ANISOTROPIC ERROR ESTIMATE FOR HIGH-ORDER PARAMETRIC SURFACE MESH GENERATION

Rémi Feuillet[1] Olivier Coulaud[2] Adrien Loseille[1]

[1]*GAMMA Team, INRIA Saclay, 1 Rue Honoré d'Estienne d'Orves, 91120 Palaiseau, France.*
{*remi.feuillet, adrien.loseille*}*@inria.fr*
[2]*Cenaero, Rue des Frères Wright 29, 6041 Gosselies, Belgium.*
olivier.coulaud@cenaero.be

ABSTRACT

Parametric surface mesh generation is one of the crucial step of the computational pipeline. Standard techniques, that are now mature, control the deviation to the tangent plane by using intrinsic quantities as the minimum and maximum curvatures. However, for high-order meshes, deriving intrinsic quantities that have the ability to control the mesh generation process is much more challenging. Indeed, those provided by the first and second fundamental forms of a surface are not sufficient when high order curved meshes are employed. In this paper, we introduce a new set of error estimates for high-order surface mesh generation. It is based on performing a Taylor expansion of the underlying surface in the tangent plane. The independence to the parametric space is obtained by using an inversion formula. High-order terms of this expansion are then used to derive an optimal metric by using the log-simplex approach. Examples are shown to prove the efficiency of the method.

Keywords: Surface mesh generation, High-order error estimates, High-order mesh generation.

INTRODUCTION

In industrial applications, the definition of the computational domain (or of a design) is provided by a continuous description composed by a collection of patches using a CAD (Computer Aided Design) system. If several continuous representations of a patch exist *via* an implicit equation or a solid model, we focus on the boundary representation (BREP). In this description, the topology and the geometry are defined conjointly. For the topological part, a hierarchical description is used from top level topological objects to lower level objects, we have:

model → bodies → faces → loops → edges → nodes

Each entity of upper level is described by a list of entities of lower level. This is represented in Figure 1 for an Onera M6 model, where a face, a loop and corresponding edges are depicted. Note that most of the time, only the topology of a face is provided, the topology between all the faces (patches) needs to be recovered. This piece of information is needed to have a watertight valid surface mesh on output for the whole computational domain. This step makes the surface mesh generation of equal difficulty as volume mesh generation and have been shown to be not trivial [1].

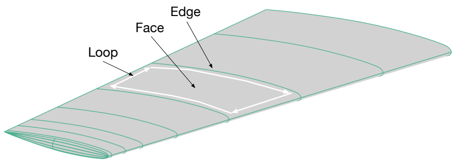


Figure 1: Topology hierarchy (Face, Loop, Edge) of the continuous representation of model using the Boundary REPresentation (BREP).

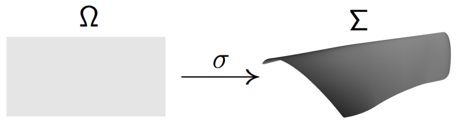


Figure 2: Mapping between the parametric and physical space.

For node, edge, and face, a geometry representation is also associated to the entity. For node, it is generally the position in space, while for edge and face a parametric representation is used. It consists in defining a mapping from a bounded domain of $\mathbb{R}^2$ onto $\mathbb{R}^3$ such that $(x, y, z) = \sigma(u, v)$ where (u, v) are the parameters (Figure 2).

Generally, σ is a NURBS function (Non-uniform rational B-spline) as it is a common tool in geometry modeling and CAD systems [2]. From a conceptual point of view, meshing a parametric surface consists in meshing a 2D domain in the parametric space.
The linear case has been widely studied for years [3, 4, 5, 6, 7, 8, 9, 10] and some approaches consist of meshing the 2D domain according to a curvature-based metric [11]. The use of this metric enables an independence to the used parameters space as the curvature is an intrinsic data. The generation of high-order meshes is on the contrary relatively new. The most common idea is to generate a linear mesh and then to project the high-order nodes on the geometry. However, the position of the nodes may not be suitable for a high-order representation of the boundary. For this reason, optimization procedures are applied to the mesh to improve its shape [12, 13, 14, 15, 16]. The procedure can be done by solving an optimization problem or performing a spring analogy. This can also be performed directly in the parameters space [17] by minimizing a *distortion measure*. In all these cases, this is *r-adaptation* which is performed. The intent of the presented work is to provide tools for high-order parametric surface mesh generation. The use of the high-order estimates developed in [18] will provide a node distribution which will be specifically tailored for the high-order with a given threshold. Note that high order surface meshes have a large set of applications. It can be naturally used as an input for the generation of 3D curved meshes. In our case, high-order surface mesh is used advantageously as a surrogate CAD (geometry) model. Indeed, it provides fast forward and inverse evaluation as required in classic linear mesh adaptation.

The first section briefly recalls the framework used for the high-order error estimates and how a metric can be deduced from it. The second section shows how the standard parametric surface mesh generation can be extended to the high-order using this framework. Finally examples are shown to highlight the process.

1. HIGH ORDER METRIC BASED MESH ADAPTATION

1.1 Continuous mesh framework

In this section, some basic tools concerning the continuous mesh framework are recalled. For further details about these techniques, one refers to [19, 20] and the references therein. The main idea underlying the metric based mesh adaptation is to change the way distances are computed, via a continuous metric field $\mathbf{M} = (\mathcal{M}(x))_{x\in\Omega}$, where for all $x \in \Omega$, the matrix $\mathcal{M}(x)$ is a metric of $\mathbb{R}^n$, namely a symmetric definite positive matrix of $\mathbb{R}^n$. As it is explained below, the notion of unit mesh with respect to a metric field establishes a link between a metric field and a mesh, which reduces the problem of finding an optimal mesh to the problem of finding an optimal metric field. First of all, the scalar product induced by a constant metric $\mathcal{M}$ is given by

$$\langle x, y\rangle_{\mathcal{M}} = {}^t x\,\mathcal{M}\,y, \text{ for all } x, y \in \mathbb{R}^n. \qquad (1)$$

The following distance results from this scalar product.

$$\|x\|_{\mathcal{M}} = \sqrt{{}^t x\,\mathcal{M}\,x}, \text{ for all } x \in \mathbb{R}^n. \qquad (2)$$

Likewise, in the Banach space $\left(\mathbb{R}^n, \|.\|_{\mathcal{M}}\right)$, the distance between two points $a, b \in \mathbb{R}^n$ is given by

$$\ell_{\mathcal{M}}\,(ab) = \sqrt{{}^t ab\,\mathcal{M}\,ab}, \qquad (3)$$

where ab stands for the vector linking a to b. In the same way, if $\mathbf{M} = (\mathcal{M}(x))_{x\in\Omega}$ is a smooth metric field on Ω, the length of the segment between a and b can be computed by using the integral formula

$$\ell_{\mathbf{M}}\,(ab) = \int_0^1 \sqrt{{}^t ab\,\mathcal{M}\,(\gamma(t))\,ab}\,dt, \qquad (4)$$

where $\gamma(t) = (1-t)a + tb$.

As the next definition shows, there is a strong correspondence between continuous metric spaces and meshes, through the notion of unit mesh with respect to a metric field.

Definition 1.1 *A mesh $\mathcal{H}$ of a domain $\Omega \subset \mathbb{R}^n$ is said to be unit with respect to a continuous metric field $\mathbf{M} = (\mathcal{M}(x))_{x\in\Omega}$ if the edges lengths of its elements equal* 1. *If $\{e_1, ...e_{n+1}\}$ are the edges of an element K of $\mathcal{H}$, then*

$$\ell_{\mathbf{M}}(e_i) = 1, \quad \textit{for all} \quad i \in \{1, ..., n+1\}\,.$$

In most of the metric based mesh adaptation methods, the problem is reduced to an optimization problem of finding the optimal metric field for which unit meshes minimize the interpolation error of the solution. In this article, the problem of surface approximation is seen as a 2D high order metric based mesh adaptation in the parameters space.

1.2 Error model

The examined error model is the on studied in [18]. Let u be a smooth solution on the domain Ω, $\mathcal{H}$ be a mesh of Ω and k be an arbitrary positive integer. In what follows, $\Pi_k u$ denotes the projection of u onto the finite elements space $\mathbb{P}^k(\mathcal{H})$, whose functions are polynomials of degree k on each element K of $\mathcal{H}$. For all $x_0 \in \Omega$, it is well known that there exists a positive constant C such that, for all $x_0 \in \Omega$ and $x \in \mathbb{R}^n$,

$$|u(x) - \Pi_k u(x)| \leq C \left| d^{(k+1)} u(x_0)(x - x_0) \right| + \mathcal{O}\left(\|x - x_0\|^{k+2} \right), \tag{5}$$

where $d^{k+1}u(x_0)$ is the differential form of u of order $k+1$ at x_0, $|.|$ is the absolute value function and $\|.\|$ denotes the Euclidean norm of $\mathbb{R}^n$. In the high order case, the main idea is to replace the right hand side of (5) by a term governed by a metric field $\mathbf{Q} = (\mathcal{Q}(x))_{x \in \Omega}$, which approximates the $k+1$ differential form of u. More precisely, we are looking for $\mathbf{Q}$ such that for all $x_0 \in \Omega, x \in \mathbb{R}^n$

$$\left| d^{k+1} u(x_0)(x - x_0) \right| \leq \left({}^t\,(x - x_0)\, \mathcal{Q}(x_0)\, (x - x_0) \right)^{\frac{k+1}{2}} \tag{6}$$

The main issue is to find the metric field $\mathbf{Q}$ such that the inequality (6) is as optimal as possible. From a geometrical point of view, the local problem is to find the largest ellipse in 2D (or the largest ellipsoid in 3D) included into the domain surrounded by the level set of level 1 of $d^{(k+1)}u(x_0)$. Indeed, let $\mathcal{B}_{\mathcal{Q}} = \left\{ x \in \mathbb{R}^n : {}^t x\, \mathcal{Q}\, x \leq 1 \right\}$ be the unit ball of a metric $\mathcal{Q}$, which is an ellipse in 2D (an ellipsoid in 3D). Now assume that, for all $x \in \mathbb{R}^n$ such that $d^{(k+1)}u(x_0)(x) = 1$, one has ${}^t x\, \mathcal{Q}\, x \geq 1$. Let $x \in \mathbb{R}^n$ and $y = \dfrac{x}{|d^{(k+1)}u(x_0)(x)|^{\frac{1}{k+1}}}$. In particular $d^{(k+1)}u(x_0)(y) = 1$, and:

$$d^{(k+1)}u(x_0)(y) \leq {}^t y\, \mathcal{Q}\, y.$$

Since $d^{(k+1)}u(x_0)$ is a homogeneous polynomial of degree $k+1$, it comes

$$1 \leq \frac{{}^t x\, \mathcal{Q}\, x}{|d^{(k+1)}u(x_0)(x)|^{\frac{2}{k+1}}},$$

and consequently

$$\left| d^{(k+1)}u(x_0)(x) \right| \leq \left({}^t x\, \mathcal{Q}\, x \right)^{\frac{k+1}{2}}, \quad \text{for all}\, x \in \mathbb{R}^n.$$

The purpose of the next section is to solve this minimization problem.

1.3 Log-simplex method

The high order mesh adaptation method which we use in this paper is the log-simplex method, introduced in [18]. It can be seen as an extension of the method described in [19, 20] to the high order case. The main difference between the P^1 and the P^k adaptation methods relies on the fact that $\mathbf{Q}$ is directly given by the Hessian matrix of u when dealing with P^1 adaptation, whereas it is mandatory to find a suitable metric field satisfying (6) for the P^k adaptation. The log-simplex algorithm is a way to compute such a metric field. It is based on a sequence of linear problems written in terms of the logarithm matrix $\mathcal{L} = \log(\mathcal{Q})$. In this section, the highlights of this method are recalled. For a full description of the algorithm, see [18].
Given a homogeneous polynomial p of degree $k+1$ on $\mathbb{R}^n$ which stands for $d^{(k+1)}u(x_0)$, a set of points $\{x_1, ..., x_m\}$ of $\mathbb{R}^n$ such that $p(x_i) = 1$ for all $i \in \{1, ..., m\}$ is considered. The optimization problem that the log-simplex method solves is the following.

Find a metric $\mathcal{Q}$ such that

$$\begin{aligned} &\det(\mathcal{Q}) \quad \text{is minimal,} \\ &{}^t x_i\, \mathcal{Q}\, x_i \geq 1, \quad \text{for all} \quad i \in \{1, ..., m\}. \end{aligned} \tag{7}$$

The first line of (7) translates the fact that we are looking for the metric with the largest area (or volume in 3D). Since the cost function of this problem is nonlinear, one rewrites it as a problem in $\mathcal{L} = \log(\mathcal{Q})$. Notice that $\mathcal{L}$ is not a metric but only a symmetric matrix. This formulation also allows the discrete counterpart of the problem to be well-posed. Indeed, in [18], it is shown that the discrete form of (7) is ill-posed. For $\det(\mathcal{Q}) = \exp(\text{trace}(\mathcal{L}))$, a linear cost function is recovered by replacing $\mathcal{Q}$ by $\mathcal{L}$ in (7). On the contrary, the constraints which are linear on $\mathcal{Q}$ become nonlinear when writing them in terms of $\mathcal{L}$. This can lead to really expensive computations. To avoid this problem, the convexity property of the exponential is used and replaces these constraints by approximated linear ones. More precisely, through the classic convexity inequality, if $x \in \mathbb{R}^n$ satisfies ${}^t x\, \mathcal{L}\, x \geq -\|x\|^2 \log\left(\|x\|^2\right)$, it ensures that ${}^t x\, \mathcal{Q}\, x \geq 1$. By this way, the following linear optimization problem is obtained.

Find a symmetric matrix $\mathcal{L}$ such that

$$\begin{aligned} &\text{trace}(\mathcal{L}) \quad \text{is minimal,} \\ &{}^t x_i\, \mathcal{L}\, x_i \geq -\|x_i\|^2 \log\left(\|x_i\|^2\right), \quad \forall i \in \{1, ..., m\}. \end{aligned} \tag{8}$$

Since this problem is linear in $\mathcal{L}$, it can be solved by a simplex method (see for instance [21]). Unfortunately, in most of the cases, solving (8) once does not provide accurate metrics, in the sense that the unit ball of the obtain metric $\mathcal{Q} = \exp(\mathcal{L})$ can be far from the level set of p (see Figure 3).

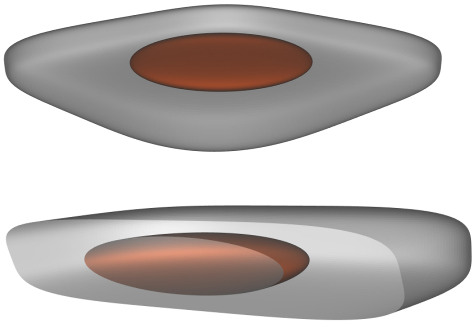


Figure 3: Illustration of the log approximation for the constraints for an error level 1 (in grey). The optimal metric (in red) is far from the boundary of the error due to the convexity approximation.

This issue is dealt by an iterative process. More precisely, once we have computed the solution of (8) and recovered $\mathcal{Q} = \exp(\mathcal{L})$, we apply the mapping $x \to \mathcal{Q}^{\frac{1}{2}}x$ by replacing p by $q = p \circ \mathcal{Q}^{-\frac{1}{2}}$. Then, we take a new set of points $\{x_1, ..., x_m\}$ such that $q(x_i) = 1$, for all $i \in \{1, ..., m\}$ and solve again (8). Finally, the log-simplex algorithm is the following.

```
input  : A mesh $\mathcal{H}$ of $\Omega$
         $d^{(k+1)}u(x)$, for all $x \in \mathcal{H}$

output: $\mathbf{Q} = (\mathcal{Q}(x))_{x \in \mathcal{H}}$

foreach $x \in \mathcal{H}$ do
    repeat
        choose a set of points $\{x_1, ..., x_n\}$ on the
          level set of $p$ of level 1
        perform the log-simplex algorithm and
          obtain a metric $\mathcal{Q}$
        replace $p$ by $p \circ \mathcal{Q}^{-\frac{1}{2}}$
    until convergence;
end
```

In order to implement numerically the log-simplex method, notice that this algorithm must contain a polynomial reduction so that the possible infinite branches in the level set of $d^{(k+1)}u$ disappear. All the theoretical and numerical issues of the log-simplex method are described in details in [18].

The objective of the following section is to find the function u on which the high-order error model should be applied and then to deduce a metric on which mesh adaptation will be performed in the parameters space.

2. HIGH-ORDER SURFACE MESH GENERATION

2.1 Metrics for linear surface mesh generation

In the case of parametric surface meshing, the whole problem is to find a suitable metric thanks to which a mesh adaptation process in the parameters space will be performed. First let us have a look on the case of curve meshing.

2.1.1 Metrics for curve meshing

When dealing with meshing of parametric curves, it is frequent to perform a local analysis on it. To do so, let us have a look on a Taylor expansion of a parametric curve $t \to \gamma(t) \in \mathbb{R}^3$, that we will assume smooth enough, in the vicinity of t_0[1]

$$\gamma(t) = \gamma(t_0) + \gamma'(t_0)(t-t_0) + \frac{\gamma''(t_0)}{2}(t-t_0)^2 + \mathcal{O}((t-t_0)^3).$$

Now if we consider a change of variable with s being the curvilinear abscissa such that $s(t_0) = 0$ and $\frac{ds}{dt} = ||\gamma'(t)||$ then the Taylor expansion becomes:

$$\gamma(s) = \gamma(0) + s\mathbf{T} + \frac{1}{2}\kappa(t_0)s^2\mathbf{N} + \mathcal{O}(s^3)$$

where $\mathbf{T} = \frac{\gamma'(t)}{||\gamma'(t)||}$ and $(\kappa, \mathbf{N})$ are such that $\frac{d\mathbf{T}}{ds} = \kappa\mathbf{N}$. $(\kappa, \mathbf{T}, \mathbf{N})$ are intrinsic data[22]. κ is called the *curvature* and can be computed with:

$$\kappa(t) = \frac{||\gamma'(t) \times \gamma''(t)||}{||\gamma'(t)||^3}.$$

By setting $\mathbf{B} = \mathbf{T} \times \mathbf{N}$, $(\mathbf{T}, \mathbf{N}, \mathbf{B})$ defines an orthornormal basis, of so-called Frénet frame.

Note that if we denote (x, y, z) the components of $\gamma(t)$ in the Frénet frame defined in t_0 with $\gamma(t_0) = (x_0, y_0, z_0)$ in this frame, we have:

$$\begin{cases} y = y_0 + \dfrac{1}{2}\kappa(t_0)(x-x_0)^2 + \mathcal{O}(|x-x_0|^3) \\ z = z_0 + \mathcal{O}(|x-x_0|^3) \end{cases} \tag{9}$$

Based on the Frénet frame, and on the curvature, a 3D metric tensor can be deduced via:

$$\mathcal{M}_1 = \left({}^t\mathbf{T}\,{}^t\mathbf{N}\,{}^t\mathbf{B}\right) \begin{pmatrix} \frac{1}{(2\sqrt{\epsilon(2-\epsilon)}\rho(t))^2} & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix} \begin{pmatrix} \mathbf{T} \\ \mathbf{N} \\ \mathbf{B} \end{pmatrix}$$

where $\lambda \in \mathbb{R}$ is an arbitrary constant[2], $\rho(t) = \frac{1}{\kappa(t)}$ is the radius of curvature, and $2\sqrt{\epsilon(2-\epsilon)}$ is a scaling

[1] In the case of NURBS, it is easy to split the NURBS according the smooth parts [2].

[2] It sets the size in the normal plane to the curve

coefficient which guarantees, for a second order approximation of the curve, to maintain a deviation gap between the mesh elements and the curve geometry of ϵ[23]. As the metric relies on only intrinsic data, it is independent of the parameterization.
The metric can be mapped back to the parameter space via the following formula:

$$\tilde{\mathcal{M}}_1 = {}^t\gamma'(t)\mathcal{M}_1\gamma'(t).$$

In this case, the formula simplifies to $\tilde{\mathcal{M}}_1 = \frac{1}{h_1^2} = \frac{||\gamma'(t)||^2}{(2\sqrt{\epsilon(2-\epsilon)}\rho(t))^2}$.
Once the metrics for curves are set, the next step is to define metrics for the surfaces.

2.1.2 Metrics for surface meshing

The meshing process of parametric surfaces is a bit more complex. It relies on some differential geometry notions [22]. For this purpose, let us consider a parametric surface $(u, v) \to \sigma(u, v) \in \mathbb{R}^3$ that we will assume smooth enough. In this case, the *first fundamental form* $I(du, dv)$ is defined as follows:

$$I(du, dv) = (du\ dv) \begin{pmatrix} ||\sigma_u||^2 & (\sigma_u, \sigma_v) \\ (\sigma_u, \sigma_v) & ||\sigma_v||^2 \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix},$$

where (du, dv) is an elementary displacement, and σ_u (resp. σ_v) the partial derivative of σ w.r.t. u (resp. v). The first fundamental form explains how the three dimensional distances are perceived in the two dimensional space. In particular, it provides a two-dimensional riemannian structure to the surface with a metric tensor defined as:

$$\mathcal{M}_I = \begin{pmatrix} ||\sigma_u||^2 & (\sigma_u, \sigma_v) \\ (\sigma_u, \sigma_v) & ||\sigma_v||^2 \end{pmatrix}.$$

In the same framework, comes also the *second fundamental form* $II(du, dv)$ that is defined as follows:

$$II(du, dv) = \begin{pmatrix} du \\ dv \end{pmatrix} \begin{pmatrix} (\sigma_{uu}, \mathbf{N}) & (\sigma_{uv}, \mathbf{N}) \\ (\sigma_{uv}, \mathbf{N}) & (\sigma_{vv}, \mathbf{N}) \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix},$$

where $(\sigma_{uu}, \sigma_{uv}, \sigma_{vv})$ are the second derivatives of σ w.r.t (u, v) and $\mathbf{N} = \frac{\sigma_u \times \sigma_v}{||\sigma_u \times \sigma_v||}$ is the normal vector to the surface. The second fundamental form expresses the gap of a surface to its tangent plane at the order two.
Based on these two quadratic forms and their matrices, we are able, for a given point of the surface, to define the principal curvatures $(\kappa_i)_{i=1,2}$ and principal directions $(\mathbf{V}_i)_{i=1,2}$ (in 3D) as solution of the generalized eigenvalue problem:

$$\begin{cases} \mathcal{M}_{II}\mathbf{v}_i = \kappa_i\mathcal{M}_I\mathbf{v}_i \\ \mathbf{V}_i = \dfrac{(\sigma_u\ \sigma_v)\,\mathbf{v}_i}{||\,(\sigma_u\ \sigma_v)\,\mathbf{v}_i||} \end{cases} i = 1, 2$$

with $\mathcal{M}_{II}$ being the symmetric matrix associated to the second fundamental form. These quantities are independent of the parameterization and when $\kappa_1 \neq \kappa_2$, $(\mathbf{V}_1, \mathbf{V}_2)$ forms an orthonormal basis of the tangent plane. If we complete the basis with $\mathbf{N}$, they form a local basis $(\mathbf{V}_1, \mathbf{V}_2, \mathbf{N})$ of $\mathbb{R}^3$. Note that if we denote (x, y, z) the components of $\sigma(u, v)$ in this local basis defined in (u_0, v_0) with $\sigma(u_0, v_0) = (x_0, y_0, z_0)$ in this basis, we have:

$$z = z_0 + \frac{1}{2}(\kappa_1(u_0, v_0)(x - x_0)^2 + \kappa_2(u_0, v_0)(y - y_0)^2) + \mathcal{O}(\|(x - x_0, y - y_0)\|^3). \tag{10}$$

Now, thanks to this basis and on the curvatures, we can define the following 3D metric tensor:

$$\mathcal{M}_2 = \left({}^t\mathbf{V}_1{}^t\mathbf{V}_2{}^t\mathbf{N}\right) \times \begin{pmatrix} \frac{1}{(c_1\rho_1(u,v))^2} & 0 & 0 \\ 0 & \frac{1}{(c_2\rho_2(u,v))^2} & 0 \\ 0 & 0 & \lambda \end{pmatrix} \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{N} \end{pmatrix}$$

where $\lambda \in \mathbb{R}$ is an arbitrary constant, $\rho_i(u, v) = \frac{1}{\kappa_i(u,v)}$ for $i = 1, 2$ are the radii of curvature, with the convention $\rho_1(u, v) \leq \rho_2(u, v)$ and $c_1$and c_2 are scaling coefficients. For the direction of greater curvature (*e.g.* the direction given by $\mathbf{V}_1$), we want to control the deviation under a threshold ϵ which comes down to set c_1 to the value of $2\sqrt{\epsilon(2 - \epsilon)}$. Now, as we want the same threshold in all the directions in the tangent plane, the coefficient c_2 is set to $2\sqrt{\epsilon\frac{\rho_1}{\rho_2}(2 - \epsilon\frac{\rho_1}{\rho_2})}$ [24]. Similarly to curves, the metric relies only on intrinsic data and is therefore independent of the parameterization
Now, the metric $\mathcal{M}_2$ can be mapped back to the parameters space by applying the first fundamental form:

$$\tilde{\mathcal{M}}_2 = \left({}^t\sigma_u\ {}^t\sigma_v\right)\mathcal{M}_2 \begin{pmatrix} \sigma_u \\ \sigma_v \end{pmatrix}.$$

This is the metric that will be used as an anisotropic metric for the mesh adaptation in the parameters space.

2.2 Computation of higher-order metrics

The object of this section is to deal with the extension of the previous framework for higher-order elements. In particular, we seek for parameterization independent Taylor expansion similar to (9) and to (10) with terms of degree greater than 2. First let us have a look on the case of the curve.

2.2.1 Case of the curve

As seen previously, the metric should rely on intrinsic data to be independent of the parameterization. A way to do so is to have a look at the formula (9). This formula gives a Taylor expansion of the gap of a curve to the straight edge at the order two and shows that it is driven by the curvature. Moreover, this expansion is done with the physical coordinates which naturally gives an independence with respect to any parameterization. A natural idea is therefore to extend the previous Taylor expansion to get higher-order terms and deduce metrics that will be fitted to high-order approximation.

To do so, let us write $\gamma(t)$ in the Frénet frame $(\mathbf{T}_0, \mathbf{N}_0, \mathbf{B}_0)$ associated to t_0, a regular point of γ. If we note $X = x - x_0, Y = y - y_0, Z = z - z_0$, we have:

$$\begin{cases} X = (\gamma(t) - \gamma(t_0), \mathbf{T}_0), \\ Y = (\gamma(t) - \gamma(t_0), \mathbf{N}_0), \\ Z = (\gamma(t) - \gamma(t_0), \mathbf{B}_0). \end{cases}$$

Let us note $\phi(t) = (\gamma(t) - \gamma(t_0), \mathbf{T}_0)$. If t_0 is a regular point of γ then $\gamma'(t_0) \neq 0$ and therefore $\phi'(t_0) \neq 0$. The inversion function theorem can thus be applied and there exists a function ψ such that $\psi(X) = \psi(\phi(t)) = t - t_0$ in the vicinity of t_0. Moreover, if ϕ is C^{k+1} then ψ is C^{k+1} and $\psi'(X) = \frac{1}{\phi'(t)}$ with $X = \phi(t)$.

Based on this statement, it is thus possible to get a Taylor expansion of $t - t_0$ with respect to X up to order k. To do so, let us compute the higher-order derivatives of ψ in t_0. As γ (and consequently ϕ) is an analytical function issued from CAD model, all its derivatives can be computed using the implementation details of [2]. The derivatives of ψ are then deduced using the following result [25]:

Theorem 2.1 (Faà di Bruno's Formula) *Let us consider $f, g : \mathbb{R} \to \mathbb{R}$ of class C^{k+1} with $k + 1 \geq n$, then*

$$\frac{\mathrm{d}^n}{\mathrm{d}x^n}(g(f(x))) =$$

$$\sum_{E} \frac{n!}{m_1!..m_n!} g^{(m_1+...+m_n)}(f(x)) \prod_{j=1}^{n} \left(\frac{f^{(j)}(x)}{j!}\right)^{m_j},$$

where $E = \{(m_1, .., m_n) \in \mathbb{N}^n | \sum_{k=1}^{n} k.m_k = n\}$[3].

In our case, we set $g = \psi$ and $f = \phi$ and it comes that $\frac{\mathrm{d}^n}{\mathrm{d}x^n}(g(f(x))) = 0$ for $k + 1 \geq n \geq 2$.

This brings us to that for $k + 1 \geq n \geq 2$:

$$(\phi'(t))^n \psi^{(n)}(x) = F(\psi'(x), .., \psi^{(n-1)}(x), \phi'(t), .., \phi^{(n)}(t)),$$

where F is a function that can be directly deduced from Theorem 2.1.

This result means that as long as $\phi'(t) \neq 0$, then $\psi^{(n)}(x)$ can be recursively computed given its previous derivatives and the derivatives of ϕ.

Thanks to this, we can now write:

$$t = t_0 + \sum_{n=1}^{k+1} a_n X^n + \mathcal{O}(|X|^{k+2}),$$

where a_n have been computed with the derivatives of ψ. Now if we recall that γ is C^{k+1}, we also have:

$$\begin{cases} Y = \sum_{n=1}^{k+1} b_n (t - t_0)^n + \mathcal{O}(|t - t_0|^{k+2}), \\ Z = \sum_{n=1}^{k+1} c_n (t - t_0)^n + \mathcal{O}(|t - t_0|^{k+2}). \end{cases}$$

By composition of both Taylor expansion, we then obtain a Taylor expansion of Y and Z in X at the order $k + 1 \geq 2$, which is independent of the parameterization and a generalization of (9).

From this, we have an intrinsic information of the gap of a curve to the straight edge up to the order $k + 1$. We can then write:

$$\begin{pmatrix} Y \\ Z \end{pmatrix} = F_k(X) + A_{k+1} X^{k+1} + \mathcal{O}(|X|^{k+2}),$$

where $F_k(X)$ is a polynomial of degree k in X and $A_{k+1} \in \mathbb{R}^2$. For an approximation of a curve at the degree k, the leading term of the error is therefore $A_{k+1} X^{k+1}$[19, 20, 18]. Thus, if we control $||A_{k+1} X^{k+1}||$, the error of approximation will be controlled. Now, if we note that:

$$||A_{k+1} X^{k+1}|| = (||A_{k+1}||^{\frac{2}{k+1}} X^2)^{\frac{k+1}{2}},$$

we can then set $\kappa_{k+1} = 2||A_{k+1}||^{\frac{2}{k+1}}$ and reuse the metrics used for the linear meshing with $\rho = \frac{1}{\kappa_{k+1}}$ for radii of curvature and $\epsilon_{k+1} = \epsilon^{\frac{2}{k+1}}$ for threshold. This way, the classic formula is found for $k = 1$ and a generalization is proposed for $k \geq 2$. Note that in some configurations, the found size for the order k can be significantly lower than the found size for the order $k + 1$. In this case the size given by order $k + 1$ is preferred.

Now , let us interest to the more complex case of the surfaces.

2.2.2 Case of the surface

Like for curves, the idea is to start from an intrinsic representation of the surface. For this purpose, let us have a look at formula (10). This formula gives a Taylor expansion of the gap of a surface to its tangent plane in physical coordinates which is an intrinsic representation. As previously, our idea is to extend this Taylor expansion to higher-order terms.

To do so, let us note $(\mathbf{V}_{1,0}, \mathbf{V}_{2,0}, \mathbf{N}_0)$ the local basis

[3] Note that in practice, the set E can be precomputed once for all for the range of values of n that are used.

defined for a regular point (u_0, v_0) of the surface. If we note $X = x - x_0, Y = y - y_0, Z = z - z_0$, we have:

$$\begin{cases} X = (\sigma(u,v) - \sigma(u_0,v_0), \mathbf{V}_{1,0}), \\ Y = (\sigma(u,v) - \sigma(u_0,v_0), \mathbf{V}_{2,0}), \\ Z = (\sigma(u,v) - \sigma(u_0,v_0), \mathbf{N}_0). \end{cases}$$

Now, if we define:

$$\Phi(u,v) = \begin{pmatrix} (\sigma(u,v) - \sigma(u_0,v_0), \mathbf{V}_{1,0}) \\ (\sigma(u,v) - \sigma(u_0,v_0), \mathbf{V}_{2,0}) \end{pmatrix},$$

and if (u_0, v_0) is a regular point of σ, then its jacobian matrix J_Φ is invertible in (u_0, v_0). Consequently, the inverse function theorem can be applied and there exists a function Ψ such that $\Psi(X,Y) = \Psi(\Phi(u,v)) = (u - u_0, v - v_0)$ in the vicinity of (u_0, v_0). Moreover, if Φ is C^{k+1} then Ψ is C^{k+1} and $J_\Psi(X,Y) = J_\Phi(u,v)^{-1}$ with $(X,Y) = \Phi(u,v)$.

With this statement, we know that we can have a Taylor expansion of $(u - u_0, v - v_0)$ with respect to (X,Y) up to order $k+1$. For this purpose, let us compute the higher-order derivatives of Ψ. As σ (and therefor Φ) is an analytical function issued from a CAD model, all its derivatives can be computed using the recipes in [2]. The derivatives of Ψ can be then deduced using the following result [26]:

Theorem 2.2 (2D Faà di Bruno's Formula) *Let us consider $f, g : \mathbb{R}^2 \to \mathbb{R}^2$ of class C^{k+1} with $k+1 \geq |\alpha|$, then*

$$\frac{\partial^{|\alpha|}}{\partial x^\alpha}(g(f(x))) = \sum_{|\sigma|=1}^{n} \alpha! \frac{\partial^{|\sigma|}}{\partial x^\sigma}(g(f(x))) \times$$
$$\sum_{E_\sigma} \prod_{i=1}^{2} \prod_{A_\alpha} \frac{1}{e_{i\alpha^i}!} \left(\frac{1}{\alpha^i!} \frac{\partial^{|\alpha^i|}}{\partial x^{\alpha^i}}(f_i(x)) \right)^{e_{i\alpha^i}},$$

where $f(x) = (f_1(x), f_2(x))$, $\alpha = (\alpha_1, \alpha_2) \in \mathbb{N}^2$,$n = |\alpha| = \alpha_1 + \alpha_2$, $\sigma = (\sigma_1, \sigma_2) \in \mathbb{N}^2$

$$E_\sigma = \left\{ (e_{1\alpha^1}, e_{2\alpha^2}) \in \mathbb{N}^2, |\alpha^i| \in [\![1, n]\!], \left(\sum_{|\alpha^i|=1}^{n} e_{i\alpha^i} = \sigma_i, \right)_{i=1,2} \right\},$$

and [4]

$$A_\alpha = \left\{ (\alpha^1, \alpha^2) : |\alpha^i| \in [\![1, n]\!], i = 1, 2, \sum_{i=1}^{2} \sum_{|\alpha^i|=1}^{n} e_{i\alpha^i} . \alpha^i = \alpha \right\}.$$

In our case, we set $g = \Psi$ and $f = \Phi$ and it comes that $\frac{\partial^{|\alpha|}}{\partial x^\alpha}(g(f(x))) = 0$ for $k+1 \geq |\alpha| \geq 2$.

If we consider all the $n+1$ α such that $|\alpha| = n$, then we have a system of equations of the kind:

$$A \left(\frac{\partial^{|\alpha|}\Phi}{\partial x^\alpha} \right)_{|\alpha|=n} \times \left(\frac{\partial^{|\alpha|}\Psi}{\partial x^\alpha} \right)_{|\alpha|=n} =$$
$$F \left(\left(\frac{\partial^{|\alpha|}\Psi}{\partial x^\alpha} \right)_{|\alpha|<n}, \left(\frac{\partial^{|\alpha|}\Phi}{\partial x^\alpha} \right)_{|\alpha|\leq n} \right),$$

where A is $(n+1) \times (n+1)$ matrix, $\left(\frac{\partial^{|\alpha|}\Psi}{\partial x^\alpha} \right)_{|\alpha|=n}$ is a vector of size $n+1$ containing the $n+1$ derivatives of Ψ of order n and F is vector function of size $n+1$ that can be deduced from theorem 2.2. Moreover, it is shown in [26] that $|A| = |J_\Phi|^n$, which proves that the system has always a solution if the inverse function theorem is successfully applied.

This way, a recursive method to compute all the derivative of Ψ in (u_0, v_0) is set and the computation of the Taylor expansion is therefore possible:

$$\begin{pmatrix} u - u_0 \\ v - v_0 \end{pmatrix} = \sum_{n=1}^{k+1} \sum_{i+j=n} A_{ij}^n X^i Y^j + \mathcal{O}(||(X,Y)||^{k+2}),$$

where $A_{ij}^n \in \mathbb{R}^2$ and is defined thanks to the partial derivatives of Ψ.

But, as σ is C^{k+1}, we also have:

$$Z = \sum_{n=1}^{k+1} \sum_{i+j=n} c_{ij}^n (u - u_0)^i (v - v_0)^j$$
$$+ \mathcal{O}(||((u - u_0), (v - v_0))||^{k+2}).$$

By composition of both Taylor expansions, we then obtain a Taylor expansion of Z in (X,Y) at the order $k+1 \geq 2$ which is independent of the parameterization and a generalization of the formula (10).

The gap to the tangent plane is thus expressed up to the order $k+1$:

$$Z = F_k(X,Y) + R_{k+1}(X,Y) + \mathcal{O}(||(X,Y)||^{k+2}), \quad (11)$$

where F_k is a polynomial of degree k and R_{k+1} is an homogeneous polynomial of degree $k+1$. For an approximation of the surface at the degree k, the leading term of the error is $R_{k+1}(X,Y)$ [19, 20, 18]. So, if we want to control the P^k approximation, we need to control $|R_{k+1}(X,Y)|$.

By applying the log-simplex algorithm explained in the previous section, we are able to find a metric that satisfies an inequality like (6), that is to say, we are able to compute a matrix Q_{k+1} such that:

$$|R_{k+1}(X,Y)| \leq \left(\frac{1}{2} (X\ Y) Q_{k+1} \begin{pmatrix} X \\ Y \end{pmatrix} \right)^{\frac{k+1}{2}},$$

where Q_{k+1} is the optimal symmetric matrix (in a sense explained in the first section) that verifies this inequality.

[4] Note that in practice, the sets E_σ and A_α can be pre-computed once for all for the range of values of α and σ that are used.

If we note $(\kappa_{i,k+1}, \mathbf{v}_{i,k+1})_{i=1,2}$, the eigenvalues and eigenvectors of Q_{k+1}, we can then reuse the metrics used for the linear meshing with $\rho_i = \frac{1}{\kappa_{i,k+1}}$ the radii of curvature, $\mathbf{V}_{i,k+1} = (\mathbf{V}_{1,0}\mathbf{V}_{2,0})\,\mathbf{v}_{i,k+1}$ the principal directions in the tangent plane and $\epsilon_{k+1} = \epsilon^{\frac{2}{k+1}}$ the threshold. This way, the classic formula is found for $k = 1$ and a generalisation is proposed for $k \geq 2$. Like for curves, in some configurations, found sizes for the order k can be significantly lower than found sizes for the order $k+1$. In this case sizes given by order $k+1$ are preferred.

2.3 Meshing process

The mesh generation process is based on the classical unit-mesh concept, where a metric field, as $\mathcal{M}_2$ or $\tilde{\mathcal{M}}_2$, is used to drive the orientation and sizing of the elements. In the context of parametric surface meshing, several approaches are typically devised to generate a final 3D surface mesh. Full 2D methods are a convenient way to avoid 3D surface meshing and inverse projection to the geometry. However, a special care is needed to handle degenerated points, periodicity, highly non uniform (even discontinuous) parameterization or degenerated edges. Approaches that mix 2D and 3D methods tend to reduce the impact of the parametric space to the final mesh.

In the paper, we consider a rather classical approach. The core steps of the procedure are decomposed as follows:

1. For each Edge

    1.1 Generate a 3D adaptive mesh using $\mathcal{M}_1$

2. For each Face

    2.1 Generate a fast (u, v)-aligned tessellation,

    2.2 Compute High-order metric $\tilde{\mathcal{M}}_2$ on the tessellation,

    2.3 Project 3D Edges of Loops as parametric curves and generate a 2D (u, v) mesh forming the boundary of the patch,

    2.4 Re-cycle points from the tessellation : insert points from the tessellation onto the current mesh,

    2.4 Move to 3D, convert $\tilde{\mathcal{M}}_2$ to $\mathcal{M}_2$ to adapt the mesh,

    2.5 Generate high-order mesh.

The EGADS API [27] is used to perform the CAD linking and the planar mesh generation process is performed using a Delaunay triangulation-based algorithm[28].

3. EXAMPLES

We illustrate our approach on several examples. To compute error indicates, the computation of an exact Haussdorf distance would be too expansive. Therefore, a normalized and an absolute error in distance are computed. For each triangle of the generated surface mesh, a sampling of the underlying parameters space is performed and the distance between a point of the curved mesh and its surface counterpart is performed:

$$\delta(\tilde{u}, \tilde{v}, \tilde{w}) = ||\sigma(\sum_{i=0}^{n} \phi_i(\tilde{u}, \tilde{v}, \tilde{w})(u_i, v_i)) - \sum_{i=0}^{n} \phi_i(\tilde{u}, \tilde{v}, \tilde{w})\sigma(u_i, v_i)||,$$

where (u_i, v_i) is the set of points of the geometry used to define the high-order triangle ϕ_i the associated shape functions and $(\tilde{u}, \tilde{v}, \tilde{w})$ are the barycentric coordinates of the triangle.
To get the normalized error in distance, the absolute distance is normalized by multiplying it by the local curvature of greater value.
The worst of all these errors then gives the overall errors of the meshing process.
Note that the relative error can be higher than the absolute one. This process can be explained for several reasons. Firstly, the absolute error, is dependent on the system of units. In the case of the shuttle, the average order of magnitude for the bounding box is $10^2 - 10^3$ while it is of $10^0 - 10^1$ in the case of the torus. Secondly, the point of worst value that determines the error is not necessarily the same for both errors. The worst value of the absolute error can occur in high stretching zones where the curvature is high, which emphasizes the value. Thus, it provides levels of error of greater magnitude.
Note that all the figures are obtained using Vizir [29].

Sphere example. Despite its simplicity, the unit sphere example is used to illustrate that our error estimate is independent to the parameterization of the model. A classic boundary representation of a sphere is to consider a surface of revolution, where the two pole maps to two degenerated edges. In the vicinity of these points, depending on the underlying CAD kernel, the definition of normals or more generally the definition of the derivatives of u, v are either undefined or unstable.

We illustrate the obtained Taylor expansion on the point of the sphere using the inversion formula given by (11). Note that the sphere is parameterized as a circle of revolution. far from the pole, we have :

$$\begin{aligned} Z &= -0.5Y^2 - 0.5X^2 - 0.125Y^4 \\ &\quad -0.25X^2Y^2 - 0.125X^4 \\ &\quad -0.0625Y^6 - 0.1875X^2Y^4 \\ &\quad -0.1875X^4Y^2 - 0.0625X^6 \\ &\quad +\mathcal{O}(||(X,Y)||^7). \end{aligned}$$

In the vicinity of the two poles, the expansion is

$$\begin{aligned} Z &= -0.5Y^2 - 0.5X^2 - 0.125Y^4 \\ &\quad -0.25X^2Y^2 - 0.125X^4 \\ &\quad -1.80444e^{-9}XY^4 + 1.9886e^{-9}X^3Y^2 \\ &\quad -0.0625Y^6 - 0.187501X^2Y^4 \\ &\quad -0.187499X^4X^2 - 0.0625X^6 \\ &\quad +\mathcal{O}(||(X,Y)||^7). \end{aligned}$$

We observe that a numerical noise appears while an almost perfect expansion is obtained as in regular points. As expected, the second order terms reveal half the principal curvatures. This numerical noise is a consequence of the manipulation of the huge values at stake in the vicinity of the apex. As the value of the derivative of the parameters is going to 0, the normalization can lead to arbitrary huge values. In the end, everything is simplified but the numerical computations are a bit impacted by this.

Torus. The torus example is based on a NURBS representation is based on two NURBS of degree 5 both defined by 6 control points and 12 knots. P^1, P^2 and P^3 meshes are generated for this case. The point-wise normalized and absolute error in distance are reported in Table 1. The meshes and point-wise errors are reported in Figures 4 and 5.

Order	DOF	Normalized	Absolute
P^1	2787	0.0371489	0.275027
P^2	666	0.0287625	0.18465
P^3	252	0.0137922	0.0883003

Table 1: Deviation to the geometry for torus geometry for P^1 to P^3 meshes.

Also, to show the efficiency of the method on this case, a study has been made between a method where the P^1 metric will be used and ours where a P^k metric is used to generate a meh of degree k. For various absolute error levels, P^1-metric based and P^k-metric based meshes are generated and comparison are done with regards to the number of needed degrees of freedom. In Table 2 (resp 3), a comparison between a P^1-metric and a P^2-metric (resp. P^3-metric) is done. It shows in particular, that for a given error, the number of requested degrees of freedom is higher using the P^1-error than the P^2 (resp. P^3) one. If we have a look at the meshes with the point-wise error in Figure 6 (resp 7), we explain it by the fact that the high-order metric is able to detect high-order features that the P^1-metric does not detect. In particular, we see that the error due to the presence of an annulus on the torus is mostly canceled. And from a more global point a view, the high-order metric provides a mesh with a more equitable distribution of the error with a lower number of degrees of freedom.

Error	P^1-metric #DOF	P^2-metric #DOF
0.5	347	156
0.2	687	242
0.03	1556	1036

Table 2: Comparison between the number of needed degrees of freedom to get a given error between a P^1-metric and a P^2-metric.

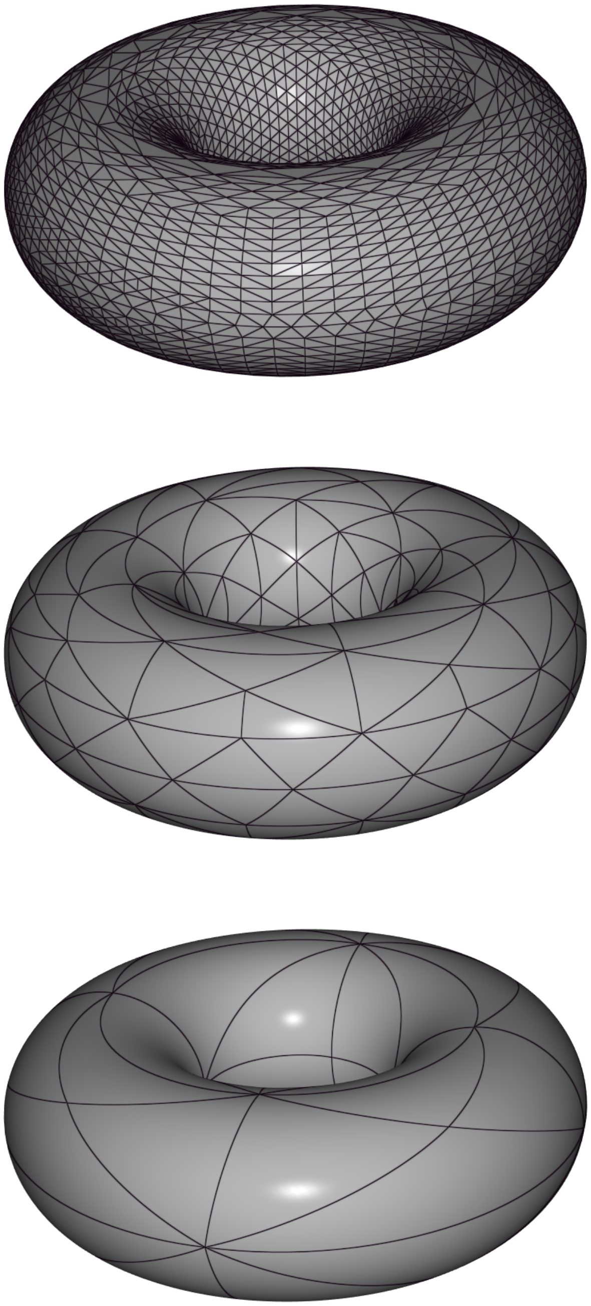
Error	P^1-metric #DOF	P^3-metric #DOF
0.1	255	238
0.002	6171	1848
0.0002	11329	5362

Table 3: Comparison between the number of needed degrees of freedom to get a given error between a P^1-metric and a P^3-metric.

Shuttle. The shuttle geometry is based on two NURBS of degree 3 defined by 8 (resp. 13) control points and 12 (resp. 17) knots with strong variation in the parametric space. The P^1 , P^2 and P^3 meshes are depicted in Figures 8 and 9. The error to the geometry are reported in Table 4.

Order	DOF	Normalized	Absolute
P^1	1647	254.551	197.814
P^2	1690	61.6727	58.3507
P^3	1791	7.02513	22.217

Table 4: Deviation to the geometry for the shuttle geometry.



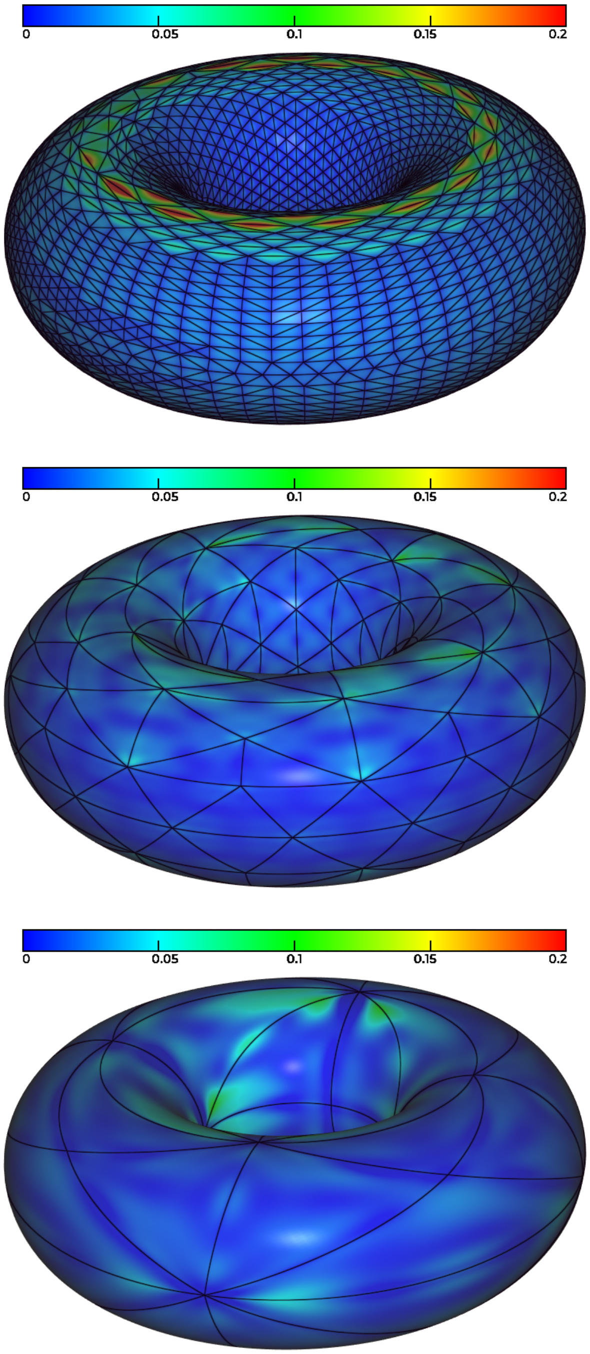


Figure 4: P^1 (top), P^2 (middle) and P^3 (bottom) meshes for the torus geometry.

Figure 5: P^1 (top), P^2 (middle) and P^3 (bottom) point-wise distance to the torus geometry.

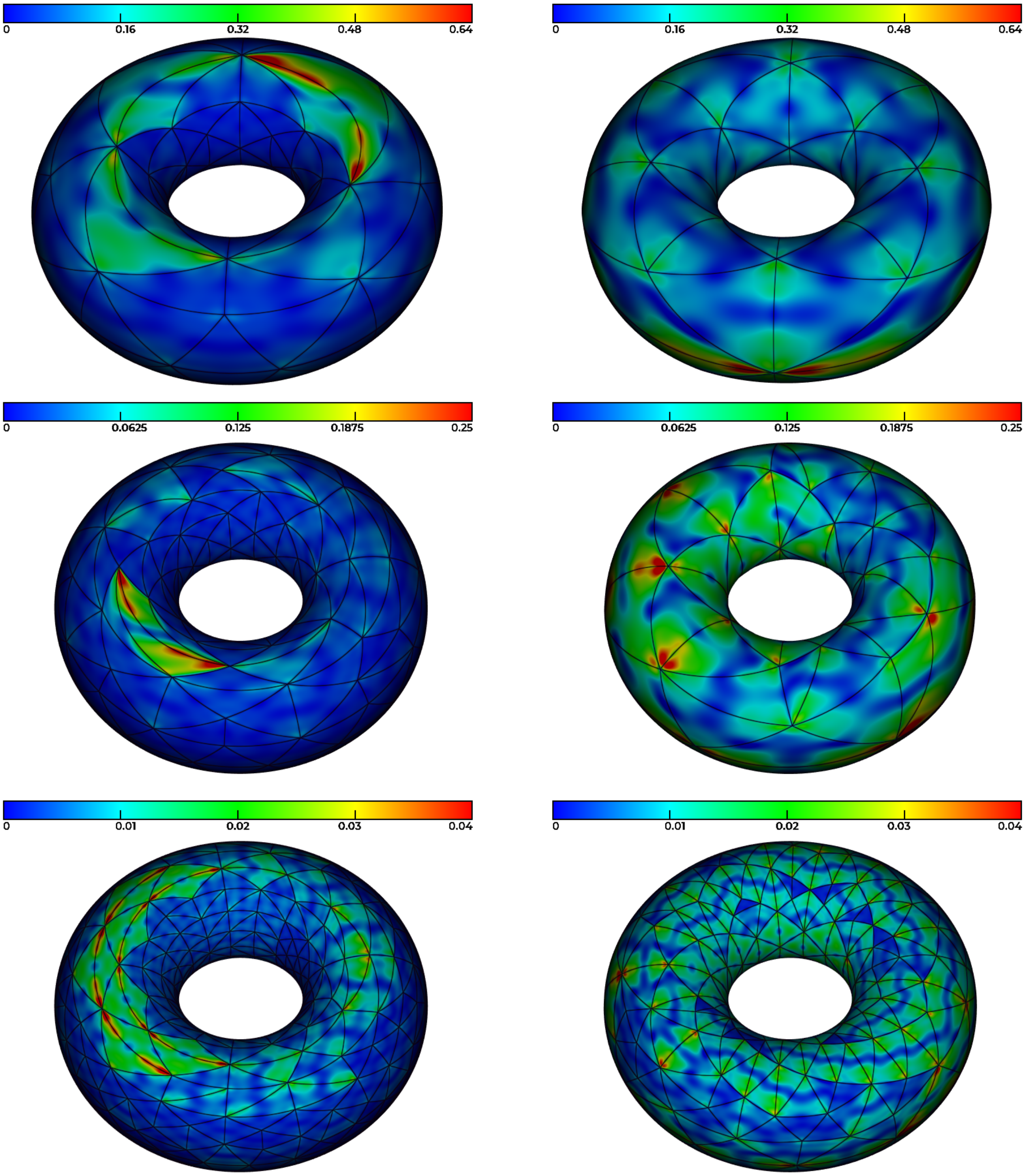


Figure 6: P^2 meshes generated with a P^1-metric (left) and P^2-metric (right) for various error levels: 0.5 (top), 0.2 (middle), 0.03 (bottom)

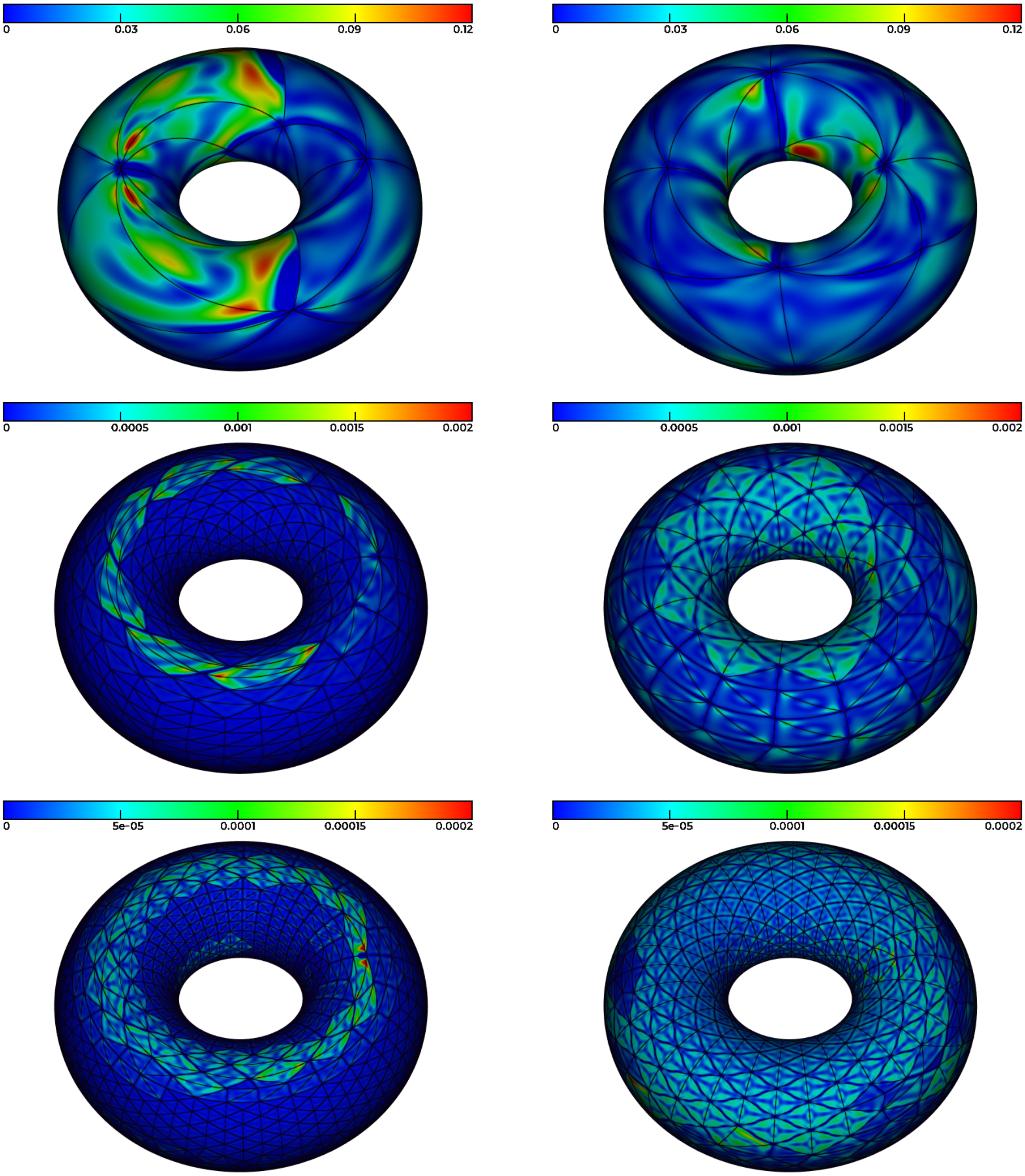


Figure 7: P^3 meshes generated with a P^1-metric (left) and P^3-metric (right) for various error levels: 0.1 (top), 0.002 (middle), 0.0003 (bottom)

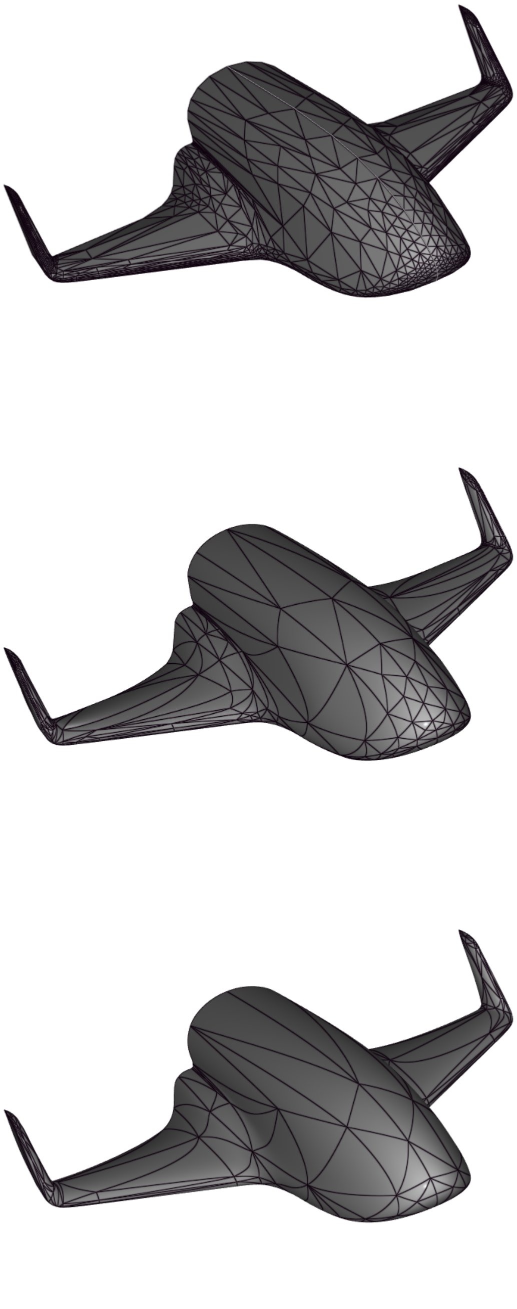


Figure 8: P^1 (top), P^2 (middle) and P^3 (bottom) meshes for the shuttle geometry.

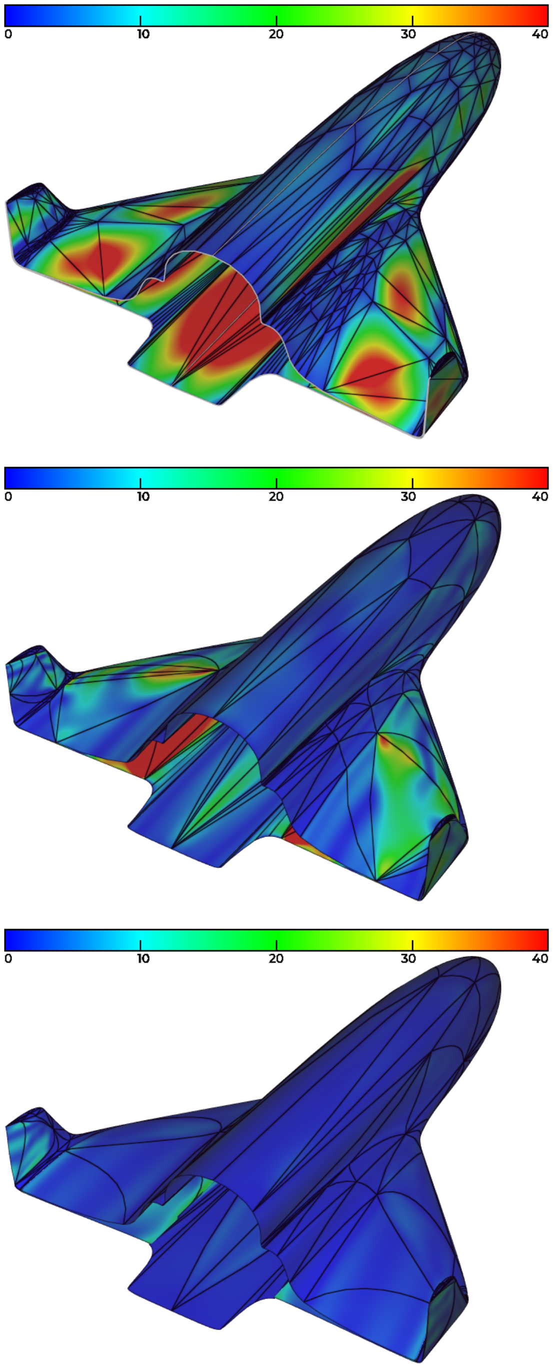


Figure 9: P^1 (top), P^2 (middle) and P^3 (bottom) point-wise distance to the shuttle geometry.

4. CONCLUSION

Generating high-order curved surface meshes from a geometry require the derivation of intrinsic quantities of the surface in order to guarantee the approximation. In the paper, we have used a Taylor expansion coupled with an inversion formula to derive a local approximation of the underlying surface in the Frénet frame. To extend the notion of principal curvature, a log-simplex approach is used to approximate optimally the variation of the polynomial by a quadratic function. The eigenvalues and eigenvectors can be viewed as "high-order" curvatures. As a metric field is naturally derived, these estimates can be used directly in any adaptive anisotropic mesh generation process.

Natural perspectives are possible for this work. Notably, this work could be extended to the implicit surface meshing. Indeed, once the implicit function theorem is applied, it is possible to apply the previous framework in order to get a Taylor expansion in any point of the surface and consequently deduce a metric. The only limitation to this meshing method would be to find the set of points where the surface is evaluated, which is the same problem as in linear meshing of implicit surfaces.

Also, the parametric mesh generation process could be improved by performing metric based curvilinear mesh adaptation to optimize the position of the high-order nodes. This problem is currently under investigation.

ACKNOWLEDGMENTS

This work was supported by the following public grants : *Investissement d'avenir* project, reference ANR-11-LABX-0056-LMH, LabEx LMH and ANR IMPACTS, reference ANR-18-CE46-0003.
The authors are also grateful to Bob Haimes (MIT) for giving some advice about the use of the EGADS API for high-order differentiation.

References

[1] Alleaume A. *Automatic Non-manifold Topology Recovery and Geometry Noise Removal*, pp. 267–279. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009

[2] Piegl L., Tiller W. *The NURBS Book (2nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997

[3] de Cougny H.L., Shephard M.S. "Surface meshing using vertex insertion." *Proceedings of the 5th International Meshing Roundtable*, pp. 243–256. 1996

[4] Tristano J.R., Owen S.J., Canann S.A. "Advancing front surface mesh generation in parametric space using a riemannian surface definition." *Proceedings of the 7th International Meshing Roundtable.* 1998

[5] Miranda A.C., Martha L.F. "Mesh generation on high-curvature surfaces based on a background quadtree structure." *Proceedings of the 11th International Meshing Roundtable.* 2002

[6] Wang D., Hassan O., Morgan K., Weatherill N. "EQSM: An efficient high quality surface grid generation method based on remeshing." *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 41-43, 5621–5633, 2006

[7] Laug P. "Some aspects of parametric surface meshing." *Finite Elements in Analysis and Design*, vol. 46, no. 1–2, 216 – 226, 2010

[8] Siqueira D., Cavalcante-Neto J.B., Vidal C.A., da Silva R.J. "A Hierarchical Adaptive Mesh Generation Strategy for Parametric Surfaces Based on Tree Structures." *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pp. 79–86. IEEE, 2010

[9] de Siqueira D.M.B., Freitas M.O., Cavalcante-Neto J.B., Vidal C.A., da Silva R.J. "An Adaptive Parametric Surface Mesh Generation Method Guided by Curvatures." *Proceedings of the 22nd International Meshing Roundtable.* 2014

[10] Aubry R., Dey S., Mestreau E., Karamete B., Gayman D. "A robust conforming NURBS tessellation for industrial applications based on a mesh generation approach." *Computer-Aided Design*, vol. 63, 26 – 38, 2015

[11] Borouchaki H., George P.L. "Maillage de surfaces paramétriques. Partie I: Aspects théoriques." Research Report RR-2928, INRIA, 1996

[12] Toulorge T., Lambrechts J., Remacle J.F. "Optimizing the geometrical accuracy of curvilinear meshes." *Journal of Computational Physics*, vol. 310, 361 – 380, 2016

[13] Ruiz-Gironès E., Sarrate J., Roca X. "Defining an L^2-disparity Measure to Check and Improve the Geometric Accuracy of Non-interpolating Curved High-order Meshes." *Procedia Engineering*, vol. 124, 122–134, 2015

[14] Ruiz-Gironès E., Roca X., Sarrate J. "High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation." *Computer-Aided Design*, vol. 72, 52 – 64, 2016. 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation

[15] Ruiz-Gironès E., Sarrate J., Roca X. "Generation of curved high-order meshes with optimal quality and geometric accuracy." *Procedia engineering*, vol. 163, 315–327, 2016

[16] Turner M. "High-order mesh generation for CFD solvers." PhD Dissertation Thesis, Imperial College, 2018

[17] Gargallo-Peiró A., Roca X., Peraire J., Sarrate J. "A distortion measure to validate and generate curved high-order meshes on CAD surfaces with independence of parameterization." *International Journal for Numerical Methods in Engineering*, vol. 106, no. 13, 1100–1130, 2016

[18] Coulaud O., Loseille A. "Very High Order Anisotropic Metric-Based Mesh Adaptation in 3D." *Procedia Engineering*, vol. 163, 353–364, 2016

[19] Loseille A., Alauzet F. "Continuous mesh framework part I: well-posed continuous interpolation error." *SIAM J. Numer. Anal.*, vol. 49, no. 1, 38–60, 2011

[20] Loseille A., Alauzet F. "Continuous mesh framework part II: validations and applications." *SIAM J. Numer. Anal.*, vol. 49, no. 1, 61–86, 2011

[21] Dantzig G.B., Thapa M.N. *Linear Programming 2: Theory and Extensions*. Springer-Verlag, 2003

[22] Do Carmo M.P. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016

[23] Frey P.J. "About surface remeshing." *Proceedings of the 9th international meshing roundtable*, 2000

[24] George P.L., Borouchaki H., Alauzet F., Laug P., Loseille A., Maréchal L. *Meshing, Geometric Modeling and Numerical Simulation 2: Metrics, Meshes and Mesh Adaptation*. John Wiley & Sons, 2019

[25] Faà di Bruno F. "Note sur une nouvelle formule de calcul différentiel." *Quarterly J. Pure Appl. Math*, vol. 1, no. 359-360, 12, 1857

[26] Encinas L.H., Masque J.M. "A short proof of the generalized Faà di Bruno's formula." *Applied mathematics letters*, vol. 16, no. 6, 975–979, 2003

[27] Haimes R., Drela M. "On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design." *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2012

[28] George P.L., Borouchaki H. *Delaunay triangulation and meshing*. Hermes, 1998

[29] Loseille A., Feuillet R. "Vizir: High-order mesh and solution visualization using OpenGL 4.0 graphic pipeline." *2018 AIAA Aerospace Sciences Meeting*, pp. AIAA 2018–1174. 2018