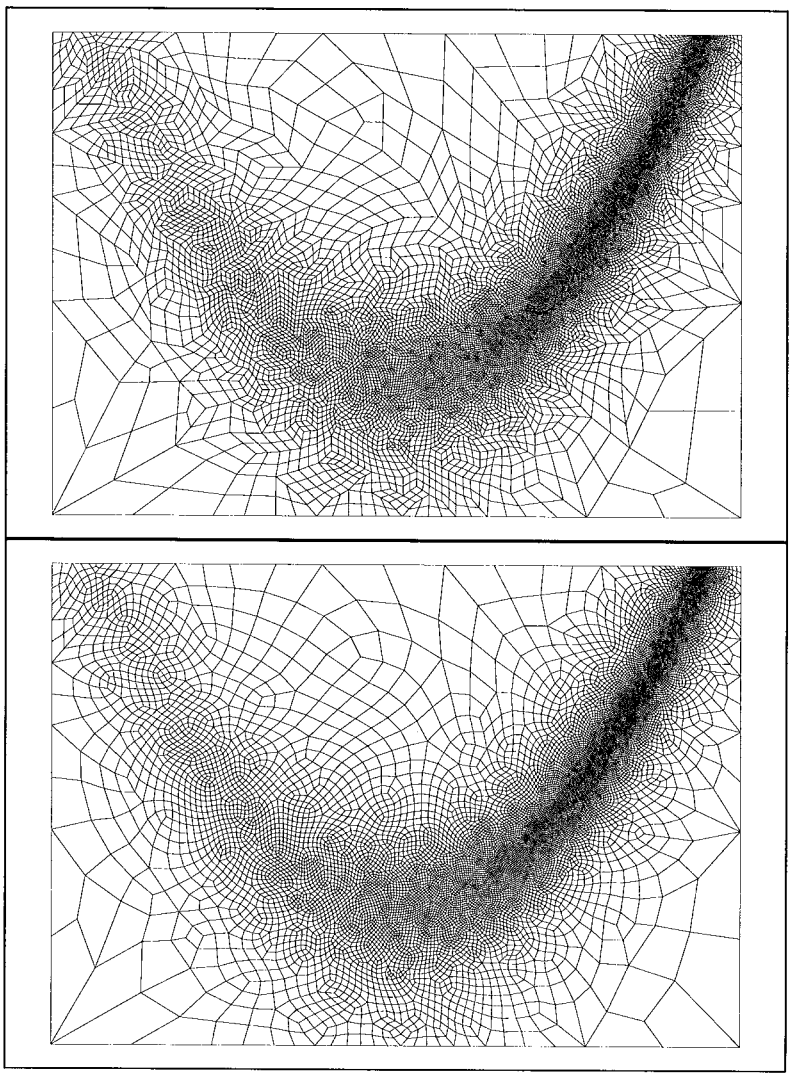

Figure 8. Mixed mesh after refinement and final mesh after optimization

mean quality is 0·68. The mesh contains 25 381 edges and the normalized edge lengths range between 0·35 (20 edges) and 2·5 (5 edges), 88 per cent between 0·75 and 1·33. The stretching factor along the medial axis is about 20. The speed-up of the quadrilateral mesh generation is 1300 quadrilaterals per second.

6.2. CFD example

A Mach 2 supersonic flow at Reynolds 10 000 is applied to a NACA0012 airfoil. The triangular mesh has been adapted six times in the anisotropic context in order to identify the shock, boundary

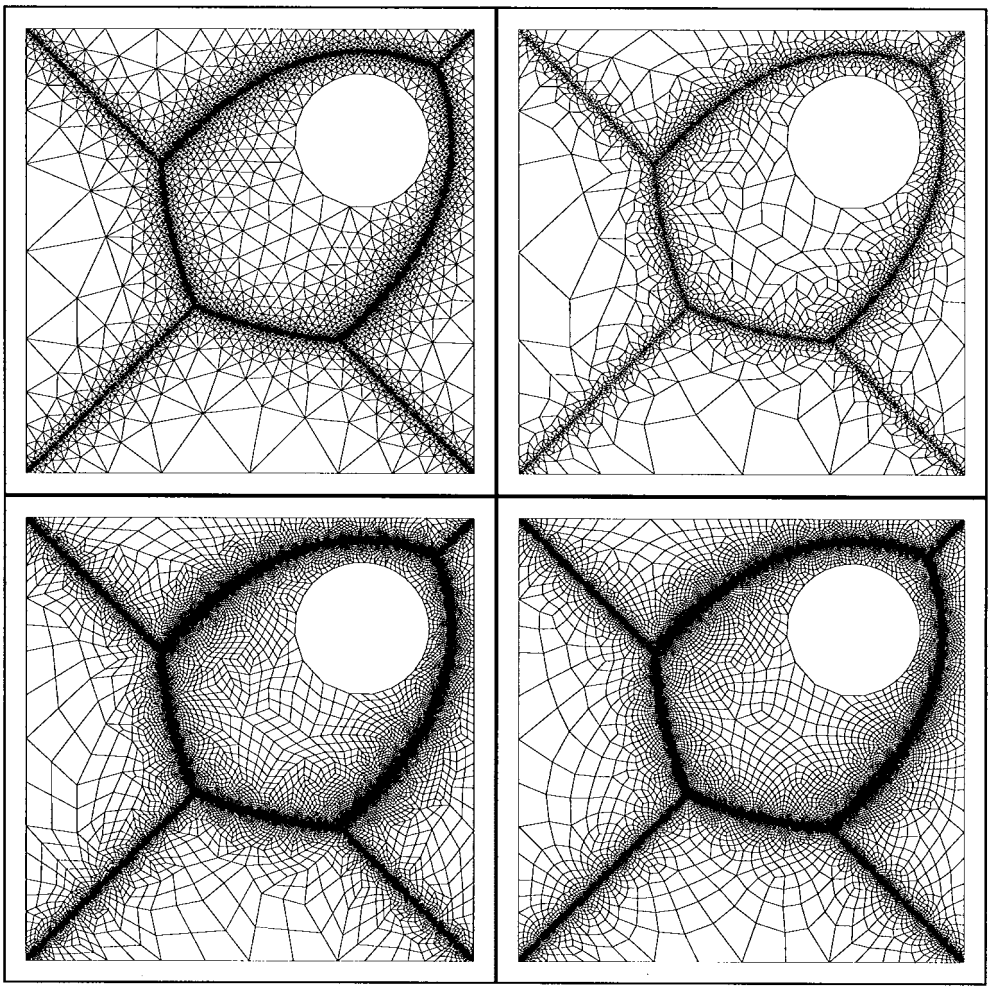

Figure 9. Meshes of Example 2, isotropic map

layer and wake regions and even the weak shocks at the trailing edge have been detected. Each mesh adaption has been followed by 500 iterations of a Navier–Stokes triangle-based solver[16]. The quadrilateral mesh has been obtained from the triangular mesh and the associated metric map, to underscore the capabilities of the proposed triangle merging technique. The whole process has taken 3620 s of CPU time (HP735/99 MHz), 3600 s for the CFD solver and 20 s for the meshing procedure.

Figure 11 shows the initial isotropic mesh (iteration 0) with 1337 vertices and 2668 triangles, the corresponding quadrilateral mesh with 5238 vertices and 5109 quadrilaterals. Similarly, Figure 12 shows the final anisotropic mesh (iteration 6) with 3846 vertices and 7686 triangles and the corresponding quadrilateral mesh with 15 272 vertices and 14 897 quadrilaterals, the stretching factor is about 286.

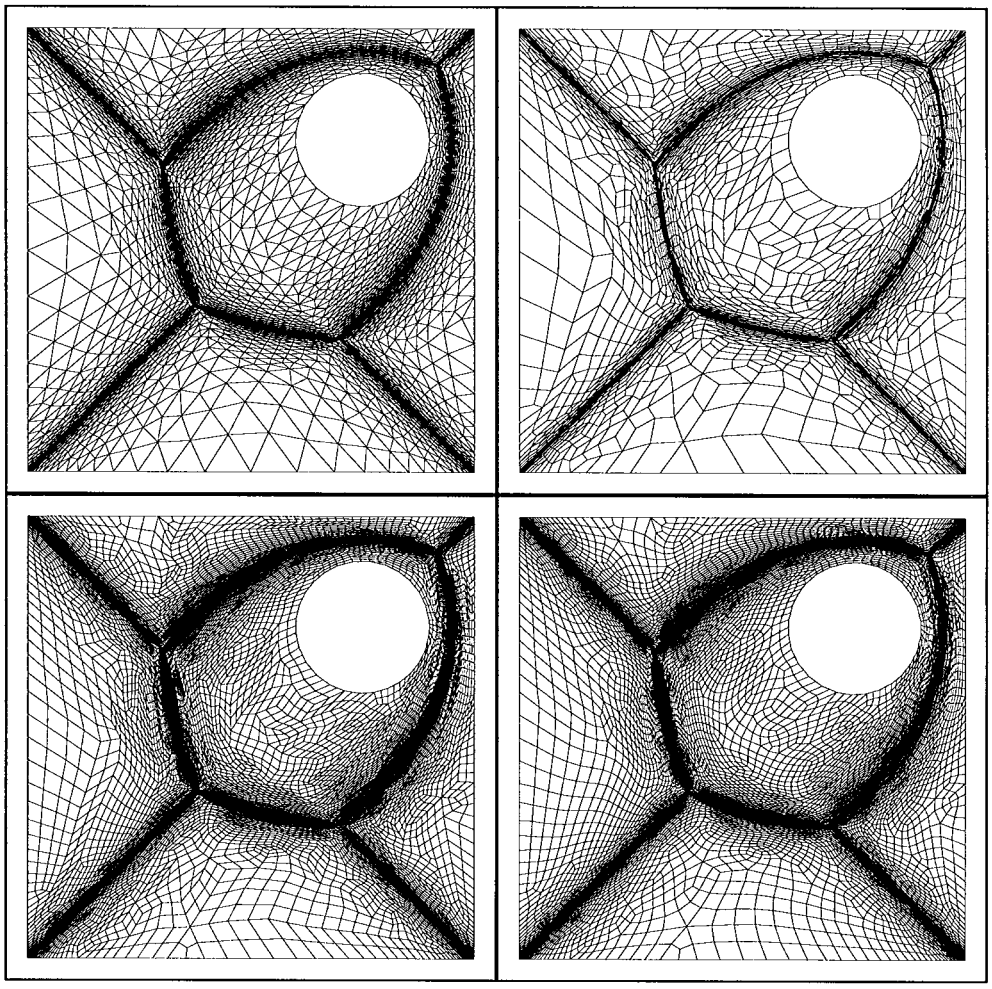

Figure 10. Meshes of Example 2, anisotropic map

The metric map related to the final mesh is shown on Figure 13, where at each vertex the associated ellipse represents the desired size in any direction. See Reference 5 for more details about the computation of the anisotropic metric map, derived from an *a posteriori* error estimate based on second-order derivatives of the solution.

The histogram of Figure 15 shows the triangular mesh quality[8] (the quality is a function varying in $[0, 1]$). The worst value is 0·1 and 90 per cent of the elements have a quality better than 0·5. The triangular mesh contains 11 151 edges. The histogram of Figure 16 shows the normalized edge lengths of the triangular mesh. The minimum value is 0·31, the maximum value is 9·87, 37 edges have a normalized edge length lesser than 0·5 and 120 have a normalized edge length bigger than 2. This result can be explained by the discontinuities of the metric map near the shock regions (cf. Figure 14).

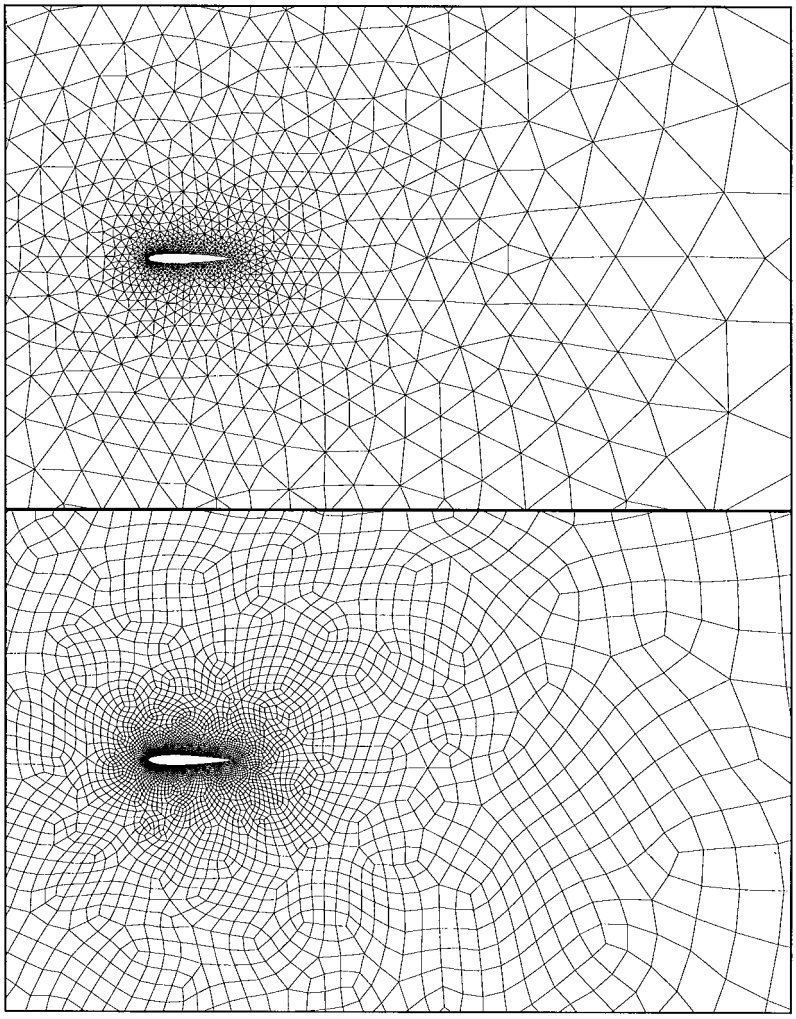

Figure 11. CFD meshes, initial triangular and quadrilateral isotropic meshes

The histogram of Figure 17 shows the quadrilateral mesh quality. The minimum value is 0·05 and the mean quality is 0·64. The quadrilateral mesh contains 30 169 edges. The histogram of Figure 18 shows the normalized edge lengths of the quadrilateral mesh. The minimum value is 0·07, the maximum value is 15·56, 578 edges have a normalized edge length lesser than 0·5 and 241 have a normalized edge length bigger than 2.

We can appreciate in Figure 19 the quality of the solution after adaptive mesh generation applied to this CFD problem. Figure 19 (top) shows the solution obtained using the fixed initial mesh after 3000 iterations of the Navier–Stokes solver and Figure 19 (bottom) illustrates the iso-density contour lines of the solutions using adaption.

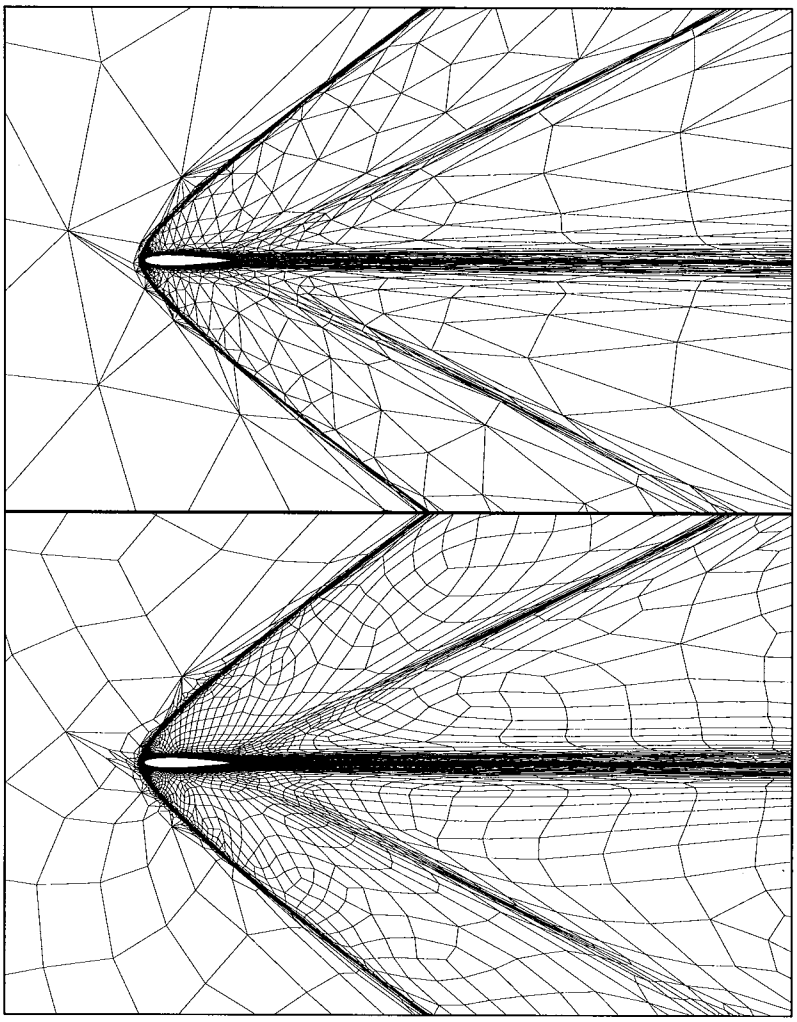

Figure 12. CFD meshes, final adapted triangular and quadrilateral anisotropic meshes

7. CONCLUSIONS

In this paper, we have briefly reviewed an adaptive mesh generation method governed by a Riemannian discrete metric map. In this context, we have introduced the concept of a unit mesh with respect to a Riemannian structure. Then, we have proposed an automatic triangular to quadrilateral mesh conversion scheme, which takes the governing control space into account. Finally, we have introduced a vertex smoothing procedure based on normalized unit edge lengths. Several examples, in particular a CFD test, illustrated the efficiency of the proposed method.

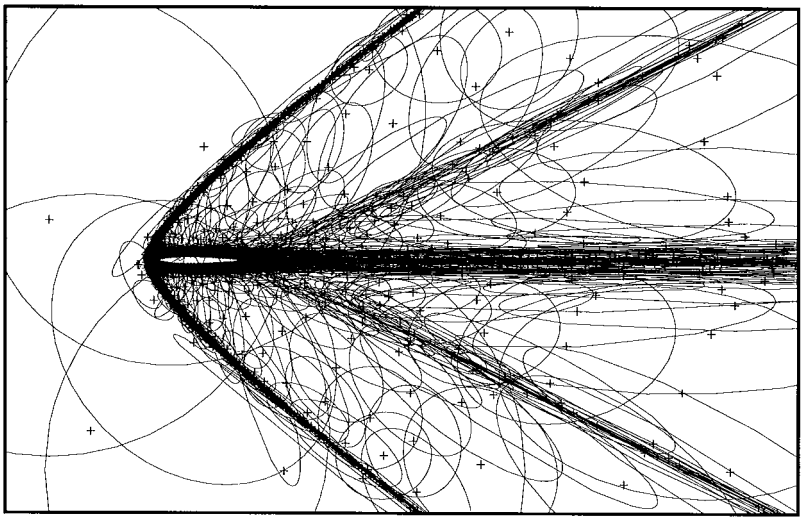

Figure 13. Metric map associated with the final adapted mesh

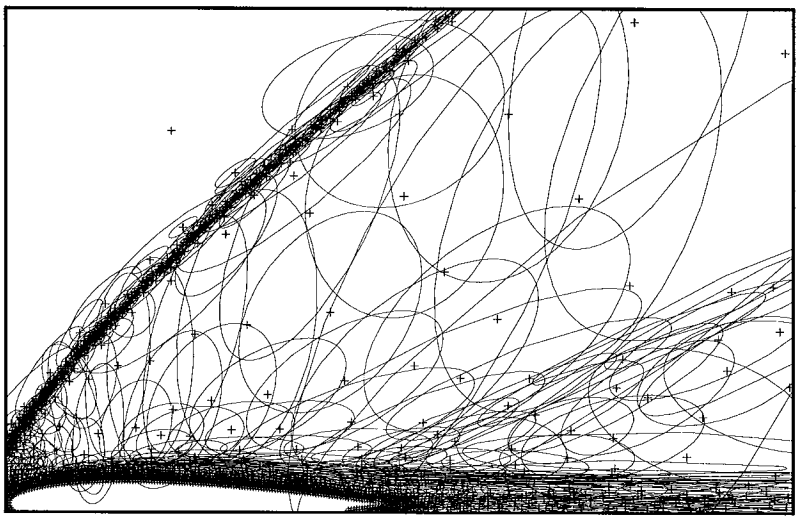

Figure 14. Metric map associated with the final adapted mesh (partial zoom)

A possible extension of the method consists in governing the merging process by adjacency using an advancing-front technique in order to obtain a pure quadrilateral mesh. This requires to generalize a topological classification of the configurations encountered in the merging of two fronts cases which eventually results in new vertices insertion. The proposed mesh generation scheme can be naturally extended to parametric surfaces triangular and quadrilateral mesh generation.[22]

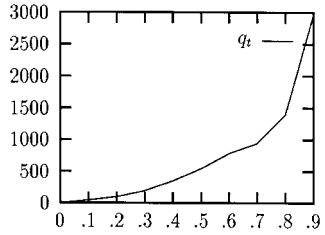

Figure 15. Triangular mesh quality

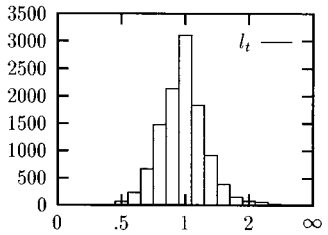

Figure 16. Normalized edge lengths

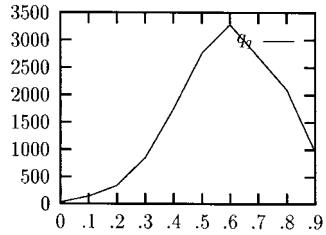

Figure 17. Quadrilateral mesh quality

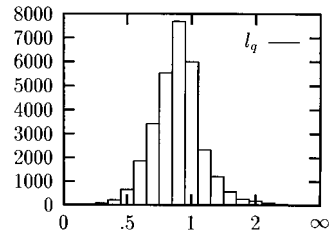

Figure 18. Normalized edge lengths

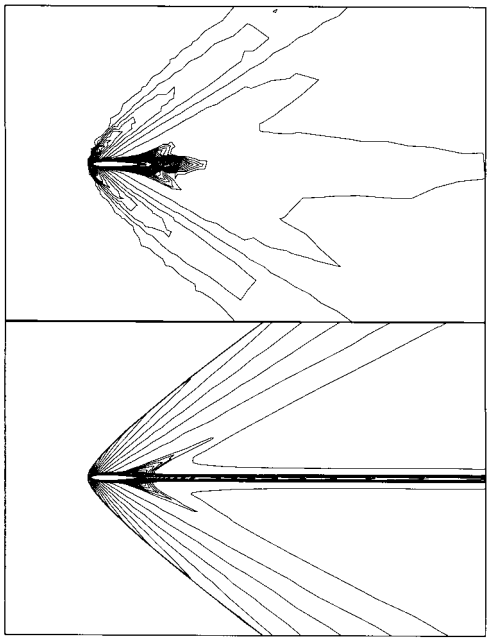

Figure 19. Iso-density contour lines of the solutions

REFERENCES

1. R. Lohner, 'Adaptive remeshing for transient problems', *Comput. Meth. Appl. Mech. Engng.*, **75,** 195–214 (1989).
2. D. J. Mavriplis, 'Adaptive mesh generation for viscous flows using Delaunay triangulation', *J. Comput. Phys.*, **90**(2), (1990).
3. J. Peraire, M. Vahdati, K. Morgan and O. C. Zienkiewicz, 'Adaptive remeshing for compressible flow computations', *J. Comput. Phys.*, **72,** 449–466 (1987).
4. N. P. Weatherill, M. J. Marchant, O. Hassan and D. L. Marcum, 'Grid adaptation using a distribution of sources applied to inviscid compressible flow simulations', *Int. J. Numer. Meth. Engng.*, **19,** 739–764 (1994).
5. M. J. Berger and A. Jameson, 'Automatic adaptive grid refinement for Euler equations', *AIAA J.*, **23,** 561–568 (1985).
6. M. C. Rivara, 'A 3-D refinement algorithm suitable for adaptive and multi-grid techniques', *Commun. Appl. Numer. Meth.*, **8,** 281–290 (1992).
7. M. J. Castro-Diaz, F. Hecht and B. Mohammadi, 'New progress in anisotropic grid adaptation for inviscid and viscous flows simulations', *Proc. 4th Int. Meshing Roundtable*, Albuquerque, NM, October 16–17, 1995.
8. H. Borouchaki, P. L. George, F. Hecht, P. Laug and E. Saltel, 'Delaunay mesh generation governed by metric specifications. Part I: Algorithms', *Finite Elements Anal. Des.*, **25,** 61–83 (1997).
9. C. G. Armstrong, D. J. Robinson, R. M. McKeag, T. S. Li, S. J. Bridgett, R. J. Donaghy, 'Applications of the medial axis transform in analysis modelling', *Nafems*, *Proc. 5th Int. Conf. on Reliability of FEM for Engng. Appl.*, Amsterdam, Netherlands, 10–12 May, 1995, pp. 415–426.
10. J. A. Talbert and A. R. Parkinson, 'Development of an automatic two-dimensional finite element mesh generator using quadrilateral elements and Bezier curve boundary definition', *Int. J. Numer. Meth. Engng.*, **29,** 1551–1567 (1991).
11. T. D. Blacker and M. B. Stephenson, 'Paving: a new approach to automated quadrilateral mesh generation', *Int. J. Numer. Meth. Engng.*, **32,** 811–847 (1991).
12. B. P. Johnston, J. M. Sullivan and A. Kwasnik, 'Automatic conversion of triangular finite element meshes to quadrilateral elements', *Int. J. Numer. Meth. Engng.*, **31,** 67–84 (1991).
13. C. K. Lee and S. H. Lo, 'A new scheme for the generation of a graded quadrilateral mesh', *Comput. Struct.*, **52,** 847–857 (1994).
14. R. W. Lewis, Y. Zheng and A. S. Usmani, 'Aspects of adaptive mesh generation based on domain decomposition and Delaunay triangulation', *Finite Elements Anal. Des.*, **20,** 47–70 (1995).
15. S. H. Lo, 'Generating quadrilateral elements on plane and over curved surfaces', *Comput. Struct.*, **31,** 421–426 (1989).
16. E. Rank, M. Schweingruber and M. Sommer, 'Adaptive mesh generation', *Commun. Numer. Meth. Engng.*, **9,** 121–129 (1993).
17. J. Z. Zhu, O. C. Zienkiewicz, E. Hinton and J. Wu, 'A new approach to the development of automatic quadrilateral mesh generation', *Int. J. Numer. Meth. Engng.*, **32,** 849–866 (1991).
18. M. P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice-Hall, Englewood Cliffs, NJ, 1976.
19. H. Borouchaki, P. L. George and B. Mohammadi, 'Delaunay mesh generation governed by metric specifications. Part II: Applications', *Finite Elements Anal. Des.*, **25,** 85–109 (1997).
20. D. F. Watson, 'Computing the *n*-dimensional Delaunay tessellation with application to Voronoï polytopes', *Comput. J.*, **24,** 167–172 (1981).
21. H. Borouchaki and P. L. George, 'Triangulation de Delaunay et métrique riemannienne. Applications aux maillages éléments finis', *Revue européenne des éléments finis*, **5,** 323–340 (1996).
22. H. Borouchaki, P. J. Frey and P. L. George, 'Maillage de surfaces paramétriques. Part III: Elements quadrangulaires', *Rapport de Recherche INRIA*, RR-2954, 1996.
23. B. Mohammadi, 'Fluid Dynamics Computation with NSC2KE—a User guide, release 1.0', *Technical Report INRIA*, RT-164, 1994.

