Finite Elements in Analysis and Design 46 (2010) 181 – 202

Contents lists available at ScienceDirect

Finite Elements in Analysis and Design

journal homepage: www.elsevier.com/locate/finel

Size gradation control of anisotropic meshes

F. Alauzet

INRIA, Projet Gamma, Domaine de Voluceau, Rocquencourt, BP 105, 78153 Le Chesnay Cedex, France

ARTICLE INFO

Available online 19 August 2009

Keywords:
Mesh gradation
Anisotropic mesh
Mesh adaptation
Metric
Unit mesh

ABSTRACT

This paper gives a status on anisotropic mesh gradation. We present two 3D anisotropic formulations of mesh gradation. The metric at each point defines a well-graded smooth continuous metric field over the domain. The mesh gradation then consists in taking into account at each point the strongest size constraint given by all these continuous metric fields. This is achieved by a metric intersection procedure. We apply it to several examples involving highly anisotropic meshes.

1. Introduction

In many engineering applications, it is desirable to generate anisotropic meshes presenting highly stretched elements in adequate directions. Numerous papers have been published on mesh adaptation for numerical simulations in computational solid or fluid mechanics. Among these papers, some have addressed the problem of creating 3D unstructured anisotropic meshes [5,6,10,12,15,17,20,21,23]. In these approaches, different mesh generation methods are considered. Nonetheless all of them are based on the notion of *unit mesh* in a Riemannian metric space.

In the context of numerical simulations based on finite element or finite volume methods, these works have proved the efficiency of unstructured mesh adaptation to improve the accuracy of the numerical solution as well as for capturing the behavior of physical phenomena. In principle, this technique enables to substantially reduce the number of degrees of freedom, thus impacting favorably (i) the cpu time, (ii) the data storing and (iii) the solution analysis (visualization). Moreover, it has been recently proved in [20] that mesh adaptation impacts positively the order of convergence of numerical scheme by computing the numerical solution with a coherent accuracy in the entire domain.

The size prescription in the generation of anisotropic meshes is achieved thanks to metric fields. However, such metric fields may have huge variations making the generation of a unit mesh difficult or impossible, thus leading to poor quality anisotropic meshes. Generating high-quality anisotropic meshes requires to smooth the metric field by bounding its variations in all directions. To this end, a mesh gradation control procedure was introduced in [4]. It consists in reducing the size prescribed at mesh vertices by checking the metric variations along the mesh edges. The authors first describe an isotropic formulation from which they deduce an extension to anisotropic meshes. They present an homothetic reduction dedicated for surface mesh generation and a non-homothetic reduction. In the context of volume meshing, the homothetic leads to inconsistency in the metric reduction during the size gradation procedure. An anisotropic mesh gradation has also been presented in [16]. This procedure considers spectral decomposition and associate eigenvectors together with ad hoc choices. We prefer a formulation that uses directly well-posed operations on metrics. In [22], an isotropic size gradation control has been applied to the generation of multi-patch parametric surface meshes.

In this paper, we give a status on anisotropic mesh gradation. We present two 3D anisotropic formulations of mesh gradation extending the formulation given in [4] and we apply it to several examples involving highly anisotropic meshes. We formulate the problem mathematically by employing the continuous modeling of a mesh proposed in [19]. The metric at each point defines a well-graded smooth continuous metric field over the domain. The mesh gradation then consists in taking into account at each point the strongest size constraint given by all these continuous metric fields. Numerically, in the context of a mesh with a metric field given at its vertices, the idea consists in imposing at each vertex a size constraint related to all the other vertices of the mesh. To this end, all vertices of the mesh span metric fields in the whole domain by growing their metrics at a rate given by the desired gradation coefficient. Then, the reduced metric at a vertex is the intersection between its metric and all these metrics. Unfortunately, this mesh gradation algorithm is intrinsically of quadratic complexity. We thus establish an approximation to solve it in a linear time.

This paper is outlined as follows. In Section 2 the notion of a metric and the metric-based method to generate anisotropic meshes are described. In Section 3 the isotropic mesh gradation is recalled.

E-mail address: Frederic.Alauzet@inria.fr.

0168-874X/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.finel.2009.06.028

Then, two anisotropic formulations of the mesh gradation are presented, Section 4. In the numerical examples, Section 5, we illustrate the efficiency of the presented method for anisotropic mesh adaptation. We also exemplify that anisotropic mesh gradation can be used for other applications such as the generation of well-graded meshes. Finally, the limits of the proposed approaches are exemplified on an analytical example in Section 6. New mesh gradation procedures improving the previous ones are then proposed to remedy these problems.

2. Metric and mesh generation

In this section, we recall a metric-based method to generate anisotropic meshes. It is based on the notion of Riemannian metric space and on the concept of *unit mesh* initially introduced in [14]. Well-posed operations on metrics are introduced and we discuss the numerical computation of the length of a given path in a metric space.

2.1. Metric notion

A *metric tensor* (or simply a *metric*) $\mathcal{M}$ in $\mathbb{R}^n$ is an $n \times n$ symmetric definite positive matrix. $\mathcal{M}$ is always diagonalizable and can be decomposed as $\mathcal{M} = {}^t\mathcal{R}\Lambda\mathcal{R}$, where $\mathcal{R}$ and Λ are the eigenvectors and the eigenvalues matrices of $\mathcal{M}$, respectively. From this definition, it follows up that the *scalar product* of two vectors in $\mathbb{R}^n$ can be defined related to a metric $\mathcal{M}$ as

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}} = \langle \mathbf{u}, \mathcal{M}\mathbf{v} \rangle = {}^t\mathbf{u}\mathcal{M}\mathbf{v} \in \mathbb{R},$$

where the natural dot product of $\mathbb{R}^n$ has been denoted by $\langle .,. \rangle$. Under this notion, the *Euclidean norm* of a vector $\mathbf{u}$ in $\mathbb{R}^n$ according to $\mathcal{M}$ is defined as

$$\|\mathbf{u}\|_{\mathcal{M}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathcal{M}}} = \sqrt{{}^t\mathbf{u}\mathcal{M}\mathbf{u}}$$

that actually measures the length of vector $\mathbf{u}$ with respect to metric $\mathcal{M}$. A metric $\mathcal{M}$ could be geometrically represented by its associated unit ball, an ellipsoid, defined by

$$\mathcal{E}_{\mathcal{M}} = \left\{ \mathbf{p} \,\middle|\, \sqrt{{}^t\mathbf{op}\mathcal{M}\mathbf{op}} = 1 \right\},$$

where $\mathbf{o}$ is the center of the ellipsoid, see Fig. 1. The main axes are given by the eigenvectors of matrix $\mathcal{M}$ and the radius along each axis is given by the inverse of the square root of the associated eigenvalue.

Definition 2.1. An *Euclidean metric space* is a vector space supplied with a scalar product $\langle .,. \rangle_{\mathcal{M}}$ defined by metric tensor $\mathcal{M}$. We denote it $(\mathbb{R}^n, \mathcal{M})$. The *distance* between two points $\mathbf{p}$ and $\mathbf{q}$ is given by $d_{\mathcal{M}}(\mathbf{p}, \mathbf{q}) = \sqrt{{}^t\mathbf{pq}\mathcal{M}\mathbf{pq}}$. Finally, the *length* of a segment $\mathbf{pq}$ is the distance between its extremities: $\ell_{\mathcal{M}}(\mathbf{pq}) = d_{\mathcal{M}}(\mathbf{p}, \mathbf{q})$.

Remark 2.1. If the metric defining the scalar product is the identity matrix, $\mathcal{M} = I_n$, then we get the standard Euclidean space $(\mathbb{R}^n, I_n)$ supplied with the natural dot product: $\langle .,. \rangle$. We denote by $\| \cdot \|_2$ the natural Euclidean norm.

It is then possible to define volumes and angles in an Euclidean metric space. Let K be a bounded subset of $\mathbb{R}^n$, the *volume* of K in metric $\mathcal{M}$ is

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det(\mathcal{M})}\, dK = \sqrt{\det(\mathcal{M})}|K|_{I_n}.$$

The *angle* between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined by the unique real $\theta \in [0, \pi]$ such that

$$\cos(\theta) = \frac{\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}}}{\|\mathbf{u}\|_{\mathcal{M}} \|\mathbf{v}\|_{\mathcal{M}}}.$$

We now consider the more general case where the metric, and thus the scalar product, vary all over the domain.

Definition 2.2. A *Riemannian metric space* is a continuous manifold $\Omega \subset \mathbb{R}^n$ supplied with a smooth metric $\mathcal{M}(\cdot)$. We denote it by $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. The restriction of the metric to a point $\mathbf{x}$ of the manifold defines a scalar product on the tangent space $T_{\mathbf{x}}\Omega$. The tangent space equipped with this structure is an Euclidean metric space.

Contrary to the Euclidean metric space case, the distance between two points, i.e., the shortest path, is no more the straight line but it is given by a geodesic. Nevertheless, in the context of mesh generation or mesh adaptation, we are not interested in the distance between two points but in the length of a path given by an edge of the mesh. More precisely, in a Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$, the *length* of an edge $\mathbf{pq}$ is calculated by using the straight line parametrization $\gamma(t) = \mathbf{p} + t\mathbf{pq}$, $t \in [0, 1]$:

$$\ell_{\mathcal{M}}(\mathbf{pq}) = \int_0^1 \|\gamma(t)\|\, dt = \int_0^1 \sqrt{{}^t\mathbf{pq}.\mathcal{M}(\mathbf{p} + t\mathbf{pq})\mathbf{pq}}\, dt. \tag{1}$$

Fig. 2 depicts iso-values of segment length from the origin, from the standard Euclidean space to a Riemannian metric space. The plotted function is $f(\mathbf{x}) = \ell_{\mathcal{M}}(\mathbf{ox})$ where $\mathbf{o}$ is the origin of the plane.

2.2. Anisotropic mesh generation

We briefly detail our approach to anisotropic mesh generation based on the notion of *unit mesh*. The generation of anisotropic meshes requires the specification of a mesh size in each direction at each point of the domain. To this end, we consider metric tensors to specify different sizes in different directions. Then, the idea is to work in an adequate Riemannian metric space specifying sizes

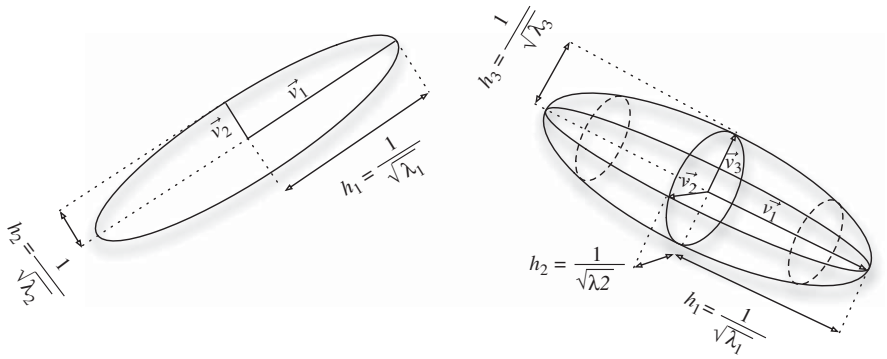


Fig. 1. Ellipse and ellipsoid representing a metric.

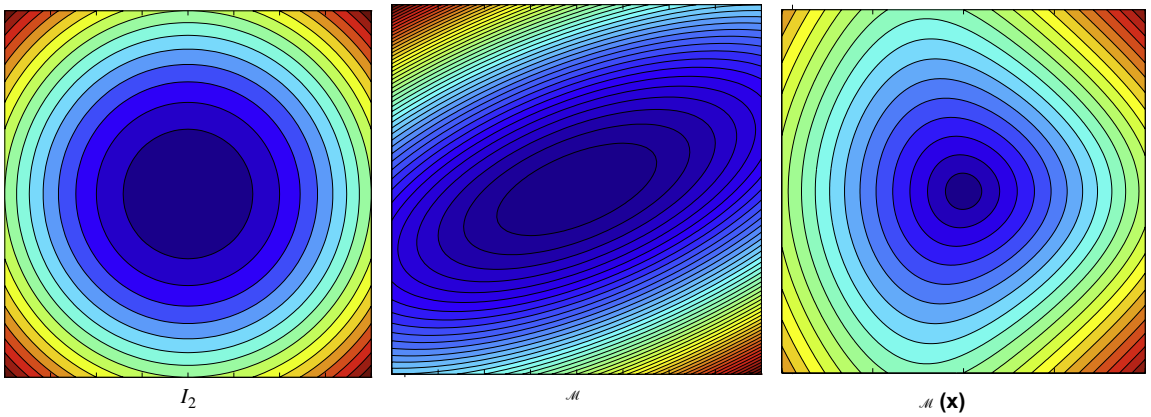


Fig. 2. Iso-values of the function $f(\mathbf{x}) = \ell_{\mathcal{M}}(\mathbf{ox})$ where $\mathbf{o}$ is the origin, i.e., segment length issued from the origin, for different Riemannian metric spaces. Left, the standard Euclidean space $([-1,1]\times[-1,1], I_2)$. Middle, an Euclidean metric space $([-1,1]\times[-1,1], \mathcal{M})$. Right, a Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in[-1,1]^2}$.

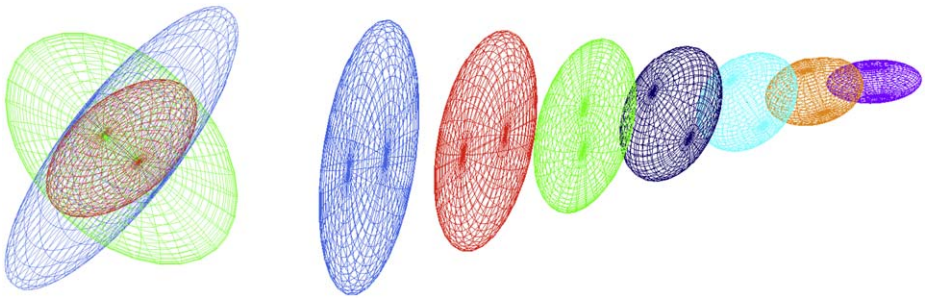


Fig. 3. Left, view illustrating the metric intersection procedure with the simultaneous reduction in three dimensions. In red, the resulting metric of the intersection of the blue and green metrics. Right, metric interpolation along a segment where the endpoints metrics are the blue and violet ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

directionally and to generate a uniform unit mesh in this metric space.

More precisely, let $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ be a Riemannian metric space. In three dimensions, the objective is to generate a mesh $\mathcal{H}$ for which each edge length is unit in the metric and each tetrahedron is regular:

$$\forall \mathbf{e} \in \mathcal{H},\ \ell_{\mathcal{M}}(\mathbf{e}) = 1 \quad \text{and} \quad \forall K \in \mathcal{H},\ |K|_{\mathcal{M}} = \frac{\sqrt{2}}{12}.$$

As the regular tetrahedron does not fill space, the mesh generator can only create a quasi-uniform mesh, i.e., the mesh elements are almost unit in the metric. To quantify the gap to unity, a quality function is used, for instance:

$$Q_{\mathcal{M}} = \frac{36}{3^{1/3}} \frac{|K|_{\mathcal{M}}^{2/3}}{\sum_{i=1}^{6} \ell_{\mathcal{M}}(\mathbf{e}_i)^2}.$$

We have seen that the size of the unit ball associated with a metric tensor $\mathcal{M}$ in the i-th eigenvector direction is given by $h_i = \lambda_i^{-1/2}$. We then deduce that the prescribed size in direction $\mathbf{e}$ is

$$h_{\mathcal{M}}(\mathbf{e}) = \frac{\|\mathbf{e}\|_2}{\ell_{\mathcal{M}}(\mathbf{e})}.$$

In the case of an Euclidean metric space, the previous relation simply reads

$$h_{\mathcal{M}}(\mathbf{e}) = \frac{\|\mathbf{e}\|_2}{\sqrt{{}^t\mathbf{e}\mathcal{M}\mathbf{e}}}. \tag{2}$$

2.3. Operations on metrics

The main advantage when working with metric spaces is the well-posedness of operations on metric tensors among which the metric intersection and the metric interpolation. These operations have a straightforward geometric interpretation when considering the ellipsoid associated with a metric.

2.3.1. Metric intersection

When several metrics are specified at a point of the domain, all these metric tensors must be reduced to a single one due to mesh generation concerns. The metric intersection consists in keeping the strongest size constraint in all directions given by this set of metrics.

Formally speaking, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two metric tensors given at a point. The metric tensor $\mathcal{M}_{1\cap 2}$ corresponding to the intersection of $\mathcal{M}_1$ and $\mathcal{M}_2$ must be such that the size prescription is the largest possible under the constraint that the size in each direction is always smaller than the sizes prescribed by $\mathcal{M}_1$ and $\mathcal{M}_2$. Geometrically speaking, it means that the ellipsoid representing $\mathcal{M}_{1\cap 2}$ is the largest ellipsoid included in the region defined by the geometric intersection of the ellipsoids corresponding to $\mathcal{M}_1$ and $\mathcal{M}_2$, cf. Fig. 3, left. The ellipsoid (metric) verifying this property is obtained by using the simultaneous reduction of the two metrics.

More precisely, the simultaneous reduction enables to find a basis (e_1, e_2, e_3) such that $\mathcal{M}_1$ and $\mathcal{M}_2$ are congruent to a diagonal matrix in this basis, and then to deduce the intersected metric. To do so, the matrix $\mathcal{N} = \mathcal{M}_1^{-1}\mathcal{M}_2$ is introduced. $\mathcal{N}$ is diagonalizable with real-eigenvalues and the sought basis is given by the normalized eigenvectors of $\mathcal{N}$ denoted by e_1, e_2 and e_3. The entries of the diagonal matrices, that are associated with the metrics $\mathcal{M}_1$ and $\mathcal{M}_2$ in this basis, are obtained with the Rayleigh formula:

$$\lambda_i = {}^t e_i \mathcal{M}_1 e_i \quad \text{and} \quad \mu_i = {}^t e_i \mathcal{M}_2 e_i \quad \text{for } i = 1, \ldots, 3.$$

Let $\mathscr{P}=(e_1e_2e_3)$ be the matrix the columns of which are the eigenvectors $\{e_i\}_{i=1,\dots,3}$ of $\mathscr{N}$. $\mathscr{P}$ is invertible as (e_1,e_2,e_3) is a basis of $\mathbb{R}^3$. Therefore, we have

$$\mathscr{M}_1 = {}^t\mathscr{P}^{-1}\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{pmatrix}\mathscr{P}^{-1}$$

and

$$\mathscr{M}_2 = {}^t\mathscr{P}^{-1}\begin{pmatrix}\mu_1 & 0 & 0\\ 0 & \mu_2 & 0\\ 0 & 0 & \mu_3\end{pmatrix}\mathscr{P}^{-1}.$$

The resulting intersected metric $\mathscr{M}_1\cap\mathscr{M}_2$ is analytically given by

$$\begin{aligned}\mathscr{M}_{1\cap2} &= \mathscr{M}_1\cap\mathscr{M}_2\\ &= {}^t\mathscr{P}^{-1}\begin{pmatrix}\max(\lambda_1,\mu_1) & 0 & 0\\ 0 & \max(\lambda_2,\mu_2) & 0\\ 0 & 0 & \max(\lambda_3,\mu_3)\end{pmatrix}\mathscr{P}^{-1}.\end{aligned}$$

The ellipsoid associated with $\mathscr{M}_{1\cap2}$ is the largest ellipsoid included in the geometric intersection region of the ellipsoids associated with $\mathscr{M}_1$ and $\mathscr{M}_2$, the proof is given in [1].

Numerically, to compute $\mathscr{M}_{1\cap2}$, the real-eigenvalues of $\mathscr{N}$ are first evaluated with a Newton algorithm. Then, the eigenvectors of $\mathscr{N}$, which define $\mathscr{P}$, are computed using the algebra notions of image and kernel spaces.

Remark 2.2. The intersection operation is not commutative. Consequently, when more than two metrics are intersected, the result depends on the order of intersection. In this case, the resulting intersected metric is no more optimal. If, we seek the largest ellipsoid included in the geometric intersection region of several (>2) metrics, the John ellipsoid has to be found thanks to an optimization problem [18].

2.3.2. Metric interpolation

In practice, the metric field is only known discretely at the mesh vertices. The use of metric interpolation enables us to define a continuous metric field all over the domain. For instance, the computation of an edge length with relation (1) requires the interpolated metrics along the segment.

Several interpolation schemes have been proposed in [2] which are based on the simultaneous reduction. The main drawback of these approaches is that the interpolation operation is not commutative. Hence, the result depends on the order in which the operations are performed when more than two metrics are interpolated. We suggest to use an interpolation derived from the log-Euclidean framework introduced in [3]. Let $(\mathbf{x}_i)_{i=1,\dots,k}$ be a set of vertices and $(\mathscr{M}(\mathbf{x}_i))_{i=1,\dots,k}$ their associated metrics. Then, for a point $\mathbf{x}$ of the domain such that

$$\mathbf{x}=\sum_{i=1}^{k}\alpha_i\mathbf{x}_i \quad \text{with} \quad \sum_{i=1}^{k}\alpha_i=1,$$

the interpolated metric is defined by

$$\mathscr{M}(\mathbf{x})=\exp\left(\sum_{i=1}^{k}\alpha_i\ln(\mathscr{M}(\mathbf{x}_i))\right). \tag{3}$$

On the diagonalizable matrices space, the logarithm and the exponential act on the metric eigenvalues term to term, that is to say, if we have the following spectral decomposition $\mathscr{M}={}^t\mathscr{R}\Lambda\mathscr{R}$ then:

$$\ln(\mathscr{M})={}^t\mathscr{R}\ln(\Lambda)\mathscr{R} \quad \text{and} \quad \exp(\mathscr{M})={}^t\mathscr{R}\exp(\Lambda)\mathscr{R}.$$

This interpolation is commutative, but its bottleneck is to perform k diagonalization and to request the use of the logarithm and the exponential functions which are time consuming. However, this procedure is essential to define continuously the metric map on the entire domain. Moreover, it has been demonstrated in [3] that this interpolation preserves the maximum principle, i.e., for an edge $\mathbf{pq}$ with endpoints metrics $\mathscr{M}_\mathbf{p}$ and $\mathscr{M}_\mathbf{q}$ such that $\det(\mathscr{M}_\mathbf{p})<\det(\mathscr{M}_\mathbf{q})$ then we have $\det(\mathscr{M}_\mathbf{p})<\det(\mathscr{M}(\mathbf{p}+t\mathbf{pq}))<\det(\mathscr{M}_\mathbf{q})$ for all $t\in[0,1]$.

Accordingly, this metric interpolation enables a continuous metric field to be defined throughout the entire discretized domain. When a metric is required at a point, we determine to which element the point belongs. Then, we apply relation (3), where α_i are the barycentric coordinates of the point with respect to the element.

Remark 2.3. The interpolation formulation (3) reduces to

$$\mathscr{M}(\mathbf{x})=\prod_{i=1}^{k}\mathscr{M}(\mathbf{x}_i)^{\alpha_i}$$

if all the metrics commute. Therefore, a geometric mean in the log-Euclidean space could be interpreted as an arithmetic mean in the space of metrics.

2.4. Numerical computation of length in a metric space

Let $\mathscr{M}$ be a discrete metric field defined at the vertices of a mesh $\mathscr{H}$ of a domain Ω_h. Thanks to the interpolation operation, we have a continuous metric field in the whole domain, i.e., a Riemannian metric space $(\mathscr{M}(\mathbf{x}))_{\mathbf{x}\in\Omega_h}$. This representation of the metric field is dependent of $\mathscr{H}$ as the interpolation law is applied at the element level. If we have two points $\mathbf{x}$ and $\mathbf{y}$ of the domain, the length of segment $\mathbf{xy}$ is given by integral (1). This integral can be numerically approximated using a k-point Gaussian quadrature with weights $(\omega_i)_{i=1,\dots,k}$ and barycentric coefficients $(\alpha_i)_{i=1,\dots,k}$:

$$\ell_\mathscr{M}(\mathbf{xy})=\int_0^1\sqrt{{}^t\mathbf{xy}\mathscr{M}(\mathbf{x}+t\mathbf{xy})\mathbf{xy}}\,\mathrm{d}t\approx\sum_{i=1}^{k}\omega_i\sqrt{{}^t\mathbf{xy}\mathscr{M}(\mathbf{x}+\alpha_i\mathbf{xy})\mathbf{xy}},$$

where $\mathscr{M}(\mathbf{x}+\alpha_i\mathbf{xy})$ is the metric at the Gauss points.

In the context of meshing, we are interested in the evaluation of edge length in the metric. This is a recurrent operation which could be time consuming for the mesh generator. Fortunately, this operation is at the edge level where the metric is known at the endpoints. The problem can thus be simplified by considering a variation law of the metric along the edge and computing analytically the edge length in the metric.

Formally speaking, let $\mathbf{e}=\mathbf{p}_1\mathbf{p}_2$ be an edge of the mesh of Euclidean length $\|\mathbf{e}\|_2$, and $\mathscr{M}_{(\mathbf{p}_1)}$ and $\mathscr{M}_{(\mathbf{p}_2)}$ be the metrics at the edge extremities $\mathbf{p}_1$ and $\mathbf{p}_2$. We denote by $\ell_i={}^t\mathbf{e}\mathscr{M}_{(\mathbf{p}_i)}\mathbf{e}$ the length of the edge in metric $\mathscr{M}_{(\mathbf{p}_i)}$. We assume $\ell_1>\ell_2$ and we set $a=\ell_1/\ell_2$. In the following, we present approximations of $\ell_\mathscr{M}(\mathbf{e})$ by considering several laws for the metric variation along the edge. If we set $\mathbf{u}=\mathbf{e}/\|\mathbf{e}\|_2$, according to relation (2), we have

$$\ell_\mathscr{M}(\mathbf{e})=\|\mathbf{e}\|_2\int_0^1\sqrt{{}^t\mathbf{u}\mathscr{M}(t)\mathbf{u}}\,\mathrm{d}t=\|\mathbf{e}\|_2\int_0^1\frac{1}{h_\mathscr{M}(t)}\,\mathrm{d}t,$$

where we have denoted in short $\mathscr{M}(t)=\mathscr{M}(\mathbf{p}_1+t\mathbf{p}_1\mathbf{p}_2)$ and $h_\mathscr{M}(t)=h_\mathscr{M}(\mathbf{p}_1+t\mathbf{p}_1\mathbf{p}_2)$. Note that we have

$$\ell_1=\frac{\|\mathbf{e}\|_2}{h_1},\quad \ell_2=\frac{\|\mathbf{e}\|_2}{h_2}\quad\text{and}\quad a=\frac{h_2}{h_1}.$$

We suggest the following interpolation laws on h or λ:

- linear interpolation on h, $h(t)=(1-t)h_1+th_2$:

$$\ell_\mathscr{M}(\mathbf{e})=\ell_1\frac{\ln(a)}{a-1},$$

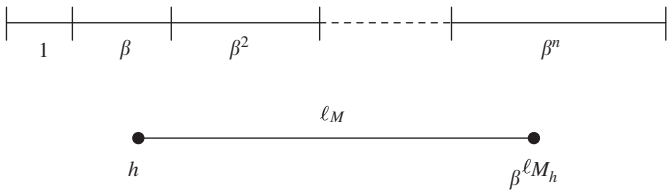


Fig. 4. Top, geometric progression of 1D element size. Down, the equivalent geometric progression of the size prescription.

- linear interpolation on λ, $h(t)=((1-t)h_1^{-2}+th_2^{-2})^{-1/2}$:

$$\ell_{\mathscr{M}}(\mathbf{e})=\ell_1\frac{2}{3}\frac{a^2+a+1}{a(a+1)},$$

- linear interpolation on h^{-1}, $h(t)=((1-t)h_1^{-1}+th_2^{-1})^{-1}$:

$$\ell_{\mathscr{M}}(\mathbf{e})=\ell_1\frac{1}{2}\frac{a+1}{a},$$

- geometric interpolation on h, $h(t)=h_1^{(1-t)}h_2^t$:

$$\ell_{\mathscr{M}}(\mathbf{e})=\ell_1\frac{a-1}{a\ln(a)}.$$

Notice that the geometric interpolation on λ is equivalent to the geometric interpolation on h.

2.4.1. Remark

The restriction of the (multi-dimensional) metric interpolation operator given by Relation (3) to an edge $\mathbf{e}=\mathbf{p}_1\mathbf{p}_2$ leads to

$$\begin{aligned}&\lambda(t)=\exp((1-t)\ln(\lambda_1)+t\ln(\lambda_2)),\\ &\iff \frac{1}{h^2(t)}=\exp\left(\ln\left(\frac{1}{h_1^{2(1-t)}h_2^{2t}}\right)\right),\\ &\iff h(t)=h_1^{(1-t)}h_2^t.\end{aligned}$$

Consequently, this multi-dimensional operator law is equivalent to a geometric interpolation low on h on the edge.

3. Isotropic mesh gradation control

In this section, we recall the main lines of the isotropic mesh gradation introduced in [4].

Let $\mathscr{M}$ be an *isotropic* metric field defined in the entire domain Ω. The metric field can be rewritten $\mathscr{M}=h^{-2}I$ with I the identity matrix. Let $\mathbf{p}$ and $\mathbf{q}$ be two points of the domain with associated sizes h_p and h_q.

Definition 3.1. The *H-shock* or *size gradation* $c(\mathbf{pq})$ related to the segment $\mathbf{pq}$ of Ω is the value:

$$c(\mathbf{pq})=\max\left(\frac{h_p}{h_q},\frac{h_q}{h_p}\right)^{1/\ell_{\mathscr{M}}(\mathbf{pq})}.$$

The anisotropic H-shock value varies in the interval $[1,\infty[$. The H-shock in mesh gradation represents the spatial geometric progression of the size prescription in the metric as depicted in 1D in Fig. 4. This notion generalizes the classical geometrical element size progression: the next element is β times larger than the previous one. Accordingly, if we have a size prescription h and we ask for an increase of β, at a distance $\ell_{\mathscr{M}}$ the size prescription is $\beta^{\ell_{\mathscr{M}}}h$. However, this value measures the variation of the size function h along $\mathbf{pq}$ in all directions and not only in the direction of $\mathbf{pq}$. This measure is extended to any points, for $\mathbf{p}\in\Omega$:

$$c(\mathbf{p})=\max_{\mathbf{q}\in\Omega}c(\mathbf{pq}).$$

The isotropic size correction consists in providing a minimal (optimal) reduction $\widetilde{\mathscr{M}}$ of $\mathscr{M}$ such that, for all points, the size gradation is bounded by a given threshold β. This correction regularizes the metric field and bounds the variations. We now illustrate this mechanism for two points $\mathbf{p}$ and $\mathbf{q}$ such that the H-shock $c(\mathbf{pq})$ is greater than β assuming $h_p\le h_q$. The problem is to reduce h_q so as to obtain a new H-shock equal to β. We denote by $\widetilde{h}_q$ this reduced size and we set $r=\widetilde{h}_q/h_p$. Then, we have

$$\beta=r^{\ell^{-1}_{\mathscr{M}}(\mathbf{pq})}\quad\iff\quad\ell_{\mathscr{M}}(\mathbf{pq})=\frac{\ln(r)}{\ln(\beta)}.$$

Thus, r verifies the following relations depending on the size law given in Section 2.3.2:

- linear interpolation on h:

$$r=1+\ell_p(\mathbf{pq})\ln(\beta),\tag{4}$$

- linear interpolation on λ:

$$\frac{r(r+1)\ln(r)}{r^2+r+1}=\frac{2}{3}\ell_p(\mathbf{pq})\ln(\beta),\tag{5}$$

- linear interpolation on h^{-1}:

$$\frac{r\ln(r)}{r+1}=\frac{1}{2}\ell_p(\mathbf{pq})\ln(\beta),\tag{6}$$

- geometric interpolation on h or λ:

$$\frac{r\ln^2(r)}{r-1}=\ell_p(\mathbf{pq})\ln(\beta).\tag{7}$$

The linear law on h provides an immediate solution for r as

$$r=1+\ell_p(\mathbf{pq})\ln(\beta)=1+\frac{\|\mathbf{pq}\|_2}{h_p}\ln(\beta),$$

which enables us to reduce the size to get analytically a gradation of β on the segment.

In the other cases, we face a non-linear equation to determine r of the form:

$$f(r)=b.$$

In each case, the function of r is strictly increasing, therefore we can retrieve the unique solution by finding the zero of the convex function g:

$$g(r)=(f(r)-b)^2,$$

thanks to a Newton algorithm.

In the following, we only consider the linear law on h for clarity purposes but all the above laws are interchangeable. To sum up, when $\mathscr{M}$ is isotropic, i.e., $\mathscr{M}=h^{-2}I$, for any couple of points $\mathbf{p}$ and $\mathbf{q}$ such that $h_p\le h_q$ and $c(\mathbf{pq})>\beta$, we reduce the metric at vertex $\mathbf{q}$ to

$$\begin{aligned}\widetilde{\mathscr{M}}_{\mathbf{q}}&=\widetilde{h_q}^{-2}I=((1+\ell_p(\mathbf{pq})\ln(\beta))h_p)^{-2}I\\&=(1+\ell_p(\mathbf{pq})\ln(\beta))^{-2}\mathscr{M}_{\mathbf{p}}=\eta^2\mathscr{M}_{\mathbf{p}}.\end{aligned}\tag{8}$$

The reduced metric in $\mathbf{q}$ is given by the metric in $\mathbf{p}$ grown by a scale factor η^2.

4. Anisotropic mesh gradation control

We propose two formulations generalizing in a continuous framework the isotropic mesh gradation to the anisotropic case. In the anisotropic context, the mesh gradation consists in reducing in all directions the size prescribed at any points if the variation of the metric field is larger than a fixed threshold. As we control the variations of the metric in all directions, it is difficult to propose a direct anisotropic extension of the H-shock notion.

The previous analysis leads us to suggest the following continuous vision of the anisotropic mesh gradation control. Each point of the domain span a metric field in the whole domain by growing its metric at a rate given by the desired gradation coefficient, Fig. 5 (left). These fields represent the size constraint imposed by each point in the entire domain. Then, each point of the domain has to take into account these constraints to guarantee a metric field with a smooth variation controlled by the size gradation. At each point, the metric is reduced by intersecting all these contributions, Fig. 5 (right). Now, we have to specify how a pointwise metric spans a metric field in the domain.

4.1. Spanning a metric field

Let $\mathbf{p}$ be a point of a domain Ω supplied with a metric $\mathcal{M}_{\mathbf{p}}$ and β the specified gradation. We propose two laws governing the metric growth in the domain. In the first one, the metric growth is homogeneous in the Euclidean metric field defined by $\mathcal{M}_{\mathbf{p}}$. The second metric growth is homogeneous in the physical space, i.e., the classical Euclidean space.

The first law is the direct extension of the isotropic case. For any point $\mathbf{x}$ of the domain, it associates a unique scale factor with the metric given by

$$\eta^2(\mathbf{px}) = (1 + \ell_p(\mathbf{px})\ln(\beta))^{-2} = (1 + \sqrt{{}^t\mathbf{px}\,\mathcal{M}_{\mathbf{p}}\mathbf{px}}\ln(\beta))^{-2}.$$

With this formulation, each pointwise metric $\mathcal{M}_{\mathbf{p}}$ spans a global continuous smooth metric field all over domain Ω parametrized by the given gradation value β:

$$(\mathcal{M}_{\mathbf{x}(\mathbf{p})})_{\mathbf{x}\in\Omega} \quad \text{with } \mathcal{M}_{\mathbf{x}(\mathbf{p})} = \eta^2(\mathbf{px})\mathcal{M}_{\mathbf{p}}. \tag{9}$$

In this case, the resulting metric field grows homogeneously in the Euclidean metric space defined by $\mathcal{M}_{\mathbf{p}}$ as the scale factor depends on the length of segment $\mathbf{px}$ with respect to $\mathcal{M}_{\mathbf{p}}$. As a result, the shape of the metric is kept unchanged while growing. This law conserves the same anisotropic ratio.

Fig. 6 illustrates this law at the ball level for several gradation parameters. A metric $\mathcal{M}_{\mathbf{p}}$ is set at the ball center which spans a metric field to all the other vertices of the ball. Another example of a metric $\mathcal{M}_{\mathbf{p}}$ specified at a point $\mathbf{p}$ and spanning a metric field in the entire domain with a prescribed gradation parameter is illustrated in Fig. 8, left.

For the second law, we associate independently a growth factor with each eigenvalue of $\mathcal{M}$:

$$\eta_i^2(\mathbf{px}) = (1 + \sqrt{\lambda_i}\|\mathbf{px}\|_2\ln(\beta))^{-2} \quad \text{for } i = 1,\dots,3.$$

The grown metric at $\mathbf{x}$ is given by

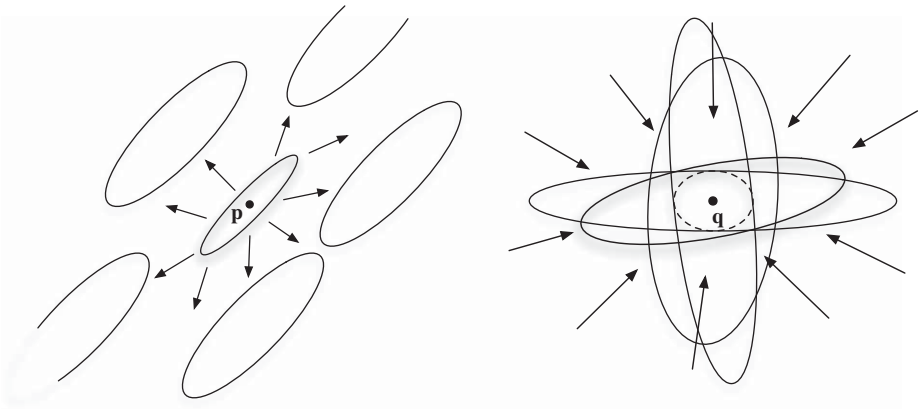
$$\mathcal{M}_{\mathbf{p}}(\mathbf{x}) = {}^t\mathcal{R}\mathcal{N}(\mathbf{px})\Lambda\mathcal{R},$$

where

$$\mathcal{N}(\mathbf{px}) = \begin{pmatrix} \eta_1^2(\mathbf{px}) & 0 & 0 \\ 0 & \eta_2^2(\mathbf{px}) & 0 \\ 0 & 0 & \eta_3^2(\mathbf{px}) \end{pmatrix}.$$

From the point $\mathbf{p}$, a global continuous smooth metric field is defined all over domain Ω as

$$(\mathcal{M}_x(\mathbf{p}))_{\mathbf{x}\in\Omega} \quad \text{with } \mathcal{M}_{\mathbf{p}}(\mathbf{x}) = {}^t\mathcal{R}\mathcal{N}(\mathbf{px})\mathcal{M}_{\mathbf{p}} \quad \text{and} \quad \eta_i^2(\mathbf{px}) = (1 + \sqrt{\lambda_i}\|\mathbf{px}\|_2\ln(\beta))^{-2}. \tag{10}$$



Spanning a metric field **Metric reduction**

Fig. 5. Illustration of the two-stepped mesh gradation algorithm. Firstly, each point spans a metric field in the whole domain by growing its metric at a rate given by the desired gradation coefficient. Secondly, the reduced metric at each point is obtained by intersecting all the defined metric fields at this point.

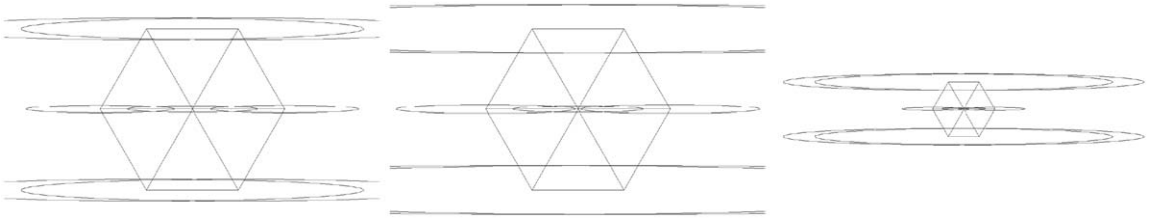


Fig. 6. A metric $\mathcal{M}_{\mathbf{p}}$ is set at the ball center. Illustration of the first law (relation (9)) spanning a metric field to all the other vertices of the ball. The metric growth is homogeneous in the Euclidean metric space defined by $\mathcal{M}_{\mathbf{p}}$. Gradation coefficients have been set to 1.1 (left) and 1.3 (middle). Right, a far view of the metric field with a gradation of 1.3.

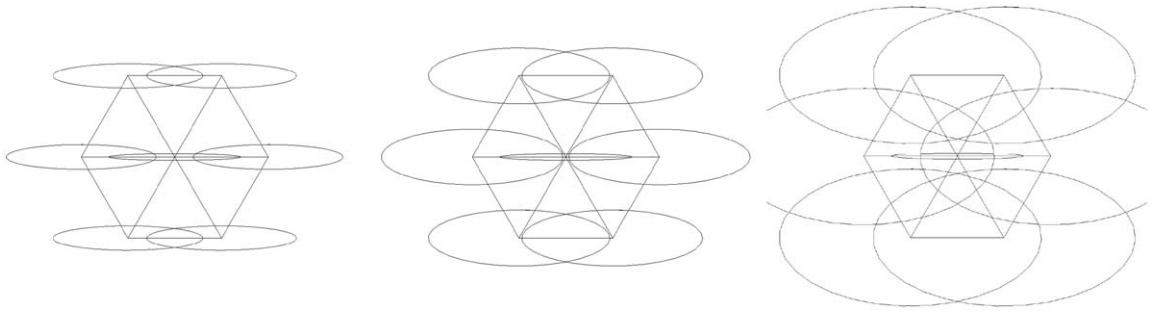


Fig. 7. A metric $\mathcal{M}_{\mathbf{p}}$ is set at the ball center. Illustration of the second law (relation (10)) spanning a metric field to all the other vertices of the ball. The metric growth is homogeneous in the standard Euclidean space defined by l_2. From left to right, gradation coefficients of 1.1, 1.3 and 2 have been set.

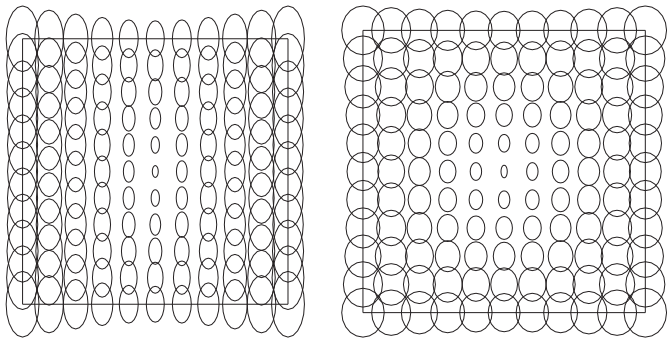


Fig. 8. A metric $\mathcal{M}$ is specified at the center of the domain. This metric spans a metric field in the entire domain thanks to the mesh gradation. Left, the homogenous gradation in the metric space, the metrics keep their shape (anisotropy) while growing. Right, the homogenous gradation in the physical space, the metrics become isotropic while growing.

The resulting metric field grows homogeneously in the physical space. Indeed, each eigenvalue grows similarly in all directions, as the factor η_i depends only on the distance (in the physical space) from the original point. Consequently, the shape, i.e., the anisotropic ratio of the metric, is no more preserved as the eigenvalues are growing separately and differently. This law gradually makes the metric more and more isotropic as it gradually propagates in the domain.

Fig. 7 illustrates this law at the ball level for several gradation parameters. A metric $\mathcal{M}_{\mathbf{p}}$ is set at the ball center which spans a metric field to all other vertices of the ball. Another example of a metric $\mathcal{M}_{\mathbf{p}}$ specified at a point $\mathbf{p}$ and spanning a metric field in the entire domain with a prescribed gradation parameter is illustrated in Fig. 8, right.

Remark 4.1. In each case, the spanned metric field represents the size constraint in all directions imposed by vertex $\mathbf{p}$ under the gradation requirement β in the whole domain.

4.2. Metric reduction

The reduced metric at a point $\mathbf{x}$ of the domain Ω is given by the strongest size constraint imposed by the metric at $\mathbf{x}$ and by the spanned metrics (parametrized by the given size gradation) of all the other points of the domain at $\mathbf{x}$:

$$\widetilde{\mathcal{M}(\mathbf{x})} = \left(\bigcap_{\mathbf{p}\in\Omega} \mathcal{M}_{\mathbf{p}}(\mathbf{x}) \right) \cap \mathcal{M}(\mathbf{x}).$$

Remark 4.2. This way of thinking results from the continuous modeling of a mesh proposed in [19]. The metric at each point defines a well-graded smooth continuous mesh over the domain. The mesh gradation then consists in taking into account at each point the strongest size constraint given by all these continuous meshes.

5. Numerical examples

We first discuss the practical implementation of the mesh gradation algorithm. Then, several applications are presented.

5.1. Practical implementation

In the case of a mesh $\mathcal{H}$ supplied with a discrete metric field given at its vertices, each vertex provides a metric for all the other vertices that imposes its size constraints in all directions. The reduced metric at vertex $\mathbf{q}$ is then given by

$$\widetilde{\mathcal{M}(\mathbf{q})} = \left(\bigcap_{\mathbf{p}\in\mathcal{H}} \mathcal{M}_{\mathbf{p}}(\mathbf{q}) \right) \cap \mathcal{M}(\mathbf{q}).$$

Unfortunately, we face a quadratic complexity algorithm. To avoid this, the mesh gradation problem is approximated with a linear complexity algorithm based on the mesh edges. More precisely, the size correction is performed edge by edge. Let $\mathbf{pq}$ be an edge of the mesh with endpoints metrics $\mathcal{M}_{\mathbf{p}}$ and $\mathcal{M}_{\mathbf{q}}$. We defined the grown metrics $\mathcal{M}_{\mathbf{q}}(\mathbf{p})$ and $\mathcal{M}_{\mathbf{p}}(\mathbf{q})$ on both extremities of the edge which are given by Relation (9) or (10).

Then, the reduction is performed at each vertex by a metric intersection:

$$\widetilde{\mathcal{M}(\mathbf{p})} = \mathcal{M}(\mathbf{p}) \cap \mathcal{M}_{\mathbf{q}}(\mathbf{p}) \quad \text{and} \quad \widetilde{\mathcal{M}(\mathbf{q})} = \mathcal{M}(\mathbf{q}) \cap \mathcal{M}_{\mathbf{p}}(\mathbf{q}). \tag{11}$$

The information is propagated in the whole domain through an iterative algorithm:

```
While correction = 1
  correction = 0
  Loop over the edge of H
    Let pq be the current edge, apply the reduction
    process summarized by relation (11).
    If one metric is modified then correction = 1
```

To accelerate the edge treatment, a dynamic list of the edge to be examined is considered. Notice that checking *a posteriori* if the gradation value is verified in all directions is equivalent to test if one ellipsoid is included in the other ellipsoid.

It is important to note that the considered edge-based propagation of the size constraints assumes that the size growth law is transitive. It is the case for the linear law on h but not for the other

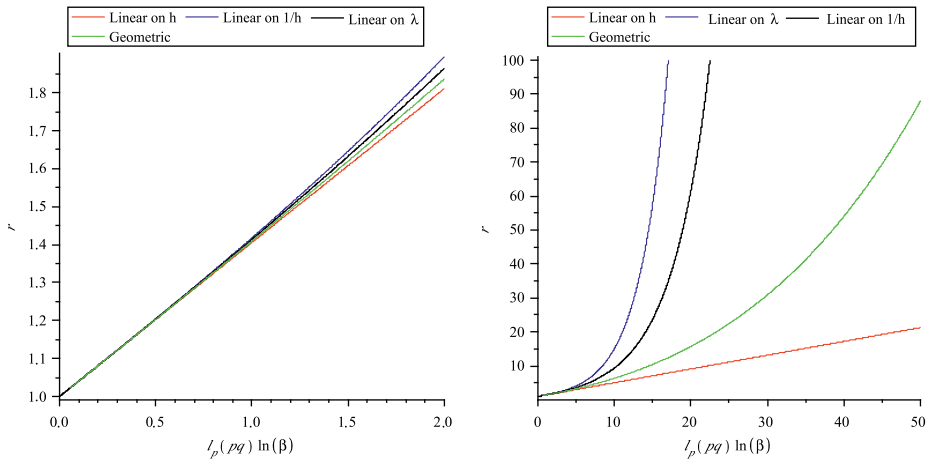


Fig. 9. Plot of the size growth r function of $\ell_p(\mathbf{pq})\ln(\beta)$. In red, the linear law on h, in blue the linear law on λ, in black the linear law on h^{-1} and in green, the geometric law on h. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

ones. This is illustrated in Fig. 9 where, for each law, size growth r is plotted as a function of the distance in the considered metric $\ell_p(\mathbf{pq})$ for a fixed gradation coefficient. We see that all the laws are quite similar when $\ell_p(\mathbf{pq})$ is close to unity which is generally the case for all the edges surrounding a vertex in a unit mesh, left plot. But, they evolve really differently for large distances, right plot. In consequence, the proposed edge-based algorithm cannot take into account the differences between each law.

5.2. Generation of graded quasi-uniform meshes

The first application is the generation of a quasi-uniform volume mesh around a geometry where the gradation is controlled. The volume mesh accuracy close to the geometry is managed thanks to the gradation parameter. The unique available data are the surface mesh, i.e., the mesh of the boundary. These data enable us to define a metric field in the entire domain by means of the mesh gradation algorithm.

More precisely, let $\mathscr{S}$ be a surface mesh. After generating an initial volume mesh, we compute a metric at each vertex of the surface mesh. To this end, for each surface triangle K, we evaluate in the triangle plane the unique 2D metric for which this triangle is unit:

$$\overline{\mathscr{M}}_K = {}^t(\bar{\mathbf{t}}_1\bar{\mathbf{t}}_2)\begin{pmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{pmatrix}(\bar{\mathbf{t}}_1\bar{\mathbf{t}}_2).$$

Then, the size on the triangle normal $\mathbf{n}$ is set through a user parameter. For instance, it can be a fixed size $h_n = h_{fix}$ or $h_n = \alpha\min(\lambda_1^{-1/2}, \lambda_2^{-1/2})$ where α is prescribed. It results in a 3D metric for each element of the surface mesh:

$$\mathscr{M}_K = {}^t(\mathbf{t}_1\mathbf{t}_2\mathbf{n})\begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & h_n^{-2}\end{pmatrix}(\mathbf{t}_1\mathbf{t}_2\mathbf{n}).$$

The metric at each vertex $\mathbf{p}$ of the surface is obtained by a local $\mathbf{L}^2$-projection in the log-Euclidean framework (see Section 2.3):

$$\mathscr{M}_{\mathbf{p}} = \exp\left(\frac{\sum_{K_i\ni P}|K_i|\ln(\mathscr{M}_{K_i})}{\sum_{K_i\ni P}|K_i|}\right).$$

This metric, only defined on the surface, is propagated in the entire domain thanks to the mesh gradation procedure. We illustrate this generation of graded mesh on two examples in three dimensions.

5.2.1. Isotropic mesh of an aircraft

We consider an aircraft in a spherical domain and we aim at generating an isotropic graded mesh with a Delaunay-based adaptive mesh generator [11]. We specify a size gradation of 1.2 and 1.5 and we obtain meshes made up of 3 574 143 and 946 236 tetrahedra, respectively. These meshes are shown in Fig. 10.

5.2.2. Anisotropic mesh of a wing profile

In this example, the goal is to generate an anisotropic graded mesh of a wing profile in a spherical domain whose surface mesh is anisotropic. In this case, we use the Delaunay-based adaptive local remesher described in [7]. The mesh gradation method, which is homogeneous in the metric space, is considered. Size gradations of 1.5 and 2 are prescribed resulting in anisotropic meshes containing 148 440 and 58 577 tetrahedra, respectively. These meshes are shown in Fig. 11.

5.3. Level set application

We present another application where the mesh gradation enables a global continuous metric field to be defined from local data. Here, the local data are a surface given by an implicit function.

In the context of the simulation of bi-fluid flows or more generally multi-fluid flows, the interface can be modeled by the level set method. In this case, a level set function is considered to represent the location of the interface separating two non-miscible fluids, for instance air and water. Formally speaking, let ϕ be a level set function, $\Phi_0=\{\mathbf{x}|\phi(\mathbf{x})=0\}$ describes the interface and $\phi(\mathbf{y})$ is the distance of $\mathbf{y}$ to the interface. Moreover, by definition the function ϕ verifies: $\nabla\phi = 1$ and $\mathbf{n}_\phi = \nabla\phi$. In the context of level set applications, we aim at meshing accurately the interface between the two fluids and at coarsening the mesh as we gradually get further from the interface. We can define a metric for the interface, i.e., Φ_0, of the form [8,13,18]:

$$\mathscr{M}_{\Phi_0} = {}^t(\nabla\phi\nabla\phi^\perp)\begin{pmatrix}h_{user}^{-2} & 0\\ 0 & \varepsilon^{-2}\end{pmatrix}(\nabla\phi\nabla\phi^\perp),$$

where h_{user} is the size in the direction normal to Φ_0 prescribed by the user and ε is the size in the direction tangent to Φ_0 which

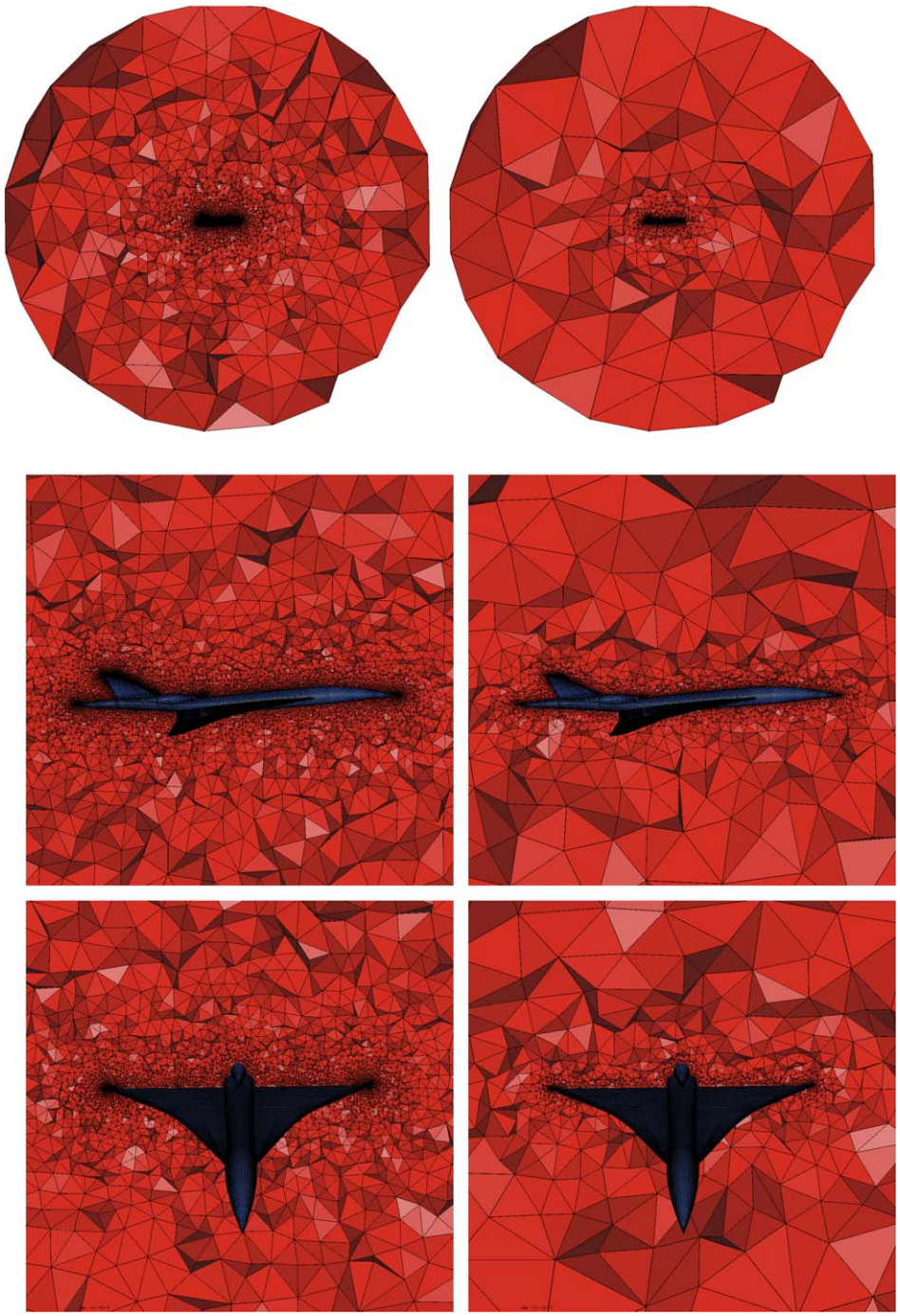


Fig. 10. Graded isotropic meshes of an aircraft in a spherical domain. Left, the mesh for a size gradation of 1.2 containing 3 574 143 tetrahedra. Right, the mesh for a size gradation of 1.5 containing 946 236 tetrahedra.

depends on a specific error model, for instance the curvature of Φ_0. The mesh gradation determines a metric field continuously in the overall domain:

$$\forall \mathbf{y} \in \Omega, \quad \mathscr{M}_\Phi(\mathbf{y}) = \bigcap_{\mathbf{x}\in\Phi_0} (1 + \ell_{\mathscr{M}_\Phi(\mathbf{x})}(\mathbf{xy}) \ln(\beta)) \mathscr{M}_\Phi(\mathbf{x}).$$

Two bi-fluid examples studied in [13] are presented.

Fig. 12 displays the use of this metric for a 2D simulation of a breaking water column which impacts an obstacle. The level set function is shown in the left and the adapted mesh generated from the level set metric in the right. The wave interface impacting the obstacle is clearly visible inside the mesh.

The 3D case is shown in Fig. 13. The complexity of the interface is illustrated in the top picture, it represents a breaking wave with several tubes. The adapted surface and volume meshes are displayed in the middle and bottom pictures, respectively.

In both cases a size gradation of 2 has been prescribed.

5.4. Anisotropic mesh adaptation

In this section, two 2D examples and one 3D example of the mesh gradation applied to anisotropic mesh adaptation are presented. In each case, the homogeneous gradation in the metric space, named *method 1*, is compared to the homogenous one in the physical space,

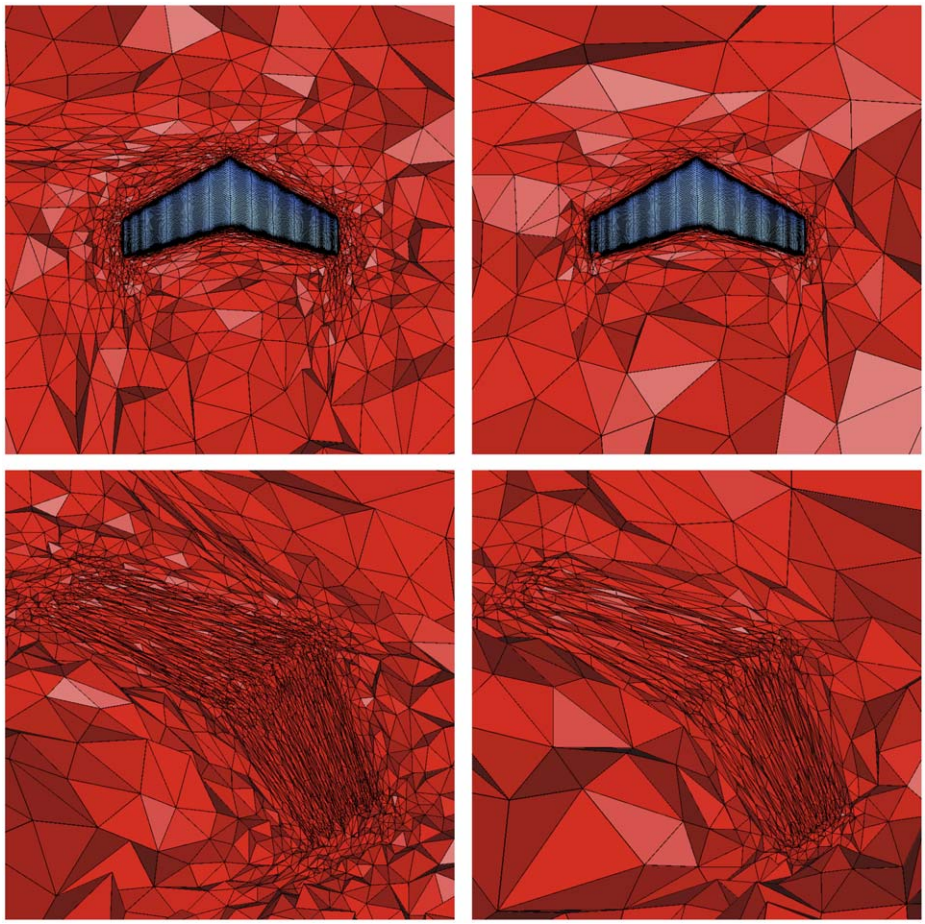


Fig. 11. Graded anisotropic meshes of a wing profile in a spherical domain. Left, the mesh for a size gradation of 1.5 containing 148 440 tetrahedra. Right, the mesh for a size gradation of 2 containing 58 577 tetrahedra.

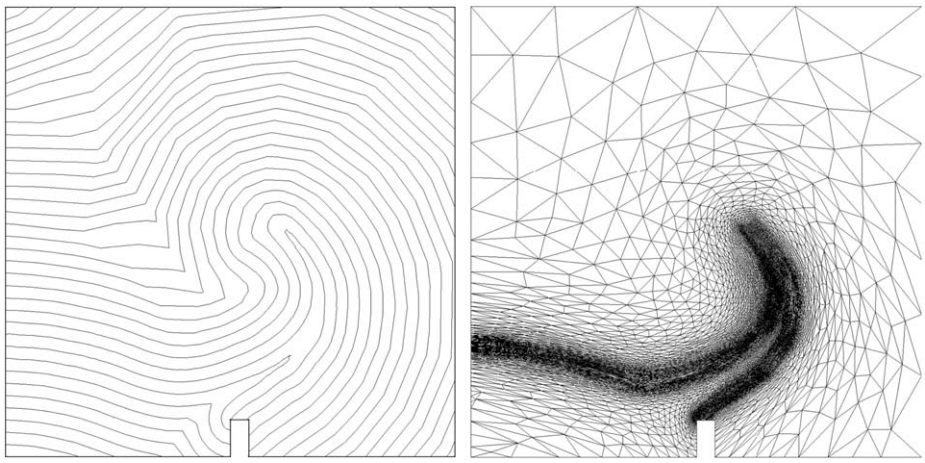


Fig. 12. Application of the anisotropic mesh gradation to a 2D level set simulation. Left, the level set function of a water wave impacting an obstacle. Right, the adapted mesh generated from the level set metric. The interface is clearly visible inside the mesh.

called *method 2*. The adaptive remeshers presented in [9,7] are considered in 2D and 3D, respectively.

5.4.1. A scramjet

The scramjet configuration at Mach 3 is typical of numerical simulations in compressible fluids involving highly anisotropic phenomena with very strong shocks. The solution of this simulation results in a complex flow with several shock reflexions and shocks that cross over. However, all these shock waves have simple straight line shapes. This flow solution leads to a metric field prescribing large aspect ratio elements in shock region with a very small size prescription in the direction orthogonal to the shock, and

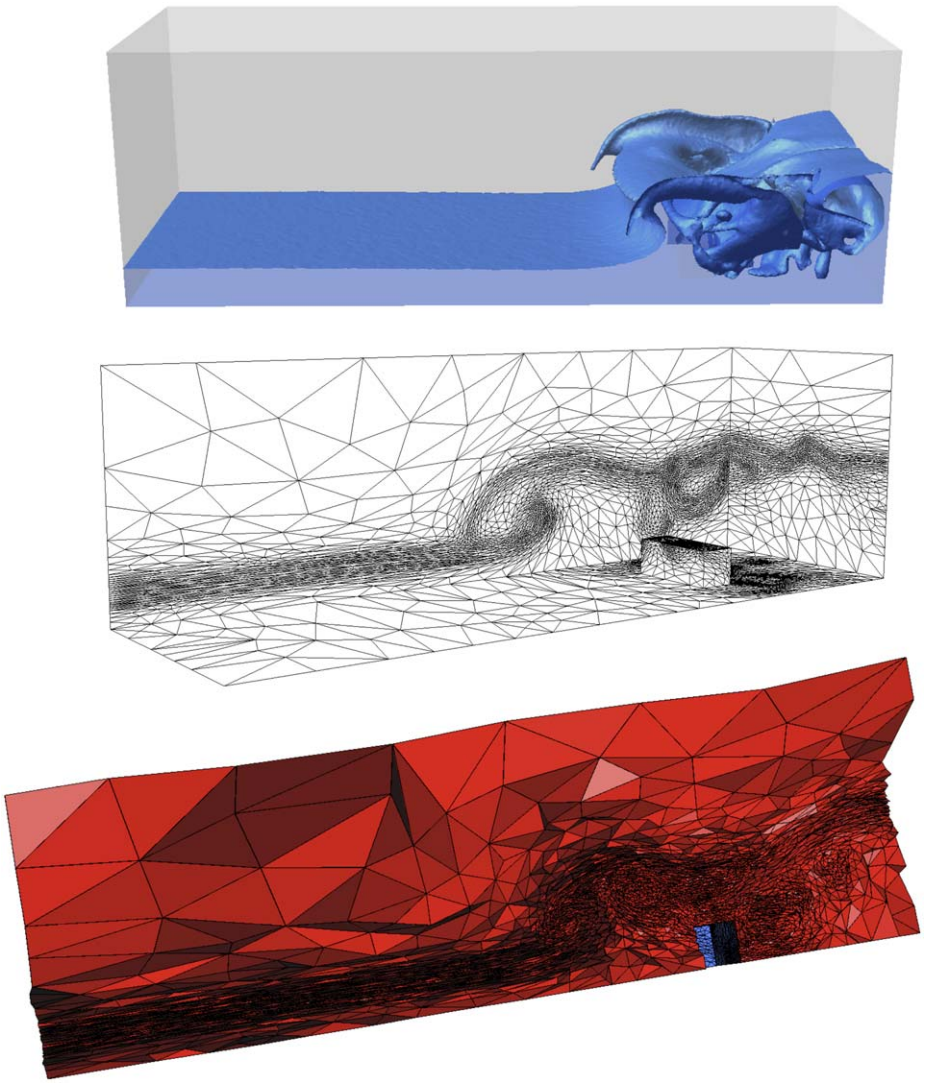


Fig. 13. Application of the anisotropic mesh gradation to a 3D level set simulation. Up, interface represented by the zeros of the level set function. Middle, the adapted surface mesh, and down, the adapted volume mesh generated from the level set metric. The interface is clearly visible inside the mesh.

Table 1
Number of vertices of the adapted meshes in each case for the scramjet simulation.

	$\beta = 1.3$	$\beta = 1.5$	$\beta = 2$	$\beta = 10^{12}$
Metric-space-homogeneous gradation	51 456	33 754	21 908	8959
Physical-space-homogeneous gradation	50 538	32 991	21 841	–

prescribing large sizes where the solution is smooth or constant. It outcomes a particularly non-regular metric field. Mesh adaptations have been performed with method 1 and method 2 and several gradation parameters. A mesh adaptation with no gradation has been also done (in fact we have prescribed a gradation of 10^{12}). Table 1 summarizes all the performed cases and the meshes sizes.

First of all, the adapted mesh with no gradation is depicted in Fig. 14. We observe that the mesh is of poor quality for a numerical computation and that the mesh generator has a lot of difficulties to generate elongated elements in shock regions. The mesh lacks of resolution in shock region. This is due to the impossibility of generating a unit mesh in the given non-regular metric field. In particular, in the neighboring region of a shock, the computed metric is highly dependent on the vertex position. If the vertex is inside the shock then an appropriate metric is computed, but if the vertex is slightly outside the shock, the information is lost. This is illustrated in Fig. 19 where the metric fields are shown before and after the gradation process. The mesh gradation clearly regularizes the initial non-smooth metric field.

The meshes obtained with method 1 and method 2 are presented in Figs. 16 and 17. In each case shock regions are meshed anisotropically with a good quality as compared to the case with no mesh gradation, see the close-up view in Fig. 15. The anisotropy is preserved and a high resolution is obtained. However, the results of the two methods are particularly different, even if for the same gradation parameter they provide meshes of almost the same size. The elements become isotropic when growing with the method 2, whereas they remain anisotropic along the shock waves with method 1.

More precisely, method 2 provides moderate anisotropic meshes as compared to method 1. This is illustrated in Fig. 18. Indeed, close to anisotropic regions, the mesh becomes smoothly isotropic when growing. This is due to the "isotropic" propagation of the size constraints with this gradation method and to the fact that

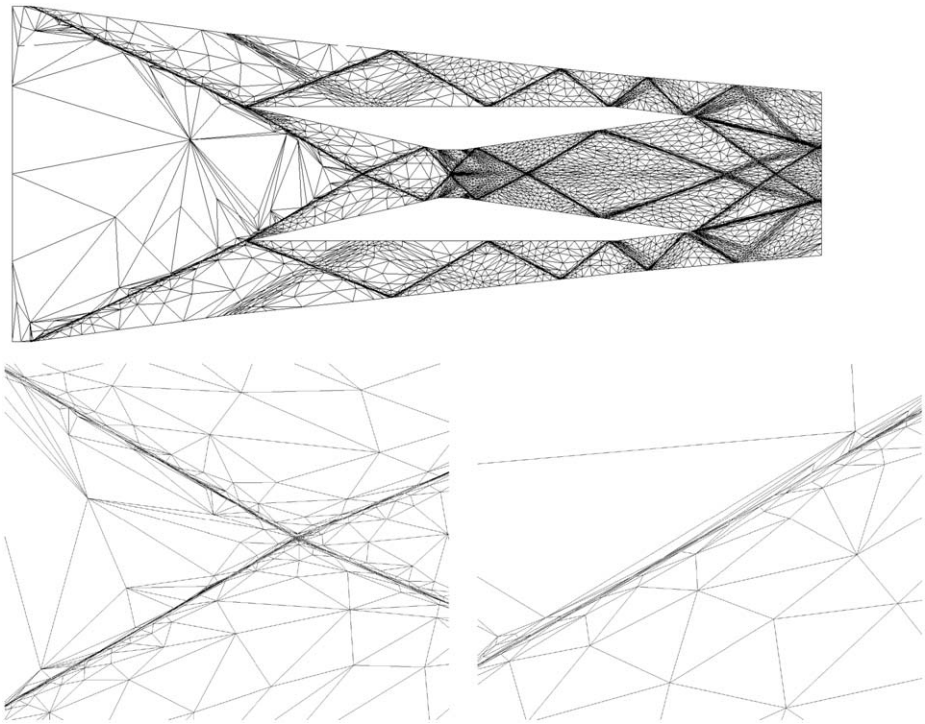


Fig. 14. Adapted mesh resulting from the simulation of a scramjet with no mesh gradation.

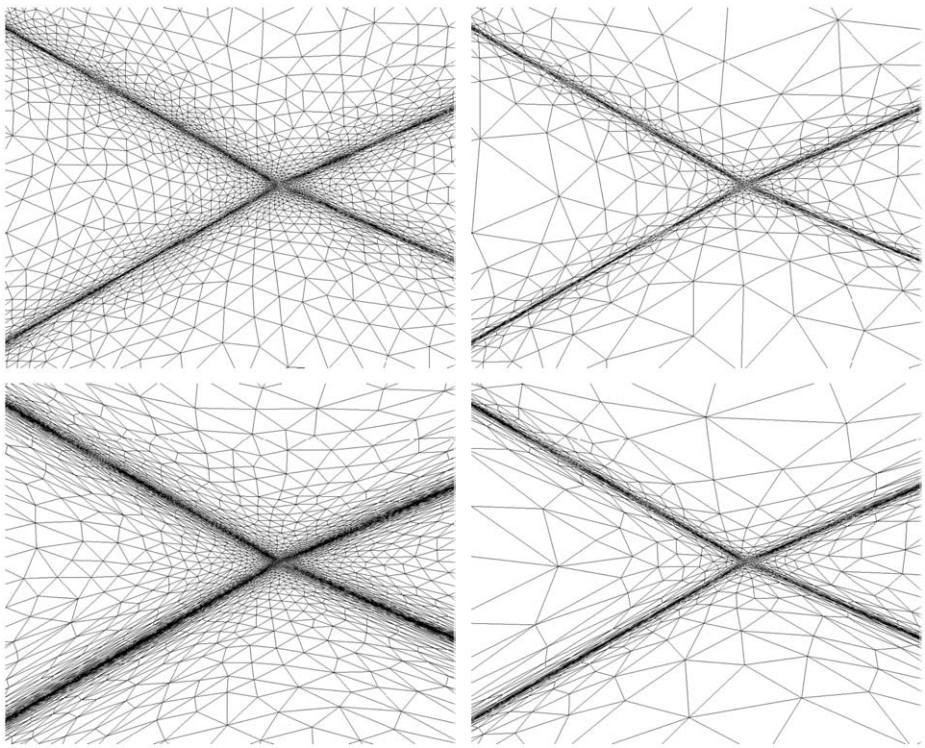


Fig. 15. Adapted meshes resulting from the simulation of a scramjet with the physical-space-homogenous (top) and the metric-space-homogeneous (bottom) gradation. This zoom on the mesh shows that the anisotropy has been preserved in each case. Left, a gradation coefficients of 1.3 and right a value of 2.

a small size prescription grows faster with this method than method 1.

As method 1 propagates the element anisotropy and the size prescription grows homogeneously in the metric space, it enforces more anisotropy in the generated mesh. More precisely, it provides a very fine and highly stretched mesh in shocks neighboring. In this example, as the shocks are straight lines, method 1 improves the initial size prescription by carrying correctly the anisotropic information

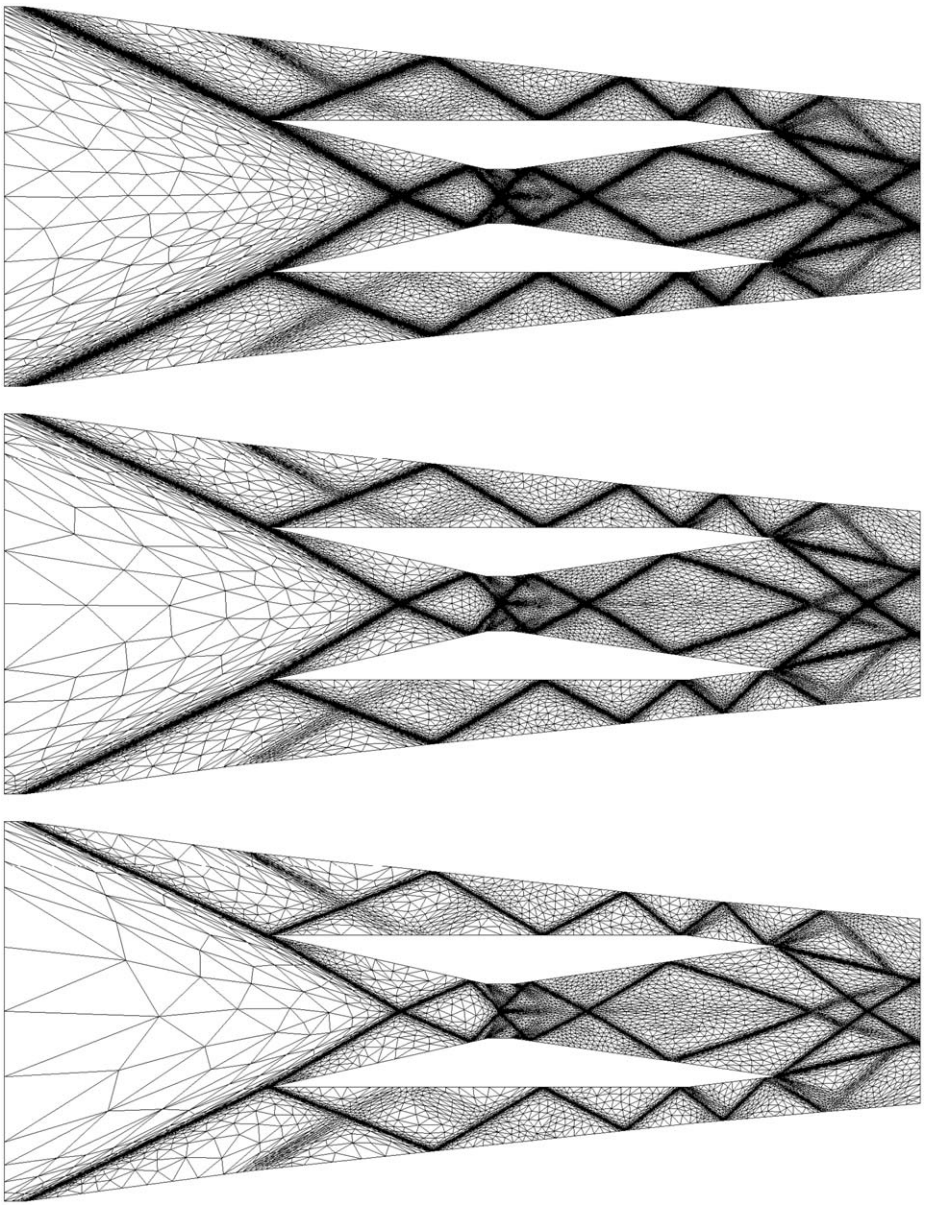


Fig. 16. Adapted meshes resulting from the simulation of a scramjet with the metric-space-homogeneous gradation. From top to bottom, gradation coefficients of 1.3, 1.5 and 2 have been used.

in the domain. In other words, the mesh gradation "helps" the mesh generator to produce a more accurate and a more anisotropic mesh by adding additional information thanks to the control of the metric variation, cf. Fig. 18.

In conclusion, for such simulations, the mesh gradation is mandatory to generate high-quality anisotropic meshes. Method 1 enhances the resolution and the accuracy of the initial field by propagating the anisotropic information. This method provides a higher resolution and more anisotropic elements in shock regions. Method 2 results in meshes with moderate accuracy and anisotropy in shock regions, although the size is consistent with the initial metric field. In consequence, in this case the metric-space-homogeneous gradation seems to be more suitable.

5.4.2. A blast in a town

The second example comes from the simulation of a blast in a town. In this application, the shock waves have curved shapes and there are several shock interactions. A global view of the meshes obtained with a gradation of 1.3 and 2 for each case is depicted in Fig. 20. The meshes statistics are given in Table 2.

In this example, the mesh accuracy and its anisotropy are quite similar with both methods. The same anisotropy is obtained in shock regions, see Fig. 21 for a close-up view in such regions of the adapted meshes. As compared to the scramjet example, no improvement is noticeable with method 1.

For method 1, the mesh gradation tries to fit the shock curvature with anisotropic elements. But, sometimes it seems to fail. The elements do not "turn" and it creates visible artifacts in the mesh like a "ray". In this example, this phenomenon is weak but it can be more important. As the metrics grow in an isotropic way with method 2, the elements follow easily the shock curvature as the mesh get bigger.

To conclude, for this application with curved shocks, both methods are equivalent in terms of anisotropy and accuracy. However, method 2 better handles the metric curvature.

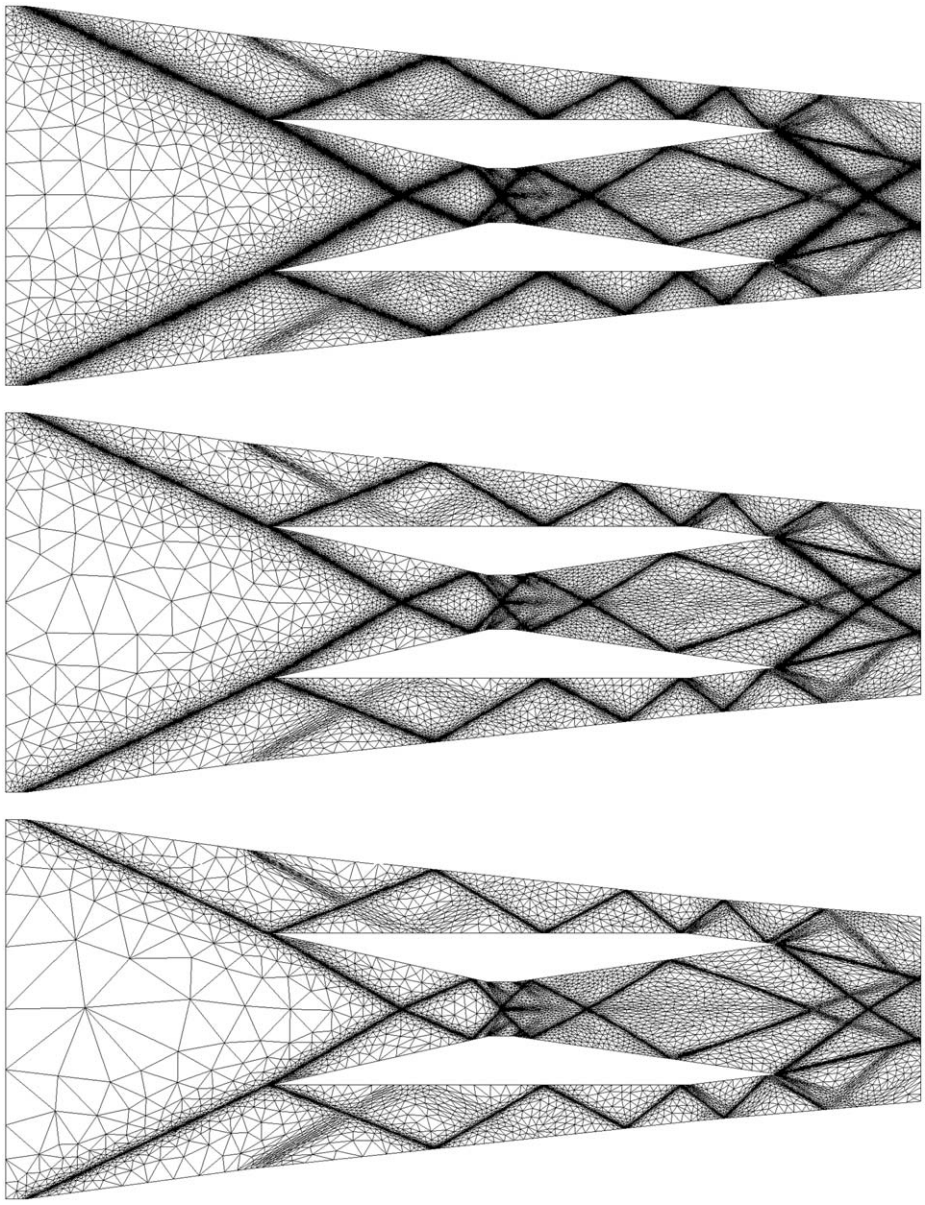


Fig. 17. Adapted meshes resulting from the simulation of a scramjet with the physical-space-homogeneous gradation. From top to bottom, gradation coefficients of 1.3, 1.5 and 2 have been used.

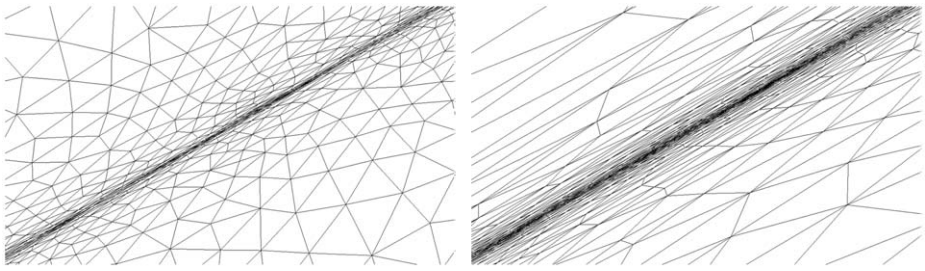


Fig. 18. Adapted meshes resulting from the simulation of a scramjet with the physical-space-homogenous (left) and the metric-space-homogeneous (right) gradation with a coefficients of 1.3.

5.4.3. A supersonic aircraft

The final example regards the adaptive simulation of a supersonic aircraft, the aircraft is the geometry presented in Section 5.2. A jet flying at a supersonic cruise speed generates a large number of shock waves. These shock waves are cone-shaped. In fact, they represent the Mach cones emitted by the geometry.

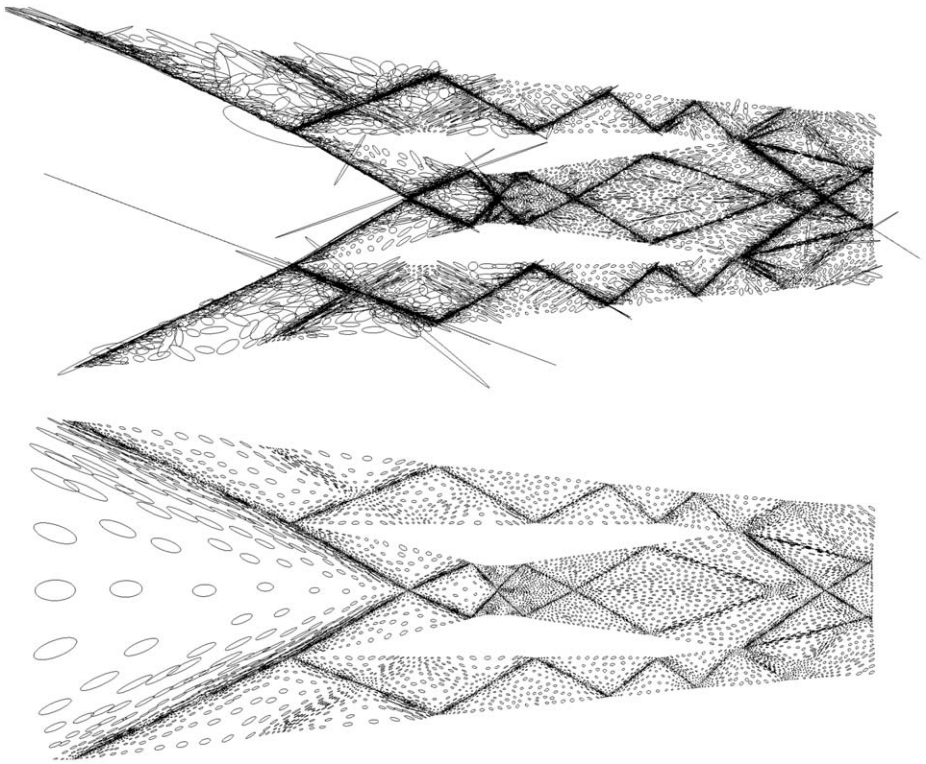


Fig. 19. Global view of the metric field at each vertex, the unit ball scaled by one-fourth of each metric is plotted. Top, the initial metric field computed with the error estimate. Bottom, the new metric field after gradation. It clearly illustrates that mesh gradation regularizes the metric field.

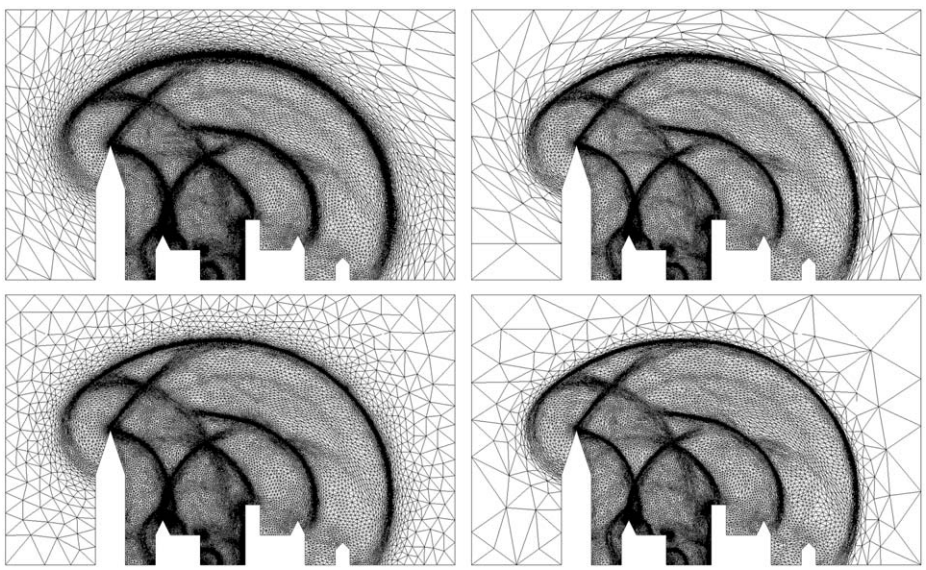


Fig. 20. Adapted meshes resulting from the blast simulation with gradation coefficients of 1.3 (left) and 2 (right). Top, the metric-space-homogeneous metric. Down, the physical-space-homogeneous metric.

Table 2
Number of vertices of the adapted meshes in each case for the blast simulation.

	$\beta = 1.3$	$\beta = 2$
Metric-space-homogeneous gradation	37 514	25 767
Physical-space-homogeneous gradation	35 609	26 215

For this simulation, a gradation of 2 has been considered. The adapted meshes obtained with method 1 and method 2 contain, respectively, 756 481 vertices and 3 934 348 tetrahedra, and 720 389 vertices and 3 768 416 tetrahedra. They are shown in Fig. 22.

For this 3D simulation the conclusions are the same as for the scramjet example. Both meshes have almost the same size in terms of vertices but they are quite different. Method 1 provides a mesh which is more anisotropic. This is emphasized in the right pictures of Fig. 22 where the cut plane is orthogonal to the cones. The meshes illustrate once again the anisotropic behavior of method 1 and the isotropic behavior of method 2. To conclude, for this example, method 1 has our favor.

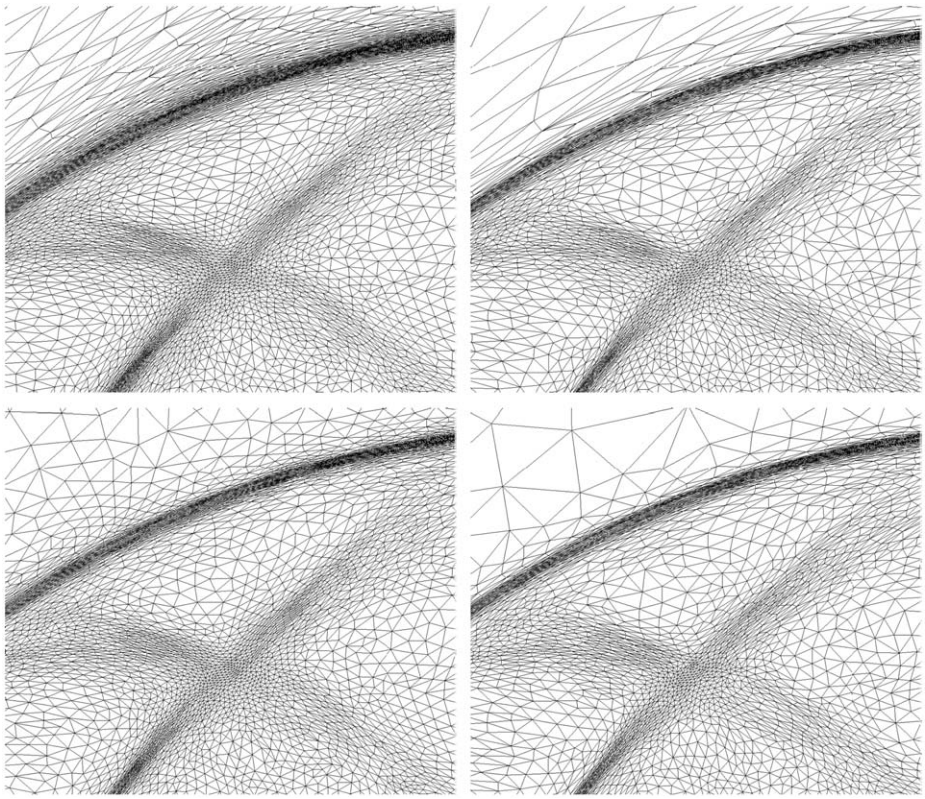


Fig. 21. Close-up view in shock regions of the adapted meshes resulting from the blast simulation with gradation coefficients of 1.3 (left) and 2 (right). Top, the metric-space-homogeneous metric. Down, the physical-space-homogeneous metric.

6. Improvement of the previous approaches

6.1. Limits of the edge-based mesh gradation algorithm

Although the edge-based approach, presented in Section 5.1, provides good results in most of the practical cases, this algorithm has weaknesses. They are due to the coarse approximation of the quadratic complexity algorithm of the mesh gradation. In the following, these weaknesses are pointed out on a 2D analytical example. We first describe how the mesh adaptation scheme is performed in the context of an analytical case.

We have an initial mesh. The metric is computed at mesh vertices thanks to the analytical function. Then, the mesh gradation algorithm is applied to this discrete metric field to define the new reduced metric field. This reduced metric field is prescribed to the mesh generator to obtain the new adapted mesh. This procedure is repeated until algorithmic convergence. It is important to note that the adaptive mesh generator only knows the pointwise metric field. It has no knowledge of the analytical function or of the mesh gradation. When vertices are moved or inserted, the mesh generator uses its own internal metric interpolation scheme to update the metric.

6.1.1. The circle analytical example

This analytical metric field specifies constant sizes of 0.1 along the tangents of the circle centered at the origin and of radius 1, and sizes of 0.0005 along orthogonal directions with a variation towards an isotropic metric while moving away from the circle. The analytical metric associated with any point (x,y) of the domain reads:

$$\mathscr{M}(x,y) = {}^t\mathscr{R}(x,y)\begin{pmatrix} h_1^{-2}(x,y) & 0 \\ 0 & h_2^{-2}(x,y) \end{pmatrix}\mathscr{R}(x,y),$$

where the rotation matrix is

$$\mathscr{R}(x,y) = \begin{pmatrix} \cos(\theta(x,y)) & -\sin(\theta(x,y)) \\ \sin(\theta(x,y)) & \cos(\theta(x,y)) \end{pmatrix}$$

and

$$\theta(x,y) = \arctan\left(\frac{x}{y}\right),$$

$$h_1(x,y) = 0.0005 + 1.5|1 - \sqrt{x^2+y^2}|,$$

$$h_2(x,y) = 0.1\sqrt{x^2+y^2} + 1.5|1 - \sqrt{x^2+y^2}|.$$

For this example, the initial mesh of the domain $\Omega = [-2,2]^2$ is a coarse uniform mesh made up of 2821 vertices.

6.1.2. Results with the first growth law

When the first law, given by relation (9) and now called *metric-space-gradation*, is considered for the metric growth, topological artifacts looking like rays appear inside the adapted mesh. This phenomenon is exemplified in Fig. 23. It is all the more important as the size gradation parameter is small. And, it is also dependent of the initial mesh topology and of the order in which the vertices are treated. A random choice when dealing with the vertices list reduces this phenomenon, see Fig. 24. Indeed, it seems to disappear in Fig. 24 (right) but lowering the mesh gradation reveals it. Consequently, in this case, the mesh gradation with this growth law seems to fail.

Let us give some explanations about this ray phenomenon. We have seen in Fig. 6 that the metric-space-gradation is highly dependent on the mesh topology. This implies that if mesh edges are perfectly aligned with a metric orthogonally to the direction of a small size prescription then this small size constraint is propagated far in the domain because it grows very slowly due to the h-linear size variation law. In this case, the metric propagation path in the iterative algorithm is a straight line aligned with the metric. On the other hand, if the path is not straight, then in the direction orthogonal to a small size prescription, this small size constraint grows at a rate faster than what it should be. In other words, in this case, smaller sizes have been enlarged too rapidly leading to a mesh coarser than what is expected.

To sum up, the rays that appear inside the mesh are not artifacts but are real size constraints that have been propagated correctly by the mesh propagation algorithm because of mesh alignments and the small growth of the linear law. As regards the coarser mesh regions, size constraints have not been propagated correctly due to the topological dependence of the edge-based algorithm. It results in a metric field that grows faster than it should. This is illustrated in Fig. 26 where the original metric is plotted on the left and the metric after gradation with this law is given in the middle plot. The ray phenomena is clearly visible inside the graded metric field which proves that it is not an artifact due to the mesh generator but it comes from the gradation algorithm.

6.1.3. Results with the second growth law

If the second law for the metric growth, given by relation (10) and now called *physical-space-gradation*, is employed, then the ray phenomenon does not appear, cf. Fig. 25. As growth is a function of the distance in the physical space, this approach is really less sensitive to the mesh topology. The impact of a non-straight propagation path on the metric growth is then insignificant. This is also illustrated in Fig. 26 on the right plot, the graded metric field is following the curvature of the circle. Moreover, this law is also slightly sensitive to the order in which the vertex list is handled.

Nevertheless, a close-up view of the circle, depicted in Fig. 29, reveals that the size of the elements with this law is really larger than the one obtained with the metric-space-gradation. The maximal accuracy, which is only prescribed on the circle of radius one, is not reached. Improving the mesh accuracy requires to reduce the size gradation. The cause of this problem is that small sizes grow really faster with this law, see Fig. 7. Therefore, if vertices are in the region where the minimal size is prescribed, then the minimal size constraint impacts only in a small surrounding region as it rapidly grows. And, if vertices are not exactly in the region where the minimal size is prescribed then this information is ignored. In consequence, this approach does not help the mesh generator to capture the analytical metric field.

6.2. Improvement of the mesh gradation algorithm

We propose two ways to improve the gradation algorithm.

6.2.1. Mixed-space homogeneous gradation

We suggest to mix the two previous approaches to significantly attenuate the ray phenomenon and to keep the accuracy of the generated mesh. For this new law, called *mixed-space-gradation*, a growth factor is associated with each eigenvalue of $\mathscr{M}$ independently:

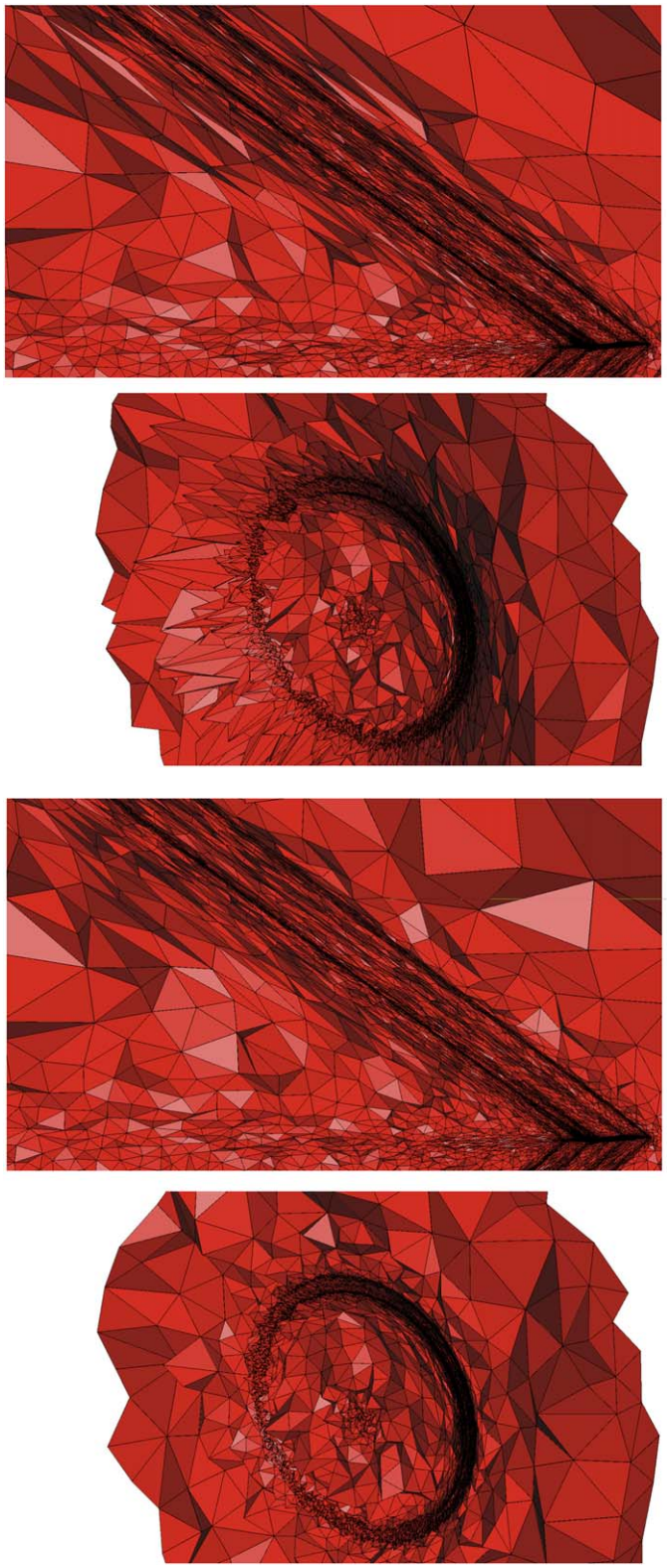


Fig. 22. Adapted mesh for the supersonic aircraft simulation with the metric-space-homogeneous metric and a gradation coefficient of 2.

$$\eta_i^2(\mathbf{px}) = \left((1+\sqrt{\lambda_i}\|\mathbf{px}\|_2 \ln(\beta))^t (1+\ell_p(\mathbf{px})\ln(\beta))^{1-t}\right)^{-2} \text{ for } 0 \le t \le 1 \text{ and } i = 1,\dots,3.$$

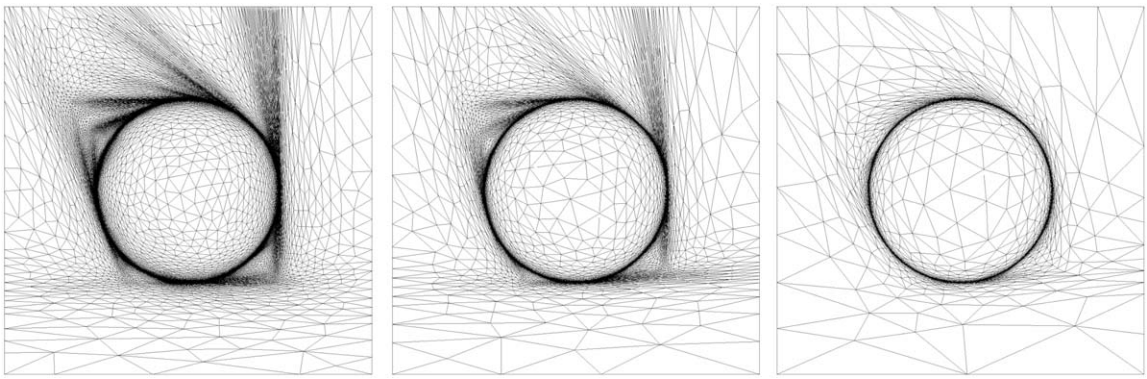


Fig. 23. Circle analytical example for the edge-based algorithm with the metric-space-gradation for the metric growth. From left to right, a gradation of 1.3, 1.5 and 2 have been prescribed.

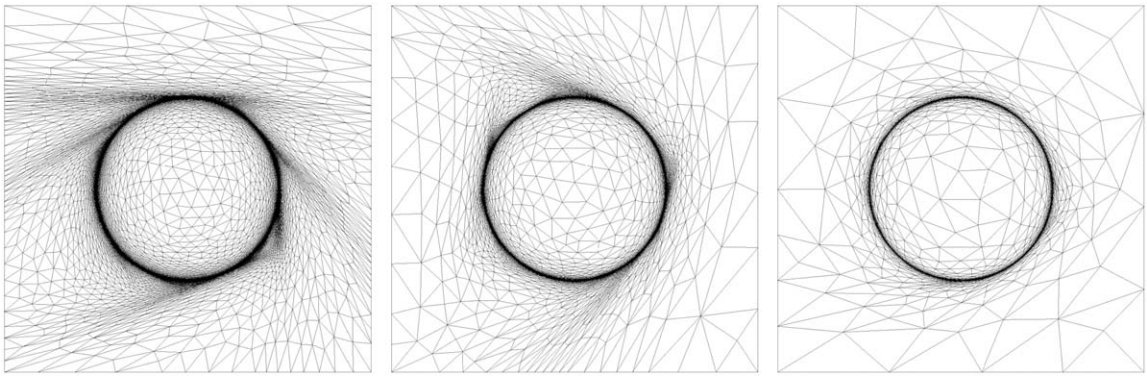


Fig. 24. Circle analytical example for the edge-based algorithm with the metric-space-gradation for the metric growth. Vertices are handled with a random order. From left to right, gradations of 1.3, 1.5 and 2 have been prescribed.

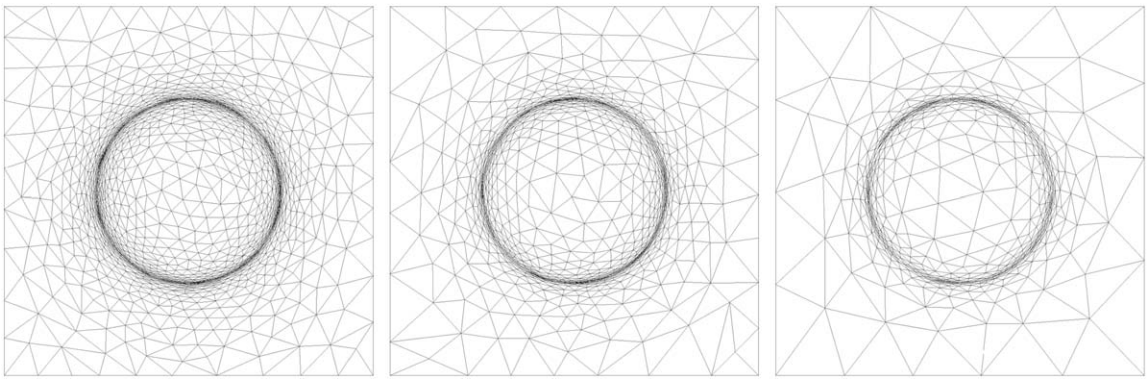


Fig. 25. Circle analytical example for the edge-based algorithm with the physical-space-gradation for the metric growth. From left to right, a gradation of 1.3, 1.5 and 2 have been prescribed.

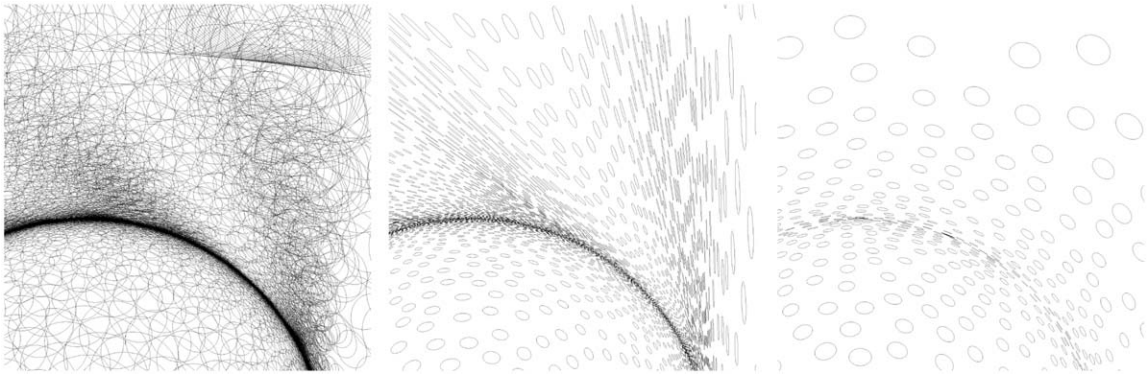


Fig. 26. Circle analytical example. Left, the original metric. Middle, the resulting metric after gradation with the metric-space-gradation for the metric growth. Right, the resulting metric after gradation with the physical-space-gradation for the metric growth.

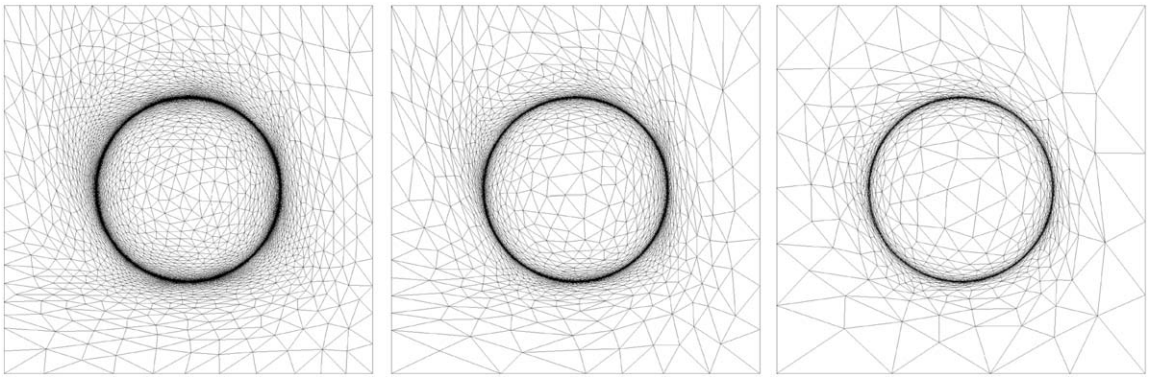


Fig. 27. Circle analytical example for the super-edge-based algorithm with the mixed-space-gradation for the metric growth. From left to right, gradations of 1.3, 1.5 and 2 have been prescribed.

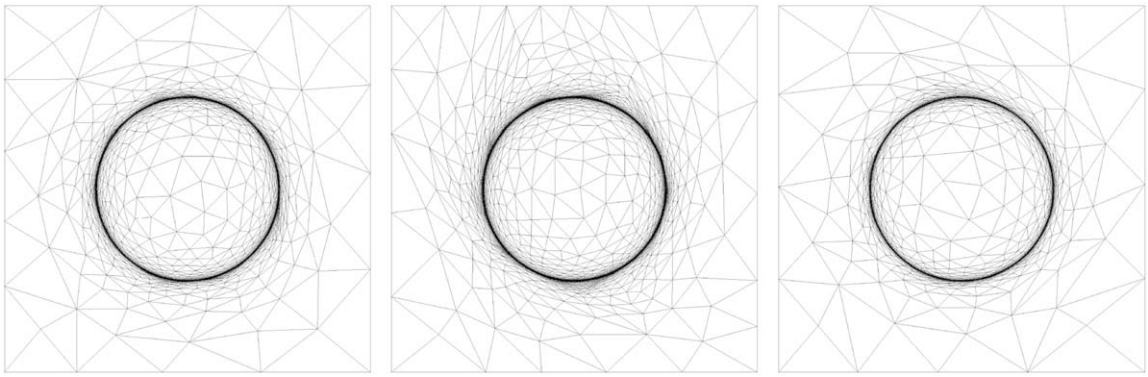


Fig. 28. Circle analytical example for the edge-based algorithm with exp-metric-space-gradation for the metric growth. From left to right, gradations of 1.3, 1.5 and 2 have been prescribed.

The growth metric at $\mathbf{x}$ is given by

$$\mathscr{N}_{\mathbf{p}}(\mathbf{x})\mathscr{M}_{\mathbf{p}} = {}^{t}\mathscr{R}\mathscr{N}(\mathbf{px})\Lambda\mathscr{R},$$

where

$$\mathscr{N}(\mathbf{px}) = \begin{pmatrix} \eta_1^2(\mathbf{px}) & 0 & 0 \\ 0 & \eta_2^2(\mathbf{px}) & 0 \\ 0 & 0 & \eta_3^2(\mathbf{px}) \end{pmatrix}.$$

From the point $\mathbf{p}$, a global continuous smooth metric field is defined all over domain Ω as

$$(\mathscr{M}_{\mathbf{p}}(\mathbf{x}))_{\mathbf{x}\in\Omega} \quad \text{with } \mathscr{M}_{\mathbf{p}}(\mathbf{x}) = {}^{t}\mathscr{R}\mathscr{N}(\mathbf{px})\Lambda\mathscr{R}. \tag{12}$$

In the numerical examples presented below, we have considered $t = \frac{1}{8}$.

Moreover, to reduce the dependence on the mesh topology, we extend the edge-based algorithm to super-edge of order n. The super-edge of order n are the virtual edges connecting each vertex to all the vertices of its order n ball. These super-edges are constructed using the vertices ball data structure. In the presented examples, we have considered the super-edge of order 2, i.e., each vertex is connected to its neighbor and to the neighbor of its neighbor.

However, this algorithm is still incompatible with size variation laws different from the h-linear law as it also requires the transitivity property.

6.2.2. Approximating an exponential size variation law

In order to significantly attenuate the ray phenomenon for the metric-space-gradation, we suggest to combine:

- An algorithm simulating an exponential size variation law instead of using the h-linear size variation law.
- A random choice when dealing with the vertices list.

To achieve a pseudo-exponential size variation law, the idea is to increase the gradation coefficient β while getting farther from a vertex. In this case, the size constraint imposed by a vertex becomes local and has no more a global impact because its size constraint is canceled by the exponential variation. As the restriction to the h-linear size variation law is due to the edge-based algorithm, the pseudo-exponential size variation law is obtained by proposing a modified edge-based algorithm with a dynamic gradation coefficient. The information is then propagated in the whole domain through the new iterative algorithm:

```
While correction = 1
correction = 0
copy the current metric M in M^new
Loop over the edge of H in a randomize order
  Let pq be the current edge, apply the reduction process
  to each vertex:
  M^new(p) = M^new(p) ∩ M_q(p) and M^new(q) = M^new(q) ∩ M_p(q)
  If one metric is modified then correction = 1 and
  increase the size gradation factor to β = αβ with α > 1
```

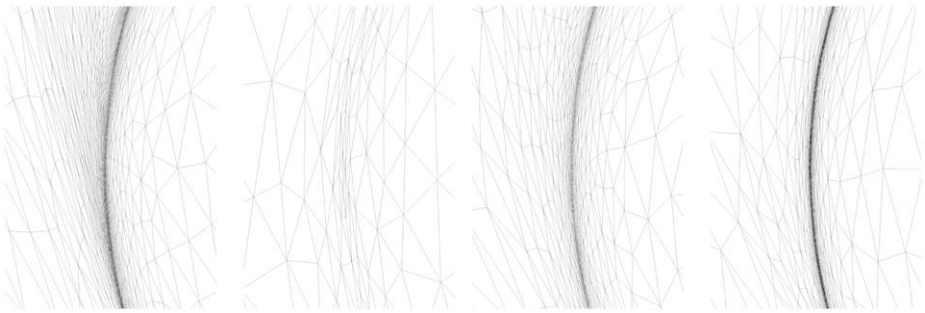


Fig. 29. Circle analytical example for each metric growth law with a gradation of 1.5. From left to right, the metric-space-gradation, the physical-space-gradation, the mixed-space-gradation and the exp-metric-space-gradation.

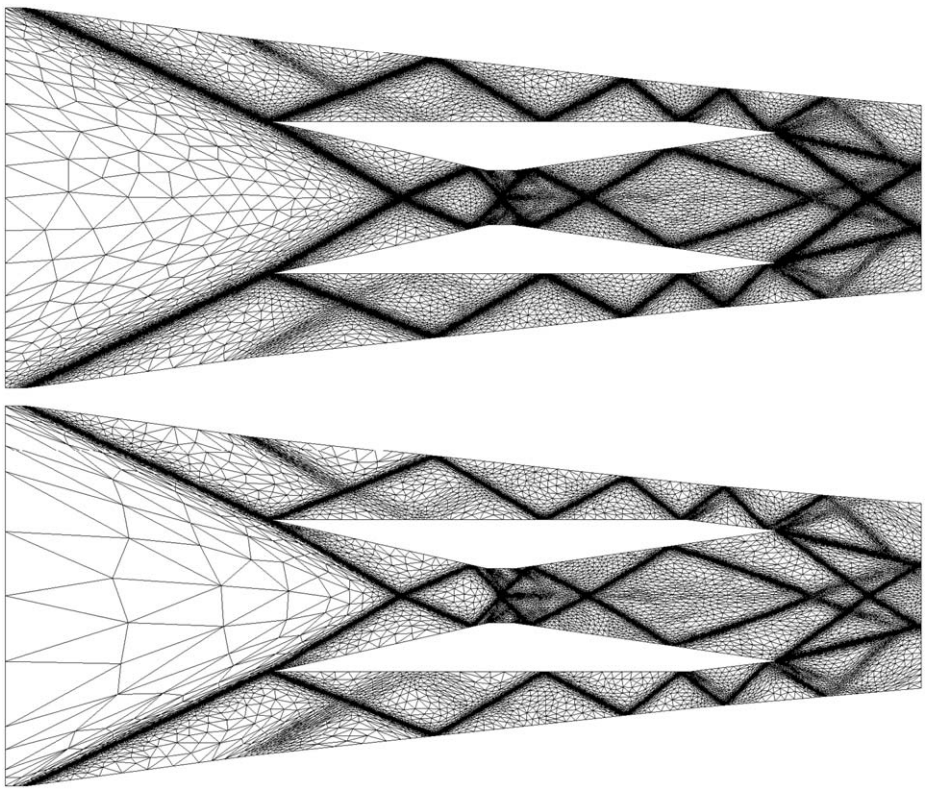


Fig. 30. Adapted mesh resulting from the simulation of a scramjet for the mixed-space-gradation (top) and the exp-metric-space-gradation (bottom). The gradation coefficient is 1.3.

In this algorithm, storing the reduced metric in $\mathscr{M}^{new}$ at each step implies that the size constraint associated with each vertex is propagated layer by layer, e.g. a vertex impacts first is first-order ball, then at iteration two its ball of order two, etc. Consequently, a vertex imposes a size gradation factor of β to its ball of order one, $\alpha\beta$ to its ball of order two, …, $\alpha^n\beta$ to its ball of order n, …. In other words, this algorithm simulates an exponential size variation law by applying the h-linear size variation law with a dynamic increasing gradation coefficient while gradually getting further from a vertex. Modifying the parameter α has a similar impact than switching from the h-linear size variation law to the λ-linear or h-geometric size variation law.

This algorithm is topology dependent because the increased gradation coefficient is not a function of the distance but it is a function of a topological distance, i.e., the ball order. Therefore, to reduce the dependence on the mesh topology, the edges are treated randomly. In the numerical examples presented below, we have considered coefficients α equal to 1.1, 1.02 and 1.02 for gradations β equal to 1.3, 1.5 and 2, respectively. Generally, larger α are associated with smaller gradation factor. This new law is called *exp-metric-space-gradation*.

6.2.3. Application to examples

Results on the circle analytical example are shown in Figs. 27 and 28 for size gradations equal to 1.3, 1.5 and 2. As expected, the ray phenomenon has been significantly attenuated. Notice in Fig. 28 how the mesh becomes rapidly coarse while getting further from the circle due to the pseudo-exponential size variation law.

As regards the mesh accuracy on the circle, we notice in Fig. 29 that it is now satisfactory even if it is less accurate for the mixed-space-gradation than for the metric-space-gradation. Nonetheless, reducing the parameter t will increase the mesh accuracy but it will limit the rays attenuation. On the other hand, the exp-metric-space-gradation does not degrade at all the accuracy on the circle.

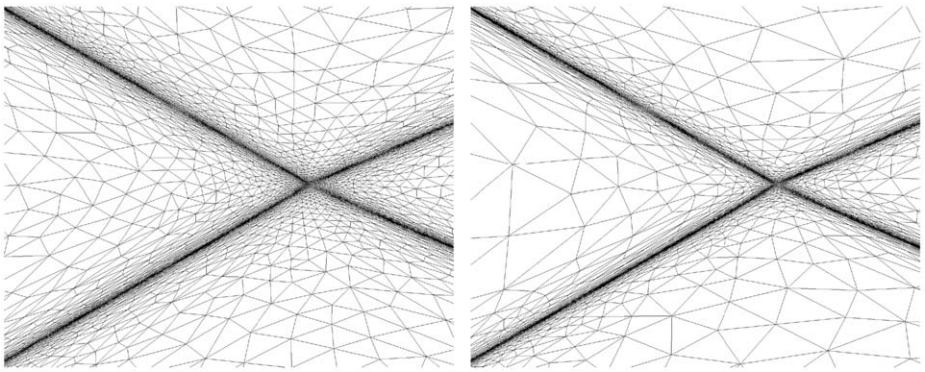


Fig. 31. Close-up view of the adapted mesh resulting from the scramjet simulation for the mixed-space-gradation (left) and the exp-metric-space-gradation (right). The gradation coefficient is 1.3.

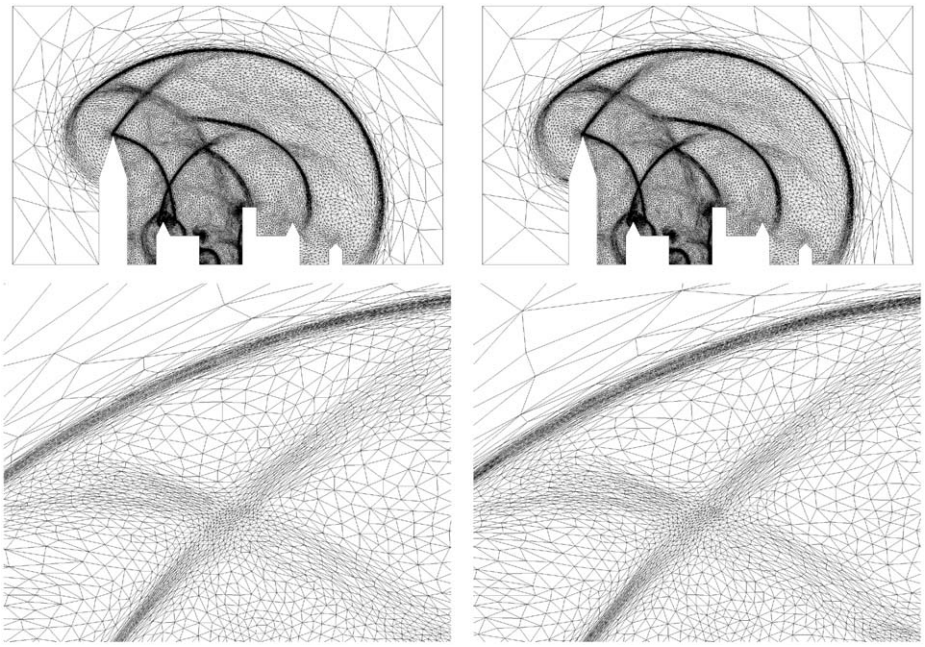


Fig. 32. Adapted meshes resulting from the blast simulation for the mixed-space-gradation (left) and the exp-metric-space-gradation (right). The gradation coefficient is 2.

The improved approaches have been applied to the scramjet and the town blast examples. The scramjet result for a gradation of 1.3 is shown in Figs. 30 and 31. The mesh is of good quality and the accuracy in shock regions has been preserved. The town blast results are presented in Fig. 32 where a gradations of 2 has been prescribed. For this example too, the meshes are accurate and of good quality. We do not see the ray effect as compared to Fig. 20.

7. Concluding remarks

We have presented and evaluated two anisotropic formulations of the mesh gradation. All approaches generally provide good results for realistic applications. The main problem with these approaches is that we must restrict to linear transitive law for the size variation.

The metric-space-gradation achieve better accuracy and higher anisotropic meshes but it encounters the ray phenomenon in specific cases when anisotropic metrics are prescribed along curves. This is due to the rough approximation of the mesh gradation problem and it is strongly dependent on the mesh topology.

The physical-space-gradation provides meshes with moderate accuracy and anisotropy. However, it is not affected by the ray phenomenon as the metrics gradually become isotropic while growing. This approach is also insensitive to the mesh topology.

A new method, the mixed-space-gradation, generalizing the previous ones has been proposed which attenuates these weaknesses. In addition, it considers a super-edge-based algorithm to better approximate the quadratic problem of the mesh gradation. This method achieves good accuracy and results in highly anisotropic meshes. It reduces significantly the ray phenomenon and the sensitivity to the mesh topology.

Finally, a new algorithm to simulate an exponential size variation law instead of the linear one has been suggested. It enables us to cancel the ray phenomenon while keeping the accuracy of the metric-space-gradation. However, this approach requires to set a new parameter which defines the exponential law.

Acknowledgments

I wish to thank A. Loseille and P.L. George for their constructive remarks concerning this work.

References

[1] F. Alauzet, Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en Mécanique des Fluides, Ph.D. Thesis, Université Montpellier II, Montpellier, France, 2003 (in French).

[2] F. Alauzet, P.J. Frey, Estimateur d'erreur géométrique et métrique anisotropes pour l'adaptation de maillage. partie I: aspects théoriques, RR-4759, INRIA, March 2003 (in French).

[3] V. Arsigny, P. Fillard, X. Pennec, N. Ayache, Log-Euclidean metrics for fast and simple calculus on diffusion tensors, Magn. Reson. Med. 56 (2) (2006) 411–421.

[4] H. Borouchaki, F. Hecht, P.J. Frey, Mesh gradation control, Int. J. Numer. Methods Eng. 43 (6) (1998) 1143–1165.

[5] C.L. Bottasso, Anisotropic mesh adaption by metric-driven optimization, Int. J. Numer. Methods Eng. 60 (2004) 597–639.

[6] Y. Bourgault, M. Picasso, F. Alauzet, A. Loseille, On the use of anisotropic error estimators for the adaptative solution of 3-D inviscid compressible flows, Int. J. Numer. Methods Fluids 59 (2009) 47–74.

[7] C. Dobrzynski, P.J. Frey, Anisotropic delaunay mesh adaptation for unsteady simulations, in: Proceedings of the 17th International Meshing Rountable, Springer, Berlin, 2008, pp. 177–194.

[8] V. Ducrot, P.J. Frey, Anisotropic level set adaptation for accurate interface capturing, in: Proceedings of the 17th International Meshing Rountable, Springer, Berlin, 2008, pp. 159–176.

[9] P.J. Frey, Yams, a fully automatic adaptive isotropic surface remeshing procedure, RT-0252, INRIA, November 2001.

[10] P.J. Frey, F. Alauzet, Anisotropic mesh adaptation for CFD computations, Comput. Methods Appl. Mech. Eng. 194 (48–49) (2005) 5068–5082.

[11] P.-L. George, Tet meshing: construction, optimization and adaptation, in: Proceedings of the 8th International Meshing Rountable, South Lake Tao, CA, USA, 1999.

[12] C. Gruau, T. Coupez, 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric, Comput. Methods Appl. Mech. Eng. 194 (48–49) (2005) 4951–4976.

[13] D. Guegan, Modélisation numérique d'écoulements bifluides 3D instationnaires avec adaptation de maillage, Ph.D. Thesis, Université de Nice Sophia Antipolis, Sophia Antipolis, France, 2007 (in French).

[14] F. Hecht, B. Mohammadi, Mesh adaptation by metric control for multi-scale phenomena and turbulence, AIAA Paper, 97-0859, 1997.

[15] W.T. Jones, E.J. Nielsen, M.A. Park, Validation of 3d adjoint based error estimation and mesh adaptation for sonic boom reduction, AIAA paper, 2006-1150, 2006.

[16] X. Li, J.-F. Remacle, N. Chevaugeon, M.S. Shephard, Anisotropic mesh gradation control, in: Proceedings of the 13th International Meshing Rountable, Springer, Berlin, 2004, pp. 401–412.

[17] X. Li, M.S. Shephard, M.W. Beal, 3D anisotropic mesh adaptation by mesh modification, Comput. Methods Appl. Mech. Eng. 194 (48–49) (2005) 4915–4950.

[18] A. Loseille, Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique, Ph.D. Thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2008 (in French).

[19] A. Loseille, F. Alauzet, Continuous mesh model and well-posed continuous interpolation error estimation, RR-6846, INRIA, March 2009.

[20] A. Loseille, A. Dervieux, P.J. Frey, F. Alauzet, Achievement of global second-order mesh convergence for discontinuous flows with adapted unstructured meshes, AIAA paper, 2007-4186, 2007.

[21] C.C. Pain, A.P. Humpleby, C.R.E. de Oliveira, A.J.H. Goddard, Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations, Comput. Methods Appl. Mech. Eng. 190 (2001) 3771–3796.

[22] S. Pippa, G. Caligiana, Gradh-correction: guaranteed sizing gradation in multi-patch parametric surface meshing, Int. J. Numer. Methods Eng. 62 (4) (2004) 495–515.

[23] A. Tam, D. Ait-Ali-Yahia, M.P. Robichaud, M. Moore, V. Kozel, W.G. Habashi, Anisotropic mesh adaptation for 3D flows on structured and unstructured grids, Comput. Methods Appl. Mech. Eng. 189 (2000) 1205–1230.