INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **51**:1097–1115
Published online 20 December 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1151

3D anisotropic unstructured grid generation

A. Ghidoni[1,*,†], E. Pelizzari[2,‡], S. Rebay[1,§] and V. Selmin[2,¶]

[1]*Dipartimento di Ingegneria Meccanica, Università di Brescia, via Branze 38, 25123 Brescia, Italy*
[2]*RCRT, Alenia Aeronautica S.P.A., Corso Marche 41, I-10146 Torino, Italy*

SUMMARY

This paper describes a fully automatic 3D anisotropic mesh generation method for domains of arbitrary shape. The spacing of the boundary mesh is computed by the analysis of the principal curvatures and directions of the boundary surfaces. The spacing in the domain is obtained by interpolation of the spacing at the boundaries on a suitably constructed background mesh. Examples which illustrate the performance of the proposed methodology are presented. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: surface meshing; anisotropic mesh; background grid

1. INTRODUCTION

The availability of ever increasing computational resources and of efficient and robust CFD solution algorithms allows the numerical simulation of very complex flow fields. The generation of the grid is probably the bottleneck in terms of human resources required to complete a CFD simulation for domains of complex shape such as, for example, an aircraft configuration. The construction of a grid which minimizes the number of nodes required to compute a flow field within a prescribed tolerance level, or even of a grid which approximates the geometry of the domain with sufficient accuracy to allow the computation of an initial coarse solution in an adaptive solution process, remains in fact a difficult and time-consuming task with the commonly available software.

Among the many tasks which require substantial user input in grid generation, that of prescribing the mesh spacing for a complex domain is probably the most demanding in terms of human resources required. We here describe a 3D anisotropic unstructured grid-generation algorithm, which, instead, requires a very limited user input even for very complex geometries

*Correspondence to: Antonio Ghidoni, Dipartimento di Ingegneria Meccanica, Università di Brescia, via Branze 38, 25123 Brescia, Italy.
†E-mail: antonio.ghidoni@ing.unibs.it
‡E-mail: epelizzari@aeronautica.alenia.it
§E-mail: stefano.rebay@ing.unibs.it
¶E-mail: vselmin@aeronautica.alenia.it

Received 2 June 2005
Revised 4 November 2005
Accepted 7 November 2005

since the mesh spacing is automatically defined as a function of the curvature of the boundary of the domain. We follow the widely adopted approach [1, 2] which consists in augmenting the domain to be meshed with a metric field $\mathbf{M}(\boldsymbol{x})$ (i.e. to define a Riemannian structure on the domain) and in requiring that all the mesh edges have the same length with respect to $\mathbf{M}(\boldsymbol{x})$ (typically unit length edges lengths in $\mathbf{M}(\boldsymbol{x})$ are chosen). This leads to an anisotropic control of the mesh spacing, which means to control not only the mesh element size but also their aspect ratio. This approach decreases significantly the nodes number used when the flow field or the geometry of the domain are characterized by anisotropic features.

Following the ideas introduced in Reference [3], the metric field on the boundary is computed according to the principal curvatures and directions of the boundary surface. This approach is, however, not appropriate if the surface curvature is very small, since it leads to excessive mesh spacings. The generally adopted approach to deal with 'nearly flat' surfaces is to limit the spacing with a threshold value prescribed in an isotropic manner. As a consequence all the informations regarding the anisotropy of the mesh are completely lost in regions of small surface curvature and much of the potential savings allowed by anisotropic unstructured meshes are not fully realized in practice. This situation is typical in many applications such as, for example, high aspect ratio wings and turbomachinery bladings. An original anisotropic treatment for 'nearly flat' surfaces has been developed in this work. After establishing the metric field distribution on the boundary, we proceed with the surface mesh generation using an approach, sometimes referred to as the 'indirect approach', which consists in generating the mesh in the parametric domain associated to the surface and in mapping the resulting mesh onto the real surface. Using the surface mesh as input data it is then possible to build a background grid, which is used to interpolate the value of the boundary metric field at any other points. The computational domain is then discretized by using the 3D anisotropic mesh generator.

The anisotropic mesh-generation approach has been applied with success to the computation of inviscid flow around complex aeronautical configurations.

2. RIEMANNIAN METRIC

An anisotropic control of the mesh spacing has been chosen for this work, which means that we will be able to control not only the mesh elements size but also their aspect ratio. This is realized by requiring that all the edges of the mesh are of unit length with respect to a suitable defined metric field $\mathbf{M}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a point of $\mathbb{R}^d$ $(d=2,3)$. $\mathbf{M}(\boldsymbol{x})$ is a $(d \times d)$ symmetric positive definite matrix. For example, in two dimensions, we consider

$$\begin{pmatrix} a(\boldsymbol{x}) & b(\boldsymbol{x}) \\ b(\boldsymbol{x}) & c(\boldsymbol{x}) \end{pmatrix} \tag{1}$$

such that $a>0$, $c>0$ and $ac-b^2>0$, for $a,b,c \in \mathbb{R}$. If the field of tensors thus defined is known, it induces a Riemannian structure over $\mathbb{R}^d$.

The length with respect to $\mathbf{M}$ of an arbitrary mesh edge $\boldsymbol{v}$ is approximated as

$$\|\boldsymbol{v}\|_{\mathbf{M}} = \sqrt{\boldsymbol{v}^{\mathrm{T}}\mathbf{M}(\hat{\boldsymbol{x}})\boldsymbol{v}} \tag{2}$$

where $\hat{\boldsymbol{x}}$ is the midpoint of edge $\boldsymbol{v}$. When $\mathbf{M}(\boldsymbol{x})$ is independent of the position $\boldsymbol{x}$, we again find the classical Euclidean case, otherwise we are in the Riemannian context.

The matrix $\mathbf{M}(\boldsymbol{x})$ can be factorized as

$$\mathbf{M}(\boldsymbol{x}) = \mathbf{R}(\boldsymbol{x})\boldsymbol{\Lambda}(\boldsymbol{x})\mathbf{R}(\boldsymbol{x})^{\mathrm{T}} \tag{3}$$

where

$$\mathbf{R}(\boldsymbol{x}) = [\boldsymbol{r}_1(\boldsymbol{x}), \ldots, \boldsymbol{r}_n(\boldsymbol{x})] \tag{4}$$

is the orthogonal matrix of the normalized (right) eigenvectors of $\mathbf{M}(\boldsymbol{x})$ and

$$\boldsymbol{\Lambda}(\boldsymbol{x}) = \mathrm{diag}(\gamma_1(\boldsymbol{x}), \ldots, \gamma_n(\boldsymbol{x})) \tag{5}$$

is the diagonal matrix of real eigenvalues $\gamma_i > 0$, $i = 1, \ldots, n$.

We can associate to $\mathbf{M}(\boldsymbol{x})$ a transformation matrix $\mathbf{T}(\boldsymbol{x})$ defined as

$$\mathbf{T}(\boldsymbol{x}) = \sqrt{\mathbf{M}(\boldsymbol{x})} = \mathbf{R}(\boldsymbol{x})\sqrt{\boldsymbol{\Lambda}(\boldsymbol{x})}\mathbf{R}(\boldsymbol{x})^{\mathrm{T}} \tag{6}$$

and we therefore can express $\mathbf{M}(\boldsymbol{x})$ in terms of $\mathbf{T}(\boldsymbol{x})$ as

$$\mathbf{M}(\boldsymbol{x}) = \mathbf{T}(\boldsymbol{x})^{\mathrm{T}}\mathbf{T}(\boldsymbol{x}) = \mathbf{T}(\boldsymbol{x})\mathbf{T}(\boldsymbol{x})^{\mathrm{T}} \tag{7}$$

The transformation matrix $\mathbf{T}$ maps an edge $\boldsymbol{v}$ of unit length with respect to $\mathbf{M}$ into an edge $\boldsymbol{w} = \mathbf{T}\boldsymbol{v}$ of unit length in the usual Euclidean norm. We have in fact that

$$\|\boldsymbol{w}\| = \sqrt{\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}} = \sqrt{\boldsymbol{v}^{\mathrm{T}}\mathbf{T}^{\mathrm{T}}\mathbf{T}\boldsymbol{v}} = \sqrt{\boldsymbol{v}^{\mathrm{T}}\mathbf{M}\boldsymbol{v}} = \|\boldsymbol{v}\|_{\mathbf{M}}$$

3. GEOMETRICAL CHARACTERISTICS OF AN ELEMENT

The geometrical characteristics of an element can be defined in terms of a set of n orthogonal directions $\boldsymbol{\alpha}_i$ and n associated element sizes δ_i, where n is the number of dimensions (Figure 1). The transformation matrix $\mathbf{T}$ can then be regarded as the result of combining n scaling operations, with factor $1/\delta_i$, in each direction $\boldsymbol{\alpha}_i$, i.e.

$$\mathbf{T} = \sum_{i=1}^{n} \frac{1}{\delta_i}\boldsymbol{\alpha}_i\boldsymbol{\alpha}_i^{\mathrm{T}} \tag{8}$$

By rewriting Equation (6) as

$$\mathbf{T} = \sum_{i=1}^{n} \sqrt{\gamma_i}\boldsymbol{r}_i\boldsymbol{r}_i^{\mathrm{T}} \tag{9}$$

and by comparing with Equation (8) we therefore have that

$$\delta_i = \frac{1}{\sqrt{\gamma_i}}, \quad \boldsymbol{\alpha}_i = \boldsymbol{r}_i \tag{10}$$

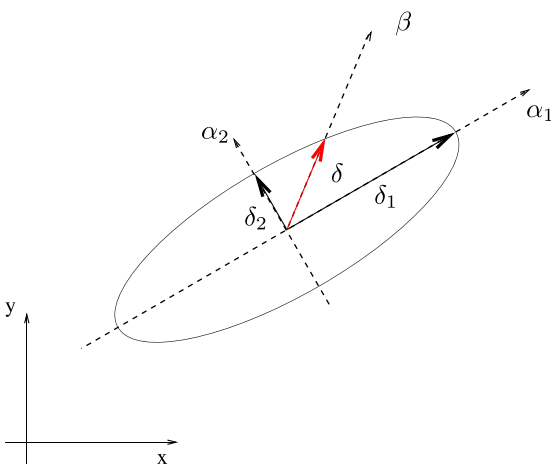

Figure 1. Spacing and directions defined by transformation matrix **T**.

The spacing $s(\boldsymbol{\beta})$ in the generic direction defined by the unit vector $\boldsymbol{\beta}$ (Figure 1) is given by

$$s(\boldsymbol{\beta}) = \|\mathbf{T}\boldsymbol{\beta}\|^{-1}$$

4. MESHING METHOD

The generation problem consists in subdividing an arbitrary complex domain into a consistent assembly of elements. The consistency of the mesh is guaranteed if the generated elements cover the entire domain and the intersection between elements occurs only on common points, sides, or triangular faces in the 3D case. The final mesh is built following a bottom-up logic. We briefly describe here the advancing front technique [1, 4], but other techniques, like those based on Delaunay triangulation [5, 6], may be used as an alternative for the unstructured mesh generation.

The process starts by discretizing each boundary curve as a set of straight-line segments. The length of these segments must be unitary with respect to the Riemannian structure.

The next step consists in generating triangular planar faces on surfaces, achieved using an indirect approach. The grid is generated in the parametric space and then mapped onto the real surface. At each iteration a front, which is a dynamic data structure containing the set of all sides currently available to form a triangular face, is defined. Initially, the front contains the edges, which discretize the boundary curves. When forming a new triangle, a front element is connected to a new node or to an existing node. After the triangle has been generated, the front is updated and the generation process proceeds until the front is empty. The size and shape of the generated triangles must be consistent with the information induced in the parametric space by the surface metric field. Finally, the mesh is mapped onto the real surface, defining the triangles which will become faces of the tetrahedra to be generated later.

The last step consists in discretizing the volume, still using the advancing front procedure. The front is now made up by the triangular faces which are available to form a tetrahedron.

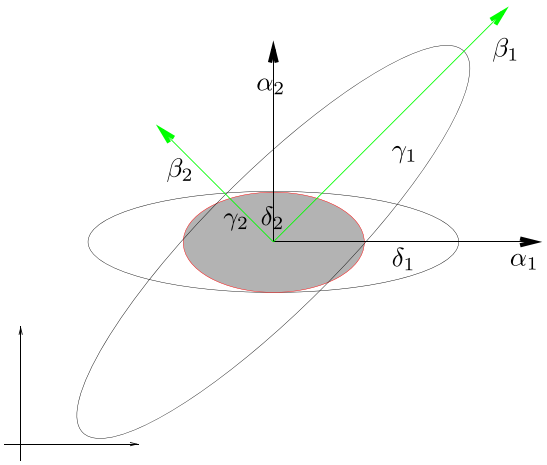

Figure 2. Metrics intersection.

The initial front is obtained by assembling the triangulations of the boundary surfaces. Nodes and elements will be simultaneously created. When forming a new tetrahedron, the three nodes belonging to a front element are connected either to an existing node or to a new node. After a tetrahedron has been generated, the front is updated. The generation procedure is completed when the front is empty.

5. MESH SPACING CONTROL ON THE BOUNDARY

The objective of surface mesh generation is to construct a triangulation in which the maximum distance from the real surface is below a prescribed threshold value—this is a useful and practical way to arrive at meshes which approximate the real surface as accurately as required. In practice, we have however to construct a metric field $\mathbf{M}_3$ on the real surface Σ so that the above criterion is satisfied by unitary (in $\mathbf{M}_3$) mesh edges, and a corresponding metric $\mathbf{M}_2$ in the parametric space Ω which will be used as input data by the surface mesh generator. The manner in which $\mathbf{M}_3$ and $\mathbf{M}_2$ are constructed is described in the following.

5.1. Mesh spacing based on local curvature

Let Σ be a surface defined by the parametrization $\boldsymbol{r}(u,v) \in \mathbb{R}^3$, P a point of Σ of coordinates $\boldsymbol{r}_P(u_P, v_P)$, $\boldsymbol{n}$ the normal to Σ at P, and $\boldsymbol{t}$ the unit tangent vector along some curves on Σ passing at P. By locally approximating in the neighbourhood of P the curve by its osculating circle, the ratio ε between δ, the maximal gap between the osculating circle and its chord of arc, and the osculating circle radius ρ, is given by

$$\varepsilon \equiv \frac{\delta}{\rho} = 1 - \sqrt{1 - (\lambda/2)^2} \tag{11}$$

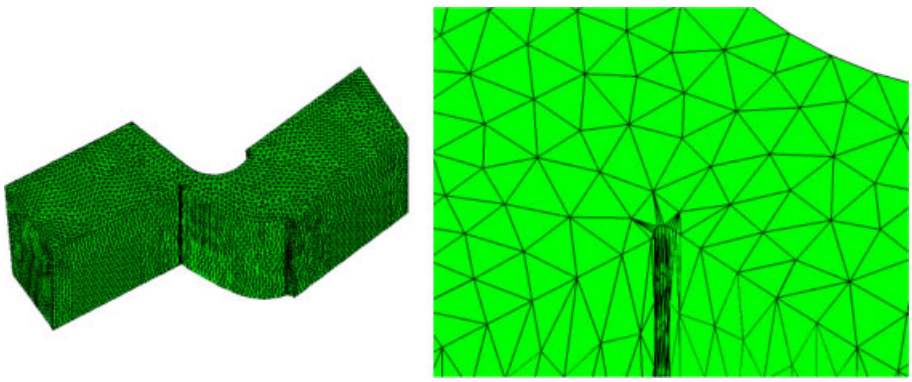

Figure 3. Effects of the discontinuity in the metric field of a stator blade.

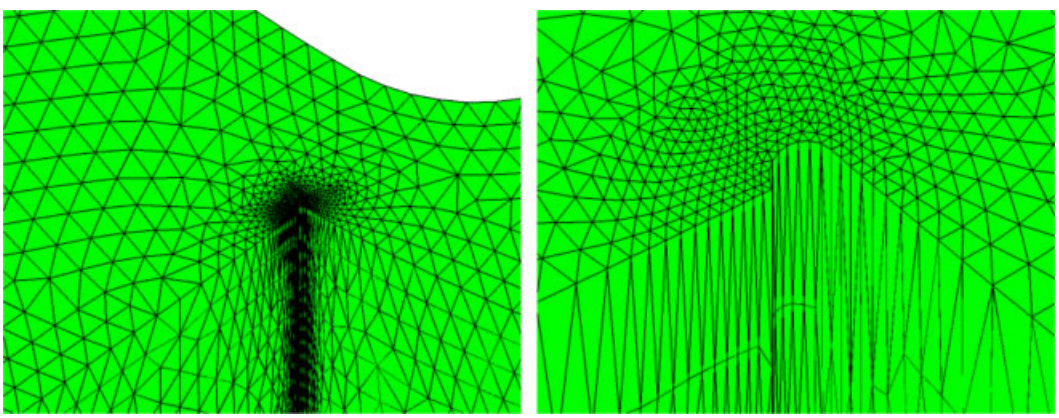

Figure 4. Metric smoothing effect at the intersection zone stator/hub.

where $\lambda = h_t/\rho$, and $h_t = \lambda\rho$ is the chord length. The parameter λ is therefore given as a function of ε according to

$$\lambda \equiv \frac{h_t}{\rho} = 2\sqrt{\varepsilon(2-\varepsilon)} \tag{12}$$

The above equation is employed to compute the chord length h_t for a prescribed value of ε—a parameter which measures how closely the chord approximates the real curve.

The geometrical characteristics of a surface element may be obtained by applying the above analysis in the surface principal directions, see, e.g. Reference [7], i.e. the directions in which the surface curvature assumes its minimum and maximum values. The two principal directions,

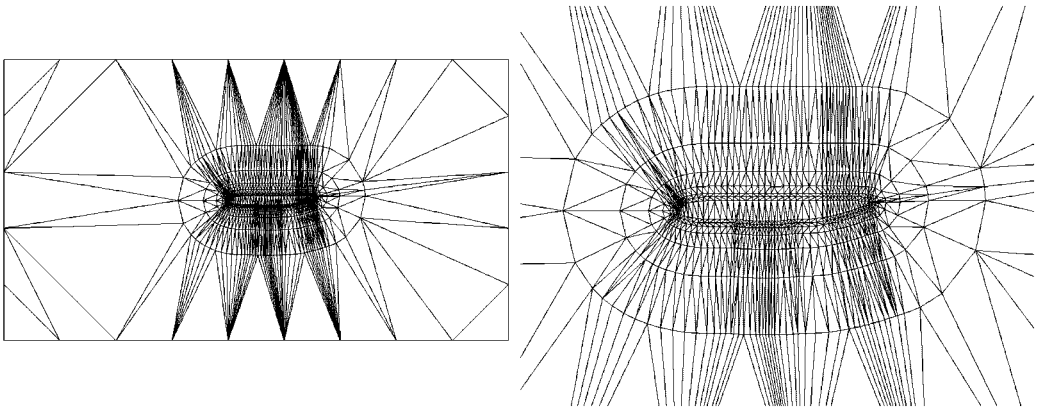

Figure 5. Construction of the symmetry plane background grid of an aeronautical configuration wing–fuselage.

$\boldsymbol{d}_i \in \mathbb{R}^3$, can be defined as

$$\boldsymbol{d}_i = \frac{\mathbf{J}(\boldsymbol{r})\dot{\boldsymbol{v}}_i}{\sqrt{\dot{\boldsymbol{v}}_i^{\mathrm{T}}\mathbf{G}\dot{\boldsymbol{v}}_i}}, \quad i = 1, 2 \tag{13}$$

where $\mathbf{J}(\boldsymbol{r})$ is the Jacobian matrix of the mapping $\boldsymbol{r} = \boldsymbol{r}(u, v)$, i.e.

$$\mathbf{J}(\boldsymbol{r}) = \left[\frac{\partial \boldsymbol{r}}{\partial u}, \frac{\partial \boldsymbol{r}}{\partial v}\right] \tag{14}$$

$\mathbf{G} = \mathbf{J}^{\mathrm{T}}\mathbf{J}$ is the first fundamental matrix of the surface Σ, and $\boldsymbol{v}_i \in \mathbb{R}^2$ the principal directions in the parametric space $u - v$. Note that, by construction, the directions $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are orthogonal. The element sizes h_i associated to the directions $\boldsymbol{d}_i$ are, then, computed as

$$h_i = 2\rho_i\sqrt{\varepsilon(2-\varepsilon)} = \frac{\sqrt{\varepsilon(2-\varepsilon)}}{\kappa_i} \tag{15}$$

where ε is a user prescribed parameter (the effect of ε on the mesh spacing is shown in Figure 1), ρ_i the principal radii of curvature and κ_i the principal curvature.

Finally, the metric tensor $\mathbf{M}_3(P)$ associated to the surface is computed according to

$$\mathbf{M}_3(P) = \sum_{i=1}^{3} \frac{1}{h_i^2}\boldsymbol{d}_i\boldsymbol{d}_i^{\mathrm{T}} \tag{16}$$

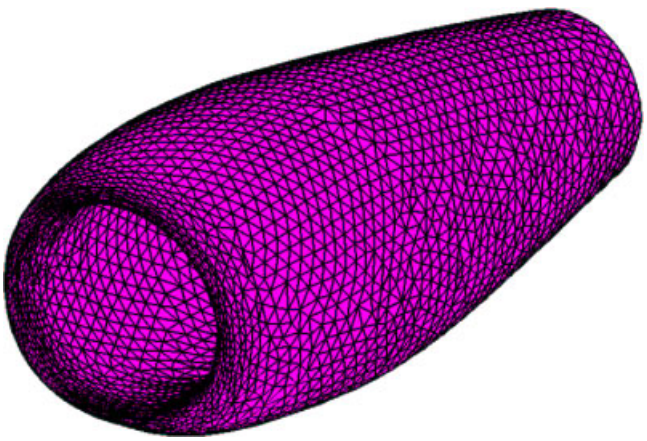

Figure 6. Grid around a nacelle.

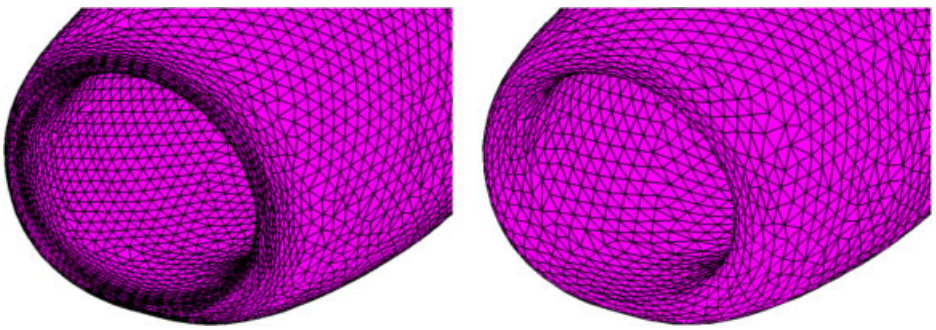

Figure 7. Grids around a nacelle for different values of ε.

where $\boldsymbol{d}_3 = \boldsymbol{n}$ and $h_3 = \min(h_1, h_2)$. The metric $\mathbf{M}_2(P)$ can be simply expressed as

$$\mathbf{M}_2(P) = \mathbf{J}(\boldsymbol{r})^{\mathrm{T}} \mathbf{M}_3(P) \mathbf{J}(\boldsymbol{r}) \tag{17}$$

where $\mathbf{J}$ is defined in Equation (14).

5.2. *Treatment of 'nearly flat' surfaces*

The values of h_i have to be occasionally limited, in order to avoid too large or too small elements. If h_1 and h_2 denote the smallest and the largest spacings, respectively, the limited spacings $\tilde{h}_i$ are defined as

$$\begin{aligned} \tilde{h}_1 &= \min(\max(h_1, h_{1,m}), h_{1,M}) \\ \tilde{h}_2 &= \min(\max(h_2, h_{2,m}), h_{2,M}, s_M h_1) \end{aligned} \tag{18}$$

Table I. Nacelle surface mesh statistics for different values of ε.

Test	ε	N_P	N_T
Nacelle	0.00125	8417	16830
Nacelle	0.0317	5358	10712

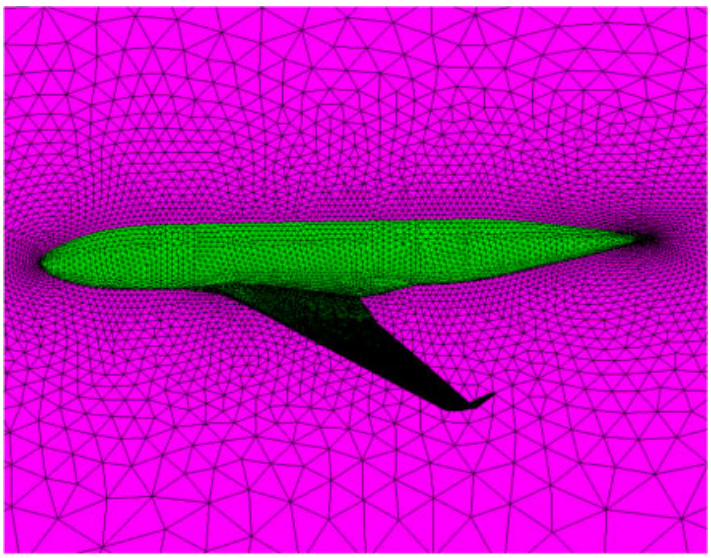

Figure 8. Wing–fuselage aeronautical configuration.

where $h_{i,m}$ and $h_{i,M}$ are the minimum and maximum value that h_i can assume. The parameter s_M represents the maximum value of element stretching. By choosing $s_M = 1$, an isotropic element will be formed.

The quantities $h_{i,m}$ and $h_{i,M}$ are defined as follows. Let us denote by l_u and l_v, the length of the curves represented parametrically by equations $\boldsymbol{u} = [u_P, v(s)]^{\mathrm{T}}$ and $\boldsymbol{u} = [u(s), v_P]^{\mathrm{T}}$, respectively. The spacing h_u and h_v are computed as a fractional part of the lengths l_u and l_v, i.e.

$$h_u = \frac{l_u}{n_u}, \quad h_v = \frac{l_v}{n_v} \tag{19}$$

where the parameters n_u and n_v ($n_u, n_v \geqslant 1$) are constants (integer) defined on Σ. The quantities $h_{i,M}$ are computed by using the Euler theorem

$$\begin{pmatrix} \dfrac{1}{h_u^2} \\ \dfrac{1}{h_v^2} \end{pmatrix} = \sum_{i=1}^{2} \frac{1}{h_{i,M}^2} \cos^2 \begin{pmatrix} \varphi_{i,u} \\ \varphi_{i,v} \end{pmatrix} \tag{20}$$

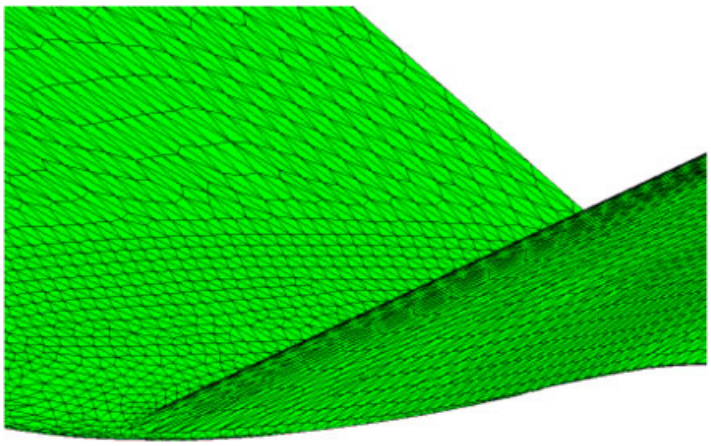

Figure 9. Detail of the wing/tip zone.

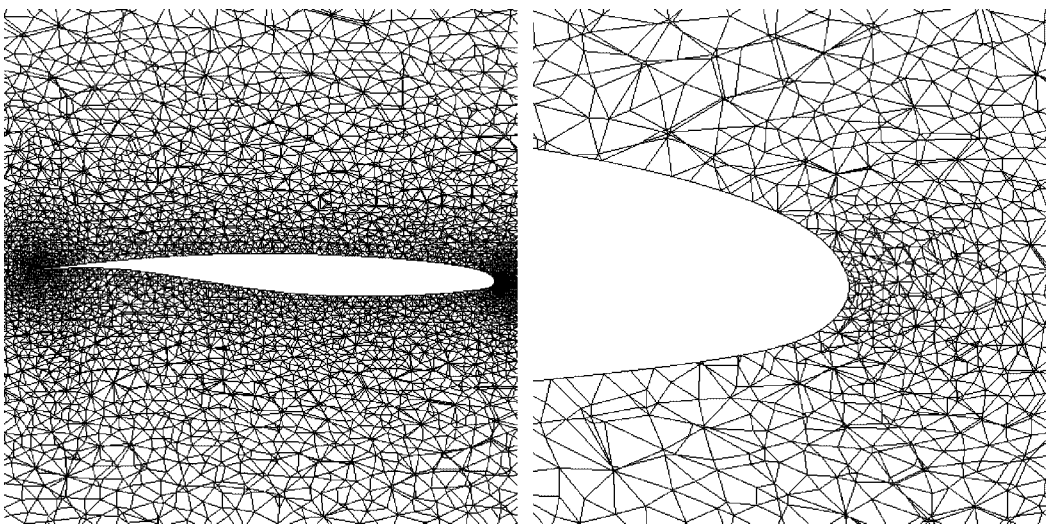

Figure 10. Wing section near the fuselage (wind coming from the right).

where $\varphi_{i,u}$ and $\varphi_{i,v}$ are the angles formed by the principal direction $\boldsymbol{d}_i$ with the tangent vectors to the curves $\mathbf{u} = [u_P, v(t)]^{\mathrm{T}}$ and $\mathbf{u} = [u(t), v_P]^{\mathrm{T}}$, denoted by $\boldsymbol{t}_u$ and $\boldsymbol{t}_v$, respectively. In addition, $h_{i,m}$ are defined as a fraction of $h_{i,M}$ or by taking a constant value on the surface.

When the geometric model is too complex, the computation of $h_{i,M}$ is performed by using the concept of background surface spacing characteristics, which are built by the knowledge of the characteristic dimensions of the object to be discretized and of the tangent vectors to the surface.

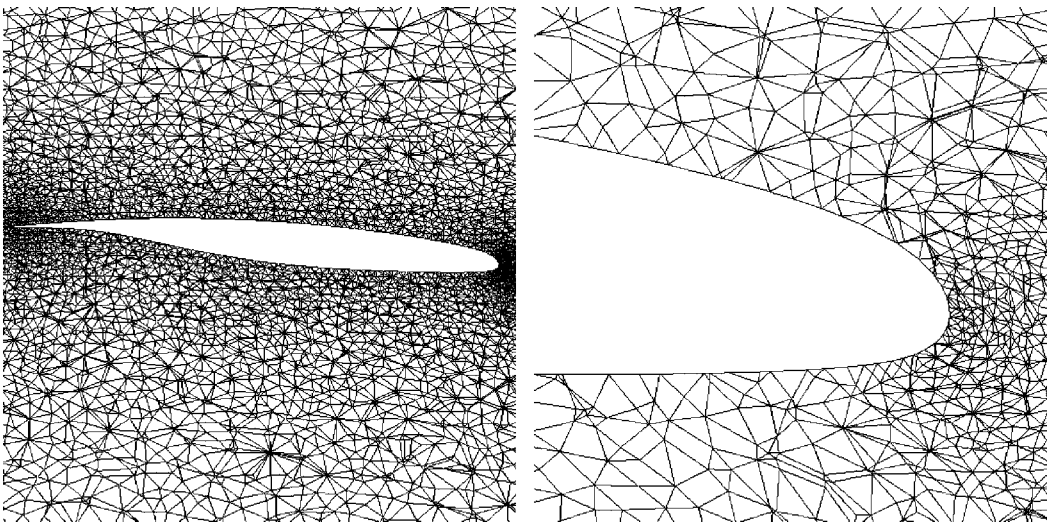

Figure 11. Wing section near the tip (wind coming from the right).

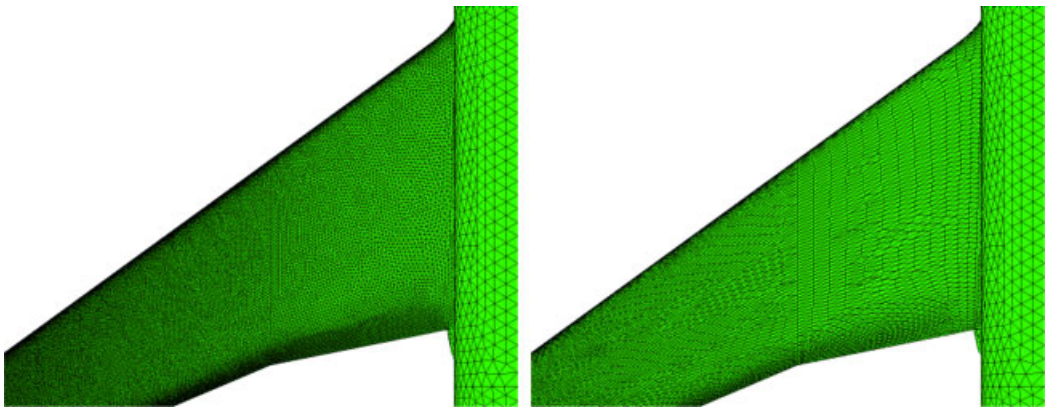

Figure 12. The isotropic (left) and anisotropic (right) grid on the wing.

When the surface is a plane ($\rho_i = \infty$) or when the principal curvatures are equal, the principal directions of normal curvature are undefined. However, the directions $\boldsymbol{t}_u$ and $\boldsymbol{t}_v$ still exist and can be chosen as directions $\boldsymbol{d}_i$, if $\boldsymbol{t}_u \cdot \boldsymbol{t}_v = 0$. In this case, the spacings $\tilde{h}_i$ associated to the directions $\boldsymbol{d}_i$ are, respectively, h_u and h_v, if $\boldsymbol{t}_u \cdot \boldsymbol{t}_v \neq 0$, $\boldsymbol{d}_1$ is chosen coincident with the direction which specifies the minimum spacing and $\boldsymbol{d}_2 = \boldsymbol{d}_1 \times \boldsymbol{n}$. In this case, $\tilde{h}_1 = \min(h_u, h_v)$ and $\tilde{h}_2$ is computed by using Equation (20).

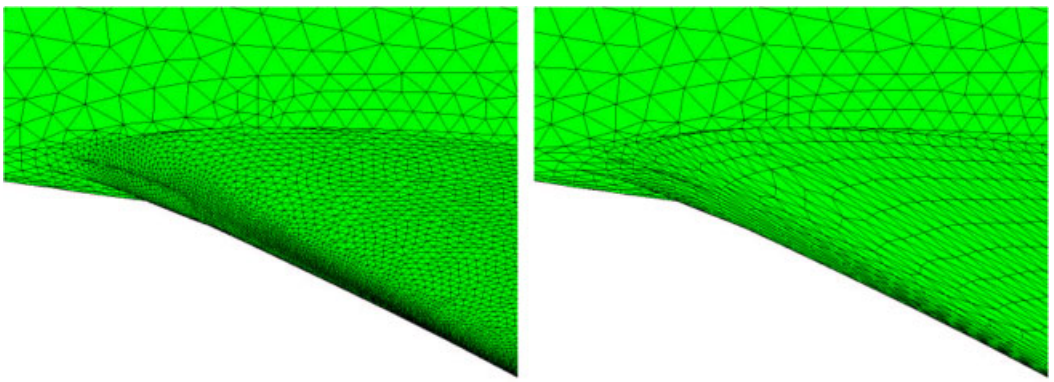

Figure 13. Detail of the isotropic (left) and anisotropic (right) grid at the wing/fuselage intersection zone.

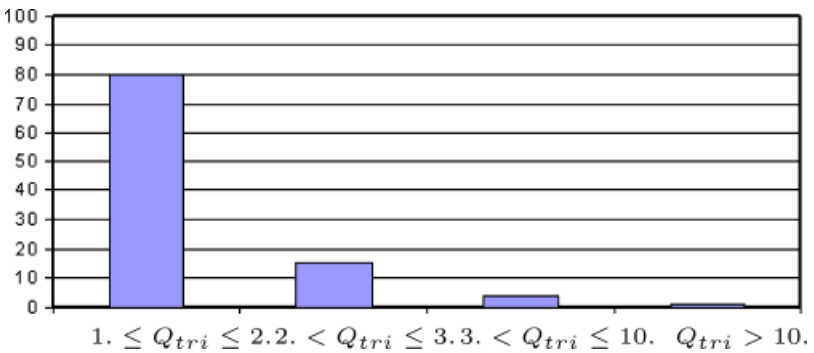

Figure 14. Surface elements quality distribution.

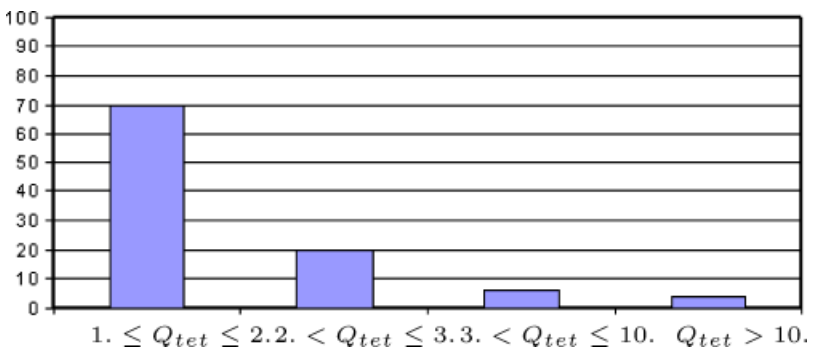

Figure 15. Tetrahedra quality distribution.

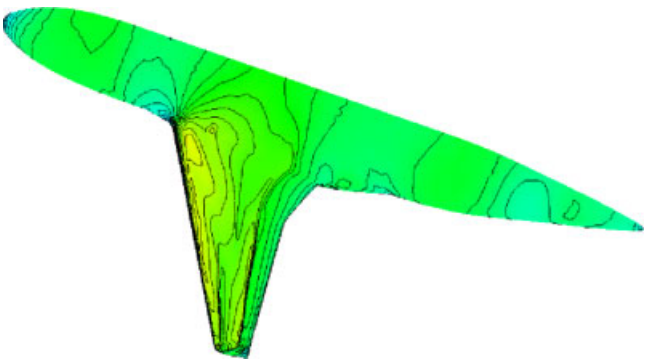

Figure 16. Pressure field on the aircraft.

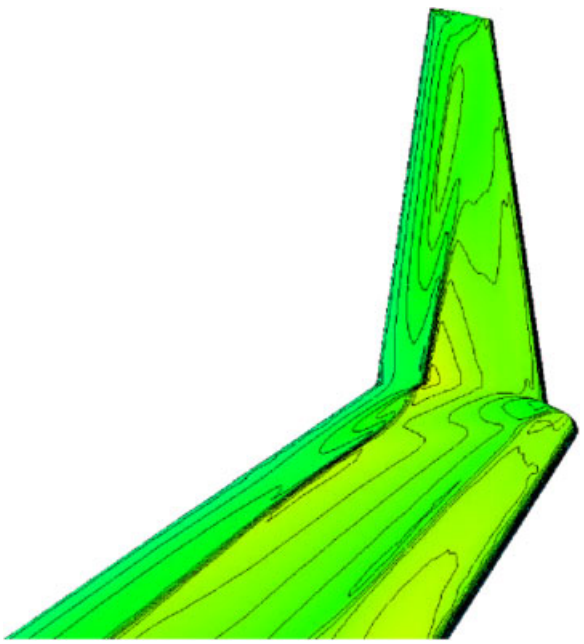

Figure 17. Pressure field on the winglet.

5.3. Metric intersection

A prerequisite for the grid-generation algorithm is the specification of only one metric at each point. However, the metrics are multi-defined on the boundary curves and at each corner. Thus, in these points, a metric intersection is requested.

Figure 18. Complex aeronautical geometry.

Let P be a point in the space at which two different metrics exist, say $\mathbf{M}_a(P)$ and $\mathbf{M}_b(P)$, defined as

$$\mathbf{M}_a(P) = \sum_{i=1}^{3} \frac{1}{\delta_i^2} \boldsymbol{\alpha}_i \boldsymbol{\alpha}_i^{\mathrm{T}}, \quad \mathbf{M}_b(P) = \sum_{i=1}^{3} \frac{1}{\gamma_i^2} \boldsymbol{\beta}_i \boldsymbol{\beta}_i^{\mathrm{T}} \tag{21}$$

The directions $\boldsymbol{\xi}_i$ of the intersection metric are chosen coincident with the directions associated to the metric matrix which specifies the minimum spacing, i.e. $\boldsymbol{\xi}_i = \boldsymbol{\alpha}_i$. The new metric $\mathbf{M}_{\mathrm{new}}(P)$ is defined (Figure 2) as

$$\mathbf{M}_{\mathrm{new}}(P) = \sum_{i=1}^{3} \frac{1}{h_i^2} \boldsymbol{\xi}_i \boldsymbol{\xi}_i^{\mathrm{T}} \tag{22}$$

where

$$h_i = \min\left(\delta_i, \frac{1}{\|\mathbf{T}_b \boldsymbol{\xi}_i\|^{-1}}\right) \tag{23}$$

5.4. Metric smoothing

Due to the fact that the metrics on the boundary curves and at each corner are, in general, multi-defined, discontinuity or large gradient may appear in the metric field, which can lead to the generation of bad quality elements. In Figure 3 we can see this effect at the intersection zone between the stator blade and the hub of a turbine.

A spring-analogy-based algorithm [8] is used to propagate and regularize the metric field. It is first assumed that each node i is connected to each adjacent node j by a fictitious spring under the generalized force $\mathbf{F}_{ij}$ defined by

$$\mathbf{F}_{ij} = K_{ij}(\mathbf{T}_j^{-1} - \mathbf{T}_i^{-1}) \tag{24}$$

where the scalar K_{ij} is the spring constant, $\mathbf{T}_i$ and $\mathbf{T}_j$ are the transformation matrices associated to nodes i and j. The spring constant can be expressed as

$$K_{ij} = K_{ij}^{\mathrm{sm}} K_{ij}^{\mathrm{lg}} \tag{25}$$

where

$$K_{ij}^{\mathrm{sm}} = \frac{A_{e1}^{ij} + A_{e2}^{ij}}{A_i} \tag{26}$$

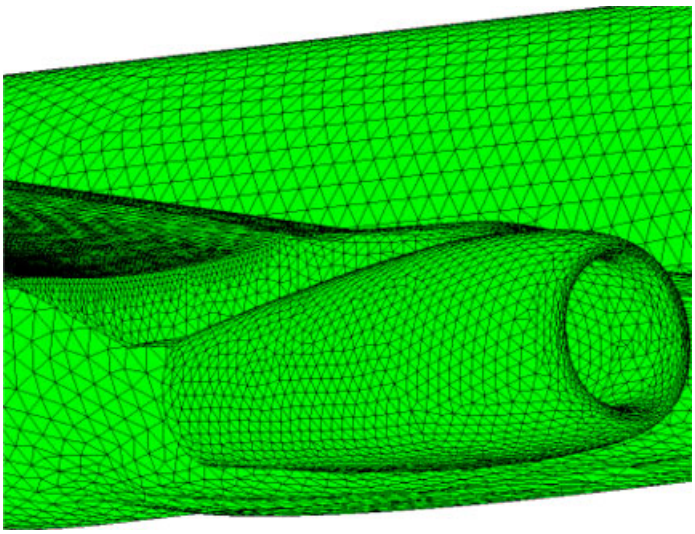

Figure 19. Detail of the wing/pylon/nacelle discretization.

is the spring constant associated to a Laplacian smoothing algorithm and

$$K_{ij}^{\mathrm{lg}} = \frac{1}{\alpha}\sqrt{\frac{(\boldsymbol{x}_j - \boldsymbol{x}_i)^{\mathrm{T}}\mathbf{M}_{ij}(\boldsymbol{x}_j - \boldsymbol{x}_i)}{(\boldsymbol{x}_j - \boldsymbol{x}_i)^{\mathrm{T}}(\boldsymbol{x}_j - \boldsymbol{x}_i)}} \tag{27}$$

is related to the locally required grid spacing. A_{e1}^{ij} and A_{e2}^{ij} are the areas of the two elements adjacent to side ij, A_i is the area of the elements surrounding node i, α a scaling factor, which is taken equal to the maximum value of K_{ij}^{lg} (with $\alpha = 1$) over the patch of elements surrounding node i, $\mathbf{M}_{ij}$ the metric matrix along side ij.

The resulting metric field is the solution of the equilibrium system for each vertex i:

$$\sum_{j \in \mathscr{K}_i} \mathbf{F}_{ij} = 0 \tag{28}$$

where $\mathscr{K}_i$ is the set of nodes surrounding i. In this work, a few iterations of the following pointwise scheme is actually used to smooth the metric field:

$$\begin{aligned} \mathbf{T}_{i,(p)}^{-1} &= \mathbf{T}_{i,(p-1)}^{-1} + \sum_{j \in \mathscr{K}_i} \mathbf{F}_{ij,(p-1)} \\ &= \mathbf{T}_{i,(p-1)}^{-1} + \sum_{j \in \mathscr{K}_i} K_{ij}(\mathbf{T}_{j,(p-1)}^{-1} - \mathbf{T}_{i,(p-1)}^{-1}) \end{aligned} \tag{29}$$

In Figure 4 the effect of the metric smoothing algorithm is shown on the geometry presented above.

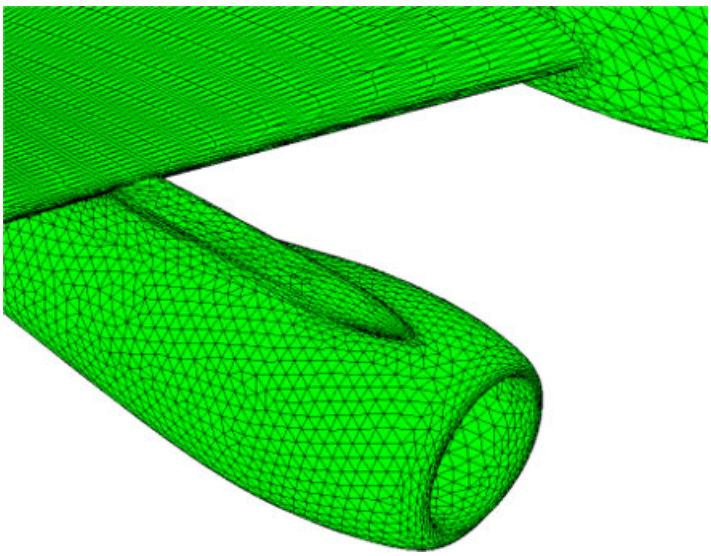

Figure 20. Detail of the wing/pylon intersection zone.

6. MESH SPACING CONTROL IN THE VOLUME

The mesh spacing in the volume is controlled through a background grid, which is used to interpolate the boundary value of the metric field at any other point of the domain to be discretized. The procedure in order to build the background grid is described in the following.

The process starts constructing the Delaunay triangulation [9, 10] of the points, which discretize the skin surface and the external boundaries. The circumcentres of the tetrahedra that have at least one node on the skin and that are not inside the body are computed. An 'average' direction $\bar{\boldsymbol{v}}_{il}$ is obtained by solving the minimization problem

$$\min \sum_{j=1}^{n_j} \left(1 - \bar{\boldsymbol{v}}_{il} \cdot \frac{\boldsymbol{v}_{ij}}{\|\boldsymbol{v}_{ij}\|}\right)^2 \tag{30}$$

where $\boldsymbol{v}_{ij}$ are the vectors that connect the node i on the skin with the set of circumcentres j associated to the elements which have node i as vertex, and n_j is the number of circumcentres belonging to the set. Then, the vector $\boldsymbol{v}_{il}$ is constructed connecting the node i with the local node l which is located at the intersection of the direction $\bar{\boldsymbol{v}}_{il}$ with the surface formed by the n_j circumcentres. The procedure is repeated by taking the new layer of nodes l as external boundary until a maximum of iterations is reached. This results in the construction of a succession of nodes layers which become closer and closer to the skin surface (Figure 5).

The background grid is defined by building the Delaunay triangulation of the nodes on the skin, on the external boundary and on additional layers. The transformation matrix $\mathbf{T}_l$ at the

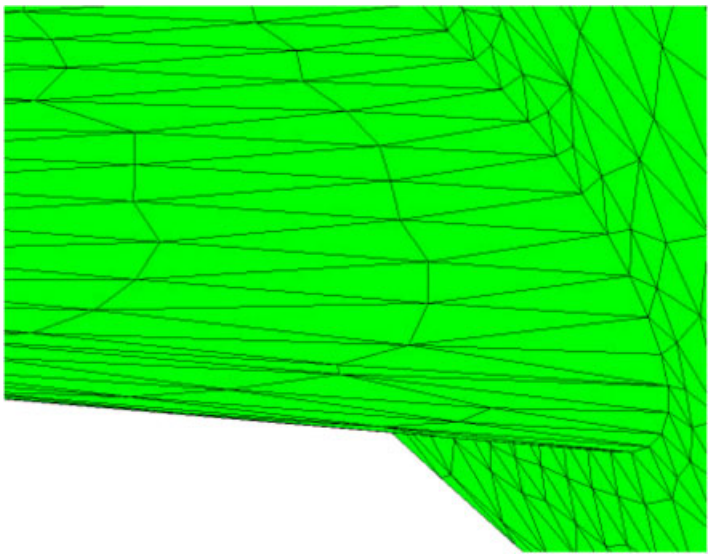

Figure 21. Detail of the grid at the wing root.

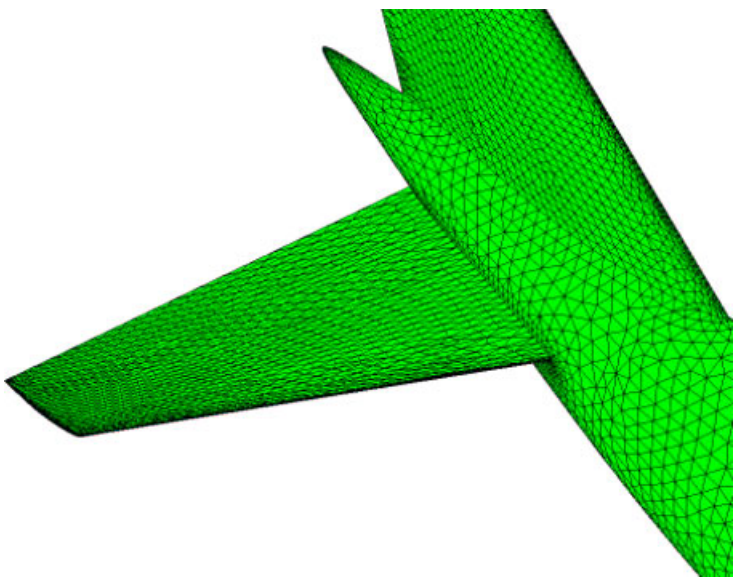

Figure 22. Rear fuselage view.

node l is computed as

$$\mathbf{T}_l = ((1-\omega)\mathbf{T}_i^{-1} + \omega\mathbf{T}_\infty^{-1})^{-1}, \quad \omega \in [0,1] \tag{31}$$

where $\mathbf{T}_\infty$ is the transformation matrix based on the element characteristics at the external boundary, $\mathbf{T}_i$ the transformation matrix associated to the skin of nodes i and ω a parameter defined as

$$\omega = f(\|\boldsymbol{v}_{il}\|, r_\infty, b) = (1-b)\left(\frac{\|\boldsymbol{v}_{il}\|}{r_\infty}\right)^2 + b\,\frac{\|\boldsymbol{v}_{il}\|}{r_\infty} \tag{32}$$

where r_∞ is the distance from the skin to the external boundary and the coefficient b the slope of the function at its origin.

7. APPLICATION EXAMPLES

We will now present the results obtained on three geometries: a nacelle, a wing–fuselage aeronautical configuration with a large winglet and a complete aircraft.

In the first case (Figure 6), the mesh on a nacelle has been generated using the scheme introduced in Section 4. The surface spacing has been computed following the approach discussed in Section 5. A value of $h_m = 2$ mm, $h_M = 60$ mm and $s_M = 30$ have been used.

Figure 7 shows the meshes obtained by using two different values of the parameter ε and Table I reports their characteristics, i.e. the number of points N_P and the number of triangles N_T.

Figure 8 illustrates the mesh obtained for a wing–fuselage aeronautical configuration. The number of nodes on the body and in the 3D domain are 15 000 and 284 000, respectively, which corresponds to 1 562 000 tetrahedra. The maximum elements stretching is 50 and the anisotropic nature of the mesh can be appreciated in Figures 9–11, where wing/tip zone, wing sections near fuselage and tip zone are shown. The anisotropic approach can lead to a reduction of nearly one order of magnitude of nodes used with respect to the isotropic case, as shown in Figures 12 and 13.

The diagrams of Figures 14 and 15 show the elements shape quality distribution of surface triangles and of tetrahedra. The quality of an element K [11] (triangle/tetrahedron) is defined as

$$Q_K = \alpha\,\frac{h_{\max}}{\rho_K} \tag{33}$$

where $h_{\max}$ is the element diameter, i.e. its longest edge while ρ_K is the inradius of the element K. Note that this value varies between 1 and ∞, and that moreover the closer Q_K is to 1, the better the element K is.

This grid has been used to compute a transonic inviscid flow at $M_\infty = 0.85$ and the angle of attack $\alpha = 2^\circ$ and the correspondent pressure coefficient distribution around the aircraft is illustrated in Figures 16 and 17.

In the last example we present the mesh around a complex aeronautical geometry (Figure 18). The number of nodes on the body and in the 3D domain are 18 900 and 315 000, respectively, which corresponds to 1 923 000 tetrahedra. The maximum elements stretching is 50 and the anisotropic nature of the mesh can be appreciated in Figures 19–22.

8. CONCLUSIONS

Most of the available unstructured grid generators can only construct isotropic meshes. This is a serious limitation since the use of anisotropic meshes can result in significant savings when the flow field or the geometry of the domain are characterized by anisotropic features. This is indeed the case in many applications and is typical in CFD.

In this paper an original procedure to anisotropically prescribe the metric field on both the surface and the volume of the computational domain is presented. The spacing on the boundary mesh is computed by the analysis of the principal curvatures and directions of the surface, and the spacing in the domain is obtained by propagating the spacing at the boundaries by means of a background grid. The special treatments required to deal with nearly flat surface and to regularize the spacings are thoroughly discussed in the paper.

The performance of the proposed approach is demonstrated by presenting the results obtained in the anisotropic 3D grid generation of complex aeronautical and turbomachinery configurations. The test computations here presented seem to indicate that a reduction of nearly one order of magnitude of nodes with respect to the isotropic case can be obtained by the proposed methodology.

The extension of the code to the generation of hybrid grids for the solution of viscous problems is under development.

REFERENCES

1. Peraire J, Morgan K, Formaggia L, Peiro J, Zienkiewicz OC. Finite element Euler computations in three dimensions. *International Journal for Numerical Methods in Engineering* 1988; **26**:765–781.
2. Borouchaki H, Laug P, Hecht F, Saltel E, George PL. Delaunay mesh generation governed by metric specification. Part I: algorithm. *Finite Elements in Analysis and Design* 1997; **25**:61–83.
3. Borouchaki H, Laug P, George PL. Parametric surface meshing using a combined advancing-front Delaunay approach. *International Journal for Numerical Methods in Engineering* 2000; **49**:233–259.
4. Marcum DL, Weatherill NP. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal* 1995; **33**:1619–1625.
5. George PL. *Delaunay Triangulation and Meshing*. Hermes, 1998.
6. Baker TJ. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers* 1989; **5**:161–175.
7. Farin G. *Curves and Surfaces for Computational Aided Geometric Design. A Practical Guide*. Academic Press: New York, 1997.
8. Selmin V, Formaggia L. Mesh adaption on 2-D unstructured mesh, in computational mesh adaptation. In *Notes on Numerical Fluid Dynamics*, Hills DP *et al.* (eds), 1999; 187–204.
9. Bowyer A. Computing Dirichlet tessellation. *Computer Journal* 1981; **24**:162–166.
10. Watson DF. Computing the n-dimensional Delaunay tessellation with application to voronoi polytopes. *Computer Journal* 1981; **24**:167–172.
11. Frey P, George PL. *Mesh Generation. Application to Finite Elements.* Hermes, 2000.

