Developments on the P^2 cavity operator and Bézier Jacobian correction using the simplex algorithm.

Lucien Rochery, Adrien Loseille

Developments on the P^2 cavity operator and Bézier Jacobian correction using the simplex algorithm.

L. Rochery * and A. Loseille †
Inria GAMMA, Palaiseau, France, 91120

This paper describes developments on the P^2 cavity operator stemming from a new Bézier untangling algorithm. Both surface and volume are adapted to an anisotropic solution field with the cavity operator as the low-level driver handling all topological changes to the mesh. The P^2 extension of the cavity operator handles curvature through Riemannian curved edge length minimization in the volume and geometry projection on the surface. In particular, the anisotropy conserving log-euclidean metric interpolation scheme was extended to high-order elements to facilitate differentiating edge length in the metric field. As a step forward from previous iterations of the P^2 cavity operator, validity is now enforced through optimization of Jacobian coefficients using the simplex algorithm for linear programs. This is made possible by the fact that Jacobian control coefficients are linear with regards to each control point and enables the global optimization of the minimum of all control coefficients surrounding an edge at once. Numerical results illustrate the ability of metric-induced curving to relatively quickly curve 3D meshes with complex geometries involved in Computational Fluid Dynamics (CFD) using only local schemes. This framework allow us to curve highly anisotropic meshes with around 10 million elements within minutes.

I. Introduction

Anisotropic mesh adaptation has established itself as an essential step of efficient numerical simulation, namely for CFD where strongly anisotropic physical phenomena are observed. The first approach, labelled *p*-adaptation, enriches the polynomial space on a per-element basis and, therefore, requires strong coupling with the solver [1]. The second, commonly called *r*-adaptation, limits mesh operations to vertex moving, thus sticking very close to the constraint on mesh complexity [2]. The third, *h*-adaptation, relies on local modifications to adhere to prescribed sizes [3–5] by applying the complete range of meshing operators. These methods have in common that they minimize simulation error at a given number of mesh vertices (mesh complexity). This has been applied to most physical situations, such as the steady [6, 7] and unsteady [8, 9] Euler equations, the steady Navier-Stokes equations in the context of RANS simulation [10, 11], fluid-structure interaction [12], acoustics, electromagnetics, magnetohydrodynamics, solid mechanics [13–16]... This work inscribes itself in the third family of mesh adaptation methods, which resorts to mesh topology changes. In our framework, all these topological changes are handled by the cavity operator [17] in a general formalism that involves deleting a set of elements and staring the boundary of the created boundary against a user-supplied point.

High-order numerical methods for the resolution of Partial Derivatives Equations (PDEs) have been attracting interest over the past years over their ability to solve a wide range of numerical problems with greater precision-over-cost ratio [18, 19]. The need for curved meshes has been established as far back as the 70s with the proof that optimal convergence of high-order methods is only possible with a curved boundary in the case of elliptic problems [20, 21] and later for hyperbolic problems, where physical features are lost when the boundary is left piecewise linear [22]. In some cases, it might even be necessary to represent the geometry with a higher polynomial degree than the solution [23]. This is not to say that volume curvature has no bearing on the numerical solution, as illustrated in [24] where strong influence of interior curvature is shown on finite-element solutions. These curved meshes are polynomial in nature with high-order (hereafter P^k) Bézier elements defined from polynomial mappings on the reference element.

Despite this, high-order computations are still most often carried out on linear meshes, with the robust and automatic construction of valid curved meshes remaining an open problem. One of the main difficulties lies in the fact that element validity criteria become increasingly costly and difficult to control as the polynomial degree of the mesh rises. The other, regarding adaptation specifically, lies in the current lack of general error estimates that would map given solution

*PhD student, Inria GAMMA, lucien.rochery@inria.fr
†Research scientist, Inria GAMMA, adrien.loseille@inria.fr

fields onto optimal element shapes. Without such estimates, meshes with curved interior edges may behave worse than straight meshes [24]. Existing methods rely on input P^1 meshes that are then elevated to a higher degree, preventing back-to-back high-order adaptation. Likewise, the main preoccupation lies in curving the boundary, whereas volume curvature is kept to the minimum that affords validity. Some methods employ *ad-hoc* PDEs [25–29] or a variational approach [30] to displace volume nodes causing invalidity. Others use direct optimization procedures [31, 32] to correct invalid meshes with curved boundary.

Metric fields link particular error estimates with automatic mesh adaptation. Both low-order [4, 5] and high-order metrics [33] have been derived to control interpolation error. Metric-based adjoint estimates allow to control the solution of a chosen PDE [7] or derived quantity of interest such as drag or lift [34]. A metric field locally distorts the measure of distance such that, when the mesh adaptation algorithm has constructed an uniform mesh in the induced Riemannian space, it happens to be strongly anisotropic in the usual Euclidean (physical) space. Therefore, anisotropy appears naturally, without it ever being explicitly sought by the (re)meshing algorithm.

A. Contributions

In previous works [35, 36], we have introduced high-order log-euclidean metric interpolation and used it to devise an efficient Riemannian edge length minimization algorithm. This was done in order to recover curvature from the metric field's variations much in the same fashion P^1 adaptation recovers anisotropy from metric values. This relied on deriving a tractable expression of P^2 edge length in a metric field as a function of the edge Lagrange control node. The main complexity came from the two metric-interpolation steps, namely along the edge once the metrics at the control and end points are known and at the control point from a so-called back-mesh after localization. Indeed, the metric field is not known analytically but rather extended from a discrete field at the vertices of the input mesh by log-euclidean metric interpolation. Previously, we had assumed the back-mesh to be P^1 as we had simple expressions of the barycentric coordinates of the Lagrange control point in the host back-element. Here, we present what has to change for this edge curving method to work with P^2 back-meshes, *i.e.* P^2 input meshes in adaptation.

In parallel, we introduced the P^2 cavity operator and used this edge curving method to initialize new high-order edges. The method used to enforce final cavity validity was very simple, consisting in having the curved edges go to their straight position in a set number of steps. This method tends to limit curvature and only works under the assumption that cavity boundaries are not too curved. Otherwise, it does no better than randomly re-curving edges, as the inner edges being straight are no longer a priviledged configuration. In this paper, we present a new atomic Jacobian correction method. It relies on the fact that Jacobian control coefficients are in fact linear with regards to a single edge control point. Using the simplex algorithm for linear programs, the minimum of all Jacobian control coefficients of all elements surrounding an edge can very quickly be optimized at once. This is rather quick (in the order of 5e-5s per run on a laptop) and provides a global solution. Since it can only be done on one edge at a time (linearity is lost otherwise, as control coefficients have crossed cubic terms), it must be coupled with an iterative procedure to correct all of a cavity or mesh.

Applications shown in previous publications [35, 36] already establish the Riemannian edge length minimization procedure's ability to curve the volume quickly, at about 40000 edges per second on a laptop. Applications shown here therefore focus on illustrating the merits of the simplex-based optimization approach and its ability to conserve a curved surface and volume edges.

B. Bézier elements

We now briefly recall the definition of polynomial (as opposed to rational) Bézier elements as well as introduce the necessary notations. The distinction with rational Bézier elements is dropped as we only deal with polynomial ones. We denote by $\widehat{K} = \{\xi \in [0,1]^3,\ \ \xi_1 + \xi_2 + \xi_3 \leq 1\}$ the reference tetrahedron, and by

$$\widehat{K}^n = \left\{(i,j,k,l) \in \mathbb{N}_+^4,\ \ i+j+k+l = n\right\}$$

the set of indices of degree n Bézier tetrahedra (hereafter degree n tetrahedra or elements). For a given $\alpha \in \widehat{K}^n$, or any tuple for that matter, we denote α_i its components. The Bernstein polynomials are defined by

$$\forall \alpha \in \widehat{K}^n,\ \xi \in \widehat{K},\ B^n_\alpha(\xi) = \binom{n}{\alpha_1}\binom{n-\alpha_1}{\alpha_2}\binom{n-\alpha_1-\alpha_2}{\alpha_3}\xi_1^{\alpha_1}\xi_2^{\alpha_2}\xi_3^{\alpha_3}(1-\xi_1-\xi_2-\xi_3)^{\alpha_4}.$$

and form a basis of degree n polynomials with real coefficients on $\widehat{K}$. Conveniently, they also form the coefficients of a convex combination at any $\xi \in \widehat{K}$:

$$\forall \alpha \in \widehat{K}^n,\ \xi \in \widehat{K},\ 0 \leq B^n_\alpha(\xi) \leq 1,$$
$$\sum_{\alpha \in \widehat{K}^n} B^n_\alpha = 1.$$

This entails that, for any $|\widehat{K}^n|$-tuple of coefficients x,

$$\min x \leq \sum_{\alpha \in \widehat{K}^n} x_\alpha B^n_\alpha \leq \max x.$$

In general, these bounds are not reached on $\widehat{K}$, as only four of the Bernstein polynomials ever reach 1. A degree n tetrahedron K is defined by control points $(P_\alpha)_{\alpha \in \widehat{K}^n}$ and the associated mapping $F_K : \widehat{K} \to K$ given by

$$F_K = \sum_{\alpha \in \widehat{K}^n} B^n_\alpha P_\alpha$$

of which we denote the Jacobian matrix by $\mathcal{J}_K$ and its determinant by J_K. The latter is a trivariate polynomial of degree $q = 3(n-1)$. As such, it can be interpolated exactly by the degree q Bernstein polynomials, such that

$$J_K = \sum_{\alpha \in \widehat{K}^q} N^K_\alpha B^q_\alpha$$

for some coefficients $(N^K_\alpha)_{\alpha \in \widehat{K}^q}$ called the control coefficients of J_K. These coefficients are known as an expression of the control points [37]. Bounds on J_K are therefore given by $\min N^K \leq J_K \leq \max N^K$ but can be further refined by subdividing K or subsets of it (an edge or face). Using this subdivision scheme iteratively yields a sequence of bounds on J_K that converge to its extrema.

One can define high-order elements using Lagrange polynomials instead of the Bernstein. Let us denote by $(\phi^n_\alpha)_{\alpha \in \widehat{K}^n}$ the degree n Lagrange basis. Since F_K is a degree n polynomial, its Lagrange interpolate is exact. Therefore:

$$F_K(\xi) = \sum_{\alpha \in \widehat{K}^n} \phi^n_\alpha(\xi) F_K(\alpha/n) = \sum_{\alpha \in \widehat{K}^n} \phi^n_\alpha(\xi) P^\ell_\alpha \tag{1}$$

where the Lagrange control points P^ℓ_α are defined by $P^\ell_\alpha = \sum_{\beta \in \widehat{K}^n} B^n_\beta(\alpha/n) P_\beta$. This naturally leads to defining the degree n Bézier-to-Lagrange matrix L by $L_{\alpha,\beta} = B^n_\beta(\alpha/n)$. One computes Lagrange control points from Bézier control points by multiplying, coordinate-wise, control points by this matrix. Likewise, one goes from a Lagrange to a Bézier representation with the help of the Lagrange-to-Bézier matrix L^{-1}.

II. The P^2 cavity operator

The original cavity operator is the single operator responsible for all topological changes in our mesh adaptation algorithm. In this section, we present the P^2 cavity operator first introduced in [36] and further developped on in [35]. The main development between the two is in the generalization of high-order log-euclidean metric interpolation since, in the earliest work, metrics on P^2 edges were derived as the P^1 log-euclidean metric interpolate in the P^1 triangle formed by the three edge control points. The generalization was not strictly necessary from a computationnal standpoint as it happens not to change the metric field along P^2 edges but it opens the door to fully P^2 mesh adaptation, *i.e.* with input P^2 meshes. Indeed, before this high-order log-euclidean metric interpolation scheme was devised, there was no consistent way to define the metrics of newly inserted points when the reference mesh was not P^1.

In this section, we briefly present the original cavity operator as well as metric fields and metric interpolation. We then recall the P^2 cavity operator as the next section pertains to an improvement brought to it.

A. The original cavity operator

We now briefly introduce the cavity operator, on which additional information can be found in [17]. It exploits a formalism by which all common topological modifications can be described in similar terms, using:

- An arbitrary collection of elements, boundary faces and ridges, *i.e.* the *cavity*,
- A point in space, to be inserted or reinserted

The principle is that cavity entities are removed from the mesh, *i.e.* the cavity is emptied out (hence the name). These entities may be of pure volume (tetrahedra) or lie on the boundary (boundary faces, ridges which discretize CAD lines). The resulting cavity boundary is then starred against the point to (re)insert in a hierarchical fashion: ridges are first reconstructed, then faces, then elements. This step can fail if created elements turn out to be invalid or if quality criteria are requested and not met. In this cacse, the operator (in its simplest form) rejects the operation and leaves the mesh as it was before deletion of the cavity. In P^1, this step may only fail if a resulting element:

- is of negative volume (i.e. an edge crosses the cavity boundary or lies completely outside of the cavity)
- does not respect general quality criteria (such as the minimal height of the tetrahedra)

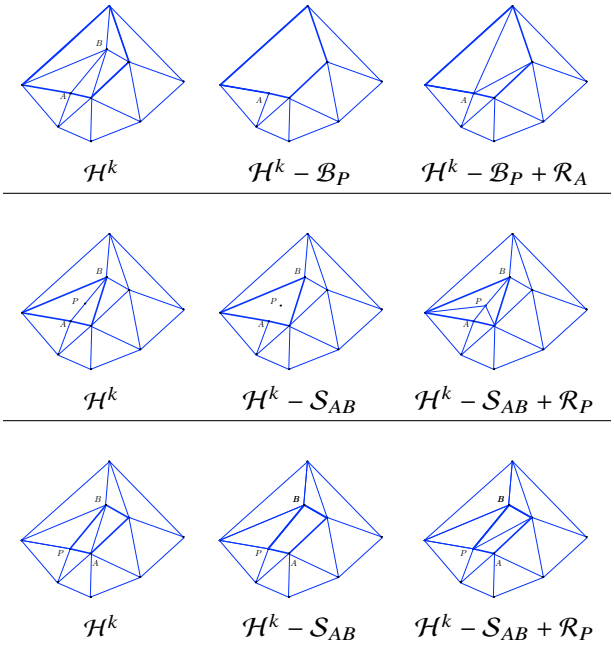

Fig. 1 Mesh operations (collapse, insertion, edge swap, from top to bottom) in cavity terms.

A more sophisticated version of the cavity operator used in practice attempts to circumvent failure by correcting the cavity [38]. For instance, the element outside the cavity against a face that yields a negative volume element is added to the cavity in an attempt to recover validity. This operator is very flexible and, indeed, it is capable of the three most common operations on a mesh, all the while controlling quality and guaranteeing validity at every step:

- Collapse: the cavity is the point's ball, a neighbouring point is reinserted
- Insertion: the cavity is any collection of elements of which one contains the point, typically its local Delaunay cavity, the point itself is inserted
- Swap: the cavity is an edge shell, any point on the boundary of the cavity but not on the edge is reinserted.

These are further illustrated by figure 1 with $2D$ examples. Another feature of the cavity operator is its ability to respect user-supplied element and vertex constraints. Cavity correction can lead to all the elements in the ball of some point to be added to the cavity, leading to that point's deletion. This is avoided by a vertex constraint. Likewise, elements can be tagged as constrained, a feature particularly useful in frontal methods such as for boundary-layer generation.

B. Metric fields and interpolation

In this section, we briefly introduce metrics and metric fields with a focus on metric interpolation in the log-euclidean framework [39]. Denoting by d the space dimension and $\mathcal{S}_d^+$ the set of $d \times d$ symetric positive definite matrices, a metric field is a mapping $\mathcal{M} : \mathbb{R}^d \to \mathcal{S}_d^+$. At any $x \in \mathbb{R}^d$, $\mathcal{M}(x)$ is naturally called a metric and allows the definition of a scalar product $(,)_{\mathcal{M}(x)} : (u, v) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto (x, \mathcal{M}y)$. Metric fields are assumed smooth enough (a condition enforced in practice through metric gradation, for instance) and geometric quantities are defined by integration using the local dot

products. For instance, the length of a path $\gamma : [0,1] \to \mathbb{R}^d$ is given by

$$\ell_{\mathcal{M}}(\gamma) = \int_0^1 \sqrt{\gamma'^T \mathcal{M}(\gamma)\gamma'} dt$$

This quantity is of interest to us as, in the case of P^1 meshes, error estimates depend on the length of element edges in the metric field [4, 5]. Due to the variations of $\mathcal{M}$, the shortest path between two points is not necessarily the straight edge. In the context of mesh adaptation, a metric field is not given as input but rather metrics are given at each mesh vertex. From this sampling of an unknown subjacent metric field, one must reconstruct a smooth field that exhibits good anisotropic conservation properties. The log-euclidean metric interpolation scheme allows just that as well as is cheap to compute. The metric associated to points P_i and weights w_i is given by $\log \mathcal{M}_1 = \sum w_i \log \mathcal{M}(P_i)$, where the log on matrices operates on eigen-values. In the context of mesh adaptation, the points P_i are the vertices of a mesh element K and the weights w_i are the barycentric coordinates of a point P that lies in K. Then, the metric at P is constructed by

$$\mathcal{M}(P) = \sum_{1 \leq i \leq d+1} w_i \log \mathcal{M}(P_i).$$

In [35], we introduced an extension of this scheme on P^1 meshes to high-order meshes, by setting the weights to the Lagrange polynomials applied to the barycentric coordinates rather than the barycentric coordinates themselves. These two naturally coincide in the P^1 case. Then, on a P^n tetrahedron K, and following the notations of the previous section, the high-order log-euclidean metric interpolation scheme becomes

$$\log \mathcal{M}(F_K(\xi)) = \sum_{\alpha \in \widehat{K}^n} \phi^n_\alpha(\xi) \log \mathcal{M}(P^\ell_\alpha)$$

This definition of the metric field in the volume is the basis for our P^2 metric-based mesh adaptation framework, of which the P^2 cavity operator presented in the following is the core. By interpolating $\log \mathcal{M}(F_K)$ in the Bernstein basis, an equivalent formulation becomes:

$$\log \mathcal{M}(F_K(\xi)) = \sum_{\alpha \in \widehat{K}^n} B^n_\alpha(\xi) \log \mathcal{M}_\alpha$$
$$\text{where} \quad \log \mathcal{M}_\alpha = \sum_{\beta \in \widehat{K}^n} L^{-1}_{\alpha,\beta} \log \mathcal{M}(P^\ell_\beta)$$

where the $L^{-1}_{\alpha,\beta}$ are the coefficients of the Lagrange-to-Bézier matrix.

This metric interpolation scheme is crucial in mesh adaptation since the metric field is not known analytically. Instead, metrics are computed at mesh vertices and given as a discrete solution field of s.d.p. matrices. Yet, geometric quantities such as length, volume or anisotropic measures of quality require a continuous description of this metric field. Likewise, when a new point is inserted into the mesh, its metric field must be initialized. This is done by localizing the new point in the so-called back-mesh. Using its barycentric coordinates in the back element the point lies in, a metric is computed by log-euclidean interpolation from the metrics at the element's vertices. This back-mesh is often chosen as simply the input mesh left unchanged throughout one adaptation step. The reason for this is that, despite its better properties, the log-euclidean scheme still dissipates anisotropy. As such, after some iterations of insertions and collapses, the metric field would degenerate into isotropy. Furthermore, the metric field defined this way is not stable by connectivity changes: an edge or face swap would locally change the metric field.

C. Riemannian edge length minimization

We now present our driver for volume edge curvature — the minimization of Bézier edge length in a metric field — in the least possible amount of detail allowing us to extend this framework from P^1 to P^2 back meshes. A more thorough exposition can be found in the previous work [35]. The problem is as follows: given a P^2 edge e of Lagrange nodes P_{20}, P^ℓ_{11} and P_{20} and a metric field $\mathcal{M}$, find the position of the control point P^ℓ_{11} such that the Riemannian length of the edge

$$\ell_{\mathcal{M}}(e) = \int_0^1 \sqrt{{}^t l' \mathcal{M}(l) l'} dt$$

is minimal, where l is the edge mapping:

$$l : t \in [0,1] \mapsto (1-t)(1-2t)P_{20} + 4t(1-t)P^{\ell}_{11} + t(2t-1)P_{02}.$$

In the spirit of sparing computational resources, we have chosen to apply a Newton algorithm to this problem, which requires derivatives of $\ell_{\mathcal{M}}(e)$ with regards to P^{ℓ}_{11}. The integral itself is approximated by quadrature, and the derivatives of l are straightforward to compute. The only difficulty resides in computing those of $\mathcal{M}(l)$. The metric at any point of the edge is interpolated by our high-order log-euclidean interpolation scheme:

$$\mathcal{M}(l(t)) = \exp(C + 4t(1-t)\mathcal{M}(P^{\ell}_{11})),$$

where C contains terms that do not depend on P^{ℓ}_{11}. Assuming the derivatives of the exponential's argument are known, the overall derivative is computed recursively by using the series definition of the matrix exponential. The metric at P^{ℓ}_{11} is computed by first localizing the Lagrange control point in the back mesh, and then applying yet another log-euclidean metric interpolation step. In the previous work, we assumed the back mesh to be a P^1 mesh. We now turn to the case where the back mesh is comprised of P^2 elements. This is necessary as the back mesh is chosen as the input mesh kept unchanged throughout one adaptation step. In the context of P^2 adaptation, there is no guarantee that a valid P^1 mesh can be obtained from the input mesh. Furthermore, information at the control points would have been lost. Therefore, let us now assume that P^{ℓ}_{11} lies in back element K with barycentric coordinates ξ, *i.e.* $P^{\ell}_{11} = F_K(\xi)$. In the case where K was P^1, a closed-form expression of ξ was known, using ratii of volumes. In this case, F_K is a degree 2 polynomial of ξ and F_K^{-1} is not known explicitely, and so

$$\begin{aligned} \xi &= F_K^{-1}(P^{\ell}_{11}) \\ \mathcal{J}(F_K^{-1}) &= \mathcal{J}_K^{-1} \end{aligned}$$

are computed, in the first case, by minimizing $\xi \mapsto \|F_K(\xi) - P^{\ell}_{11}\|_2^2$ and, in the second case, by simply evaluating the Jacobian matrix $\mathcal{J}_K$ of F_K at the solution of the previous step and inversing the matrix.

D. Extension to P^2 meshes

To extend the operator to P^2 meshes, two steps are added. First, linear entities are elevated to their high-order counterparts, creating new control nodes. Second, geometric *criteria* are replaced by high-order ones. For instance, volume positivity is replaced by minimum Jacobian control coefficient positivity as a measure of validity. The difficulty lies in that the new mesh configuration is not unique, but rather indexed on a domain of $\mathbb{R}^{3N}$, N being the number of new interior edges. Indeed, new high-order control points need not be placed so as to form straight edges and such edges may even not guarantee validity given a curved cavity boundary.

The P^2 cavity operator handles curvature in two steps. The first is to curve individual new edges and surface edges or to leave them straight, as requested by the user. The curvature driver is the Riemannian edge length optimization algorith described in [35, 36] in the volume and mid-point CAD projection on the surface. The second step is to correct the new cavity. This is useful whether new edges have been curved or left straight, as the cavity boundary may itself be curved and new elements could be invalid because of this. The P^2 cavity operator is therefore structured as follows:

1) Connect point against cavity boundary, creating high-order nodes
2) Apply requested curvature (optional: Riemmannian edge length in volume, CAD projection on surface)
3) If quality and validity criteria verified, exit.
4) Optimize new elements for validity and quality.
5) If quality or validity criteria not verified, reject operation.

The P^2 cavity operator guarantees validity rather than prescribed curvature. In the worst case, it behaves like the P^1 cavity operator by rejecting all requested curvature in the correction phase. Assuming that maximum vertex ball size is bounded independently of mesh size, these curvature steps add linear overall cost. This hypothesis is common and leads to finite element matrices having $O(N_P)$ non-zero entries, as a point of comparison with some PDE-based global curving methods. Thus, the resulting algorithm should be asymptotically faster than global methods involving a number of matrix-vector product iterations that increases with mesh size. A comparison in [27] evidences a method with iteration count independent of mesh size (the very method presented in the reference) as well as another PDE-based method [25] with a required iteration count that increases with mesh size. In the latter case, overall complexity is more than linear. In the former, asymptotical cost should remain similar as that of the cavity-based approach.

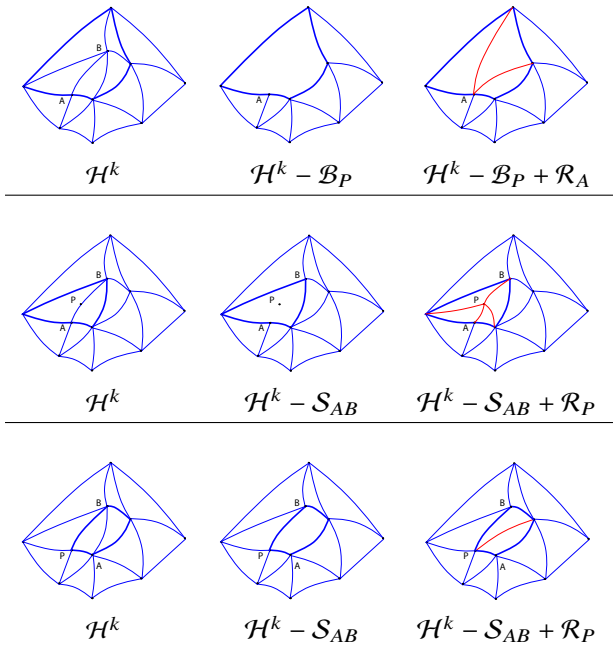

Fig. 2 High-order mesh operations (collapse, insertion, edge swap, from top to bottom) in cavity terms. Volume edges to curve highlighted in red.

This does not mean that our cavity correction step should not be optimized. In the early version of the cavity operator, it was a very simple iterative algorithm bringing each edge to its straight position. We now turn to a more sophisticated algorithm relying on a specific property of the Jacobian of P^k elements.

III. Improvements on Bézier element Jacobian correction

Most methods to generate high-order Bézier meshes take a P^1 mesh as input and proceed in two steps: first curve the boundary by projecting new Lagrange nodes on the geometry, then curve tetrahedra to correct any elements the first step invalidated. Although dealing with CAD systems brings problems of its own, none are particular to generating high-order meshes. The main challenge that arises with high-order meshing is therefore the second step, that of ensuring validity of Bézier tetrahedra. Strictly speaking, validity entails only that the mapping from the reference tetrahedron $\widehat{K}$ to any given element K be invertible. One necessary condition is thus that the Jacobian $\mathcal{J}_K$ of these mappings be invertible all over the reference element. Denoting $J_K = \det \mathcal{J}_K : \widehat{K} \to \mathbb{R}$ the Jacobian determinant, this condition is equivalent to $J_K \neq 0$, itself equivalent by continuity to $J_K > 0$ up to a reordering of element vertices. This leads to global invertibility as long as the element does not self-intersect, which is equivalent to the local condition on the Jacobian in the case of P^2 Bézier elements (likely true for higher degree elements as well), in that no P^2 tetrahedra or triangles with non-zero Jacobian determinant everywhere are self-intersecting. This bare minimum condition can be made more strict by imposing a lower bound on J_K.

The Jacobian determinant is a n-variate polynomial of degree $n(d-1)$, where n is the space dimension and d is the element degree. To our knowledge, suitable methods to compute its minimum exactly either do not exist or were not applied to it; instead, the minimum is either approximated by sampling or by interpolation on a partition of unity basis of polynomials (namely, the Bernstein polynomials). In the latter case, J_K is typically interpolated by the Bernstein polynomials, a case in which the interpolation coefficients are known analytically [37, 40]. This has the advantage of offering a rigorous lower bound on J_K, albeit a conservative one, with known dependency to element geometry.

A number of methods have been devised to deal with negative or small Jacobian determinants. In [32, 41, 42], differential optimization procedures are proposed to minimize functionals depending on the Jacobian and possibly additionnal quality measures.

The first version of the P^2 cavity operator corrected the initialy curved cavity by a very simple relaxation process, with inner edges going to their straight position in a number of steps with new validity verifications. This process works

as long as exterior cavity faces, which the cavity operator is not allowed to modify, are not too curved. In case this step failed to produce a valid cavity, the whole operation was rejected. This algorithm was meant as a placeholder, and its successor is presented in this section. The new cavity correction algorithm works by optimizing the entire edge shells of inner cavity edges all at once. It maximizes minimum Jacobian control coefficient efficiently in one pass using the simplex algorithm for linear programs. The cavity is then corrected by iterating over these edge shells. Naturally, this can be used as a standalone P^2 mesh optimization or correction method as well.

A. The Jacobian as a linear function of a given control point

The first, and crucial, step for setting this method up is to notice a particular structure of the Jacobian determinant of Bézier elements (of any degree). By definition, the Jacobian matrix of a degree n element K is given by

$$\mathcal{J}_K = \left(\sum_{\alpha\in\widehat{K}^n} \partial_{\xi_1} B^n_\alpha P_\alpha \quad \sum_{\alpha\in\widehat{K}^n} \partial_{\xi_2} B^n_\alpha P_\alpha \quad \sum_{\alpha\in\widehat{K}^n} \partial_{\xi_3} B^n_\alpha P_\alpha\right),$$

and its determinant J_K can be writen as

$$J_K = \det \mathcal{J}_K = \sum_{\alpha,\alpha',\alpha''\in\widehat{K}^n} \partial_{\xi_1} B^n_\alpha \partial_{\xi_2} B^n_{\alpha'} \partial_{\xi_3} B^n_{\alpha''} \det\left(P_\alpha P_{\alpha'} P_{\alpha''}\right). \tag{2}$$

Each derivative of a degree n Bernstein polynomial is a degree $n-1$ polynomial. Therefore (2) is a degree $q = 3(n-1)$ polynomial of ξ which we can rewrite

$$J_K = \sum_{\alpha\in\widehat{K}^q} N_\alpha B^q_\alpha \tag{3}$$

where the control coefficients N_α are linear combinations of the terms $\det\left(P_\alpha P_{\alpha'} P_{\alpha''}\right)$:

$$\forall \alpha \in \widehat{K}^q,\ N_\alpha = \sum_{\beta,\beta',\beta''\in\widehat{K}^n} M^{(\beta,\beta',\beta'')}_\alpha \det\left(P_\beta P_{\beta'} P_{\beta''}\right) \tag{4}$$

with some scalar coefficients $M^{(\beta,\beta',\beta'')}_\alpha$ that do not depend on ξ nor any of the control points $(P_{\alpha'})_{\alpha'\in\widehat{K}^n}$. Let us now choose $\alpha_0 \in \widehat{K}^n$ and examine the dependency of the control coefficients N with regards to the control point P_{α_0}. As such, the coefficients N_α are now seen as functions of $P_{\alpha_0} \in \mathbb{R}^3$, although they obviously also depend on the other control points.

The terms of the sum in (4) where none of the 3 indices equal α_0 trivially do not depend upon P_{α_0}: we cluster them into a single term $C^{\alpha_0}_\alpha$, a constant with regards to P_{α_0}. Because of the alternating property of the determinant, the only other non-zero terms are those for which one and only one of the indices β, β' or β'' is equal to α_0, since otherwise two columns under the determinant would be equal. Take one of these terms, assuming $\beta'' = \alpha_0$ without loss of generality, the determinant can be rewriten as

$$\det\left(P_\beta P_{\beta'} P_{\beta''}\right) = x_{\alpha_0} \det\left(P_\beta P_{\beta'} e_1\right) + y_{\alpha_0} \det\left(P_\beta P_{\beta'} e_2\right) + z_{\alpha_0} \det\left(P_\beta P_{\beta'} e_3\right),$$

where $P_{\alpha_0} = x_{\alpha_0} e_1 + y_{\alpha_0} e_2 + z_{\alpha_0} e_3$ and the $(e_i)_{1\leq i\leq 3}$ are basis vectors of $\mathbb{R}_3$. This constitutes a linear function of P_{α_0}. Therefore, these terms of (4) can be clustered into a single term $L^{\alpha_0}_\alpha$ that is a linear function of P_{α_0}. In summary, for all $\alpha \in \widehat{K}^q$, there exist $C^{\alpha_0}_\alpha \in \mathbb{R}$ and $L^{\alpha_0}_\alpha : \mathbb{R}^3 \to \mathbb{R}$ linear such that

$$\forall\ P_{\alpha_0},\ N_\alpha(P_{\alpha_0}) = C^{\alpha_0}_\alpha + L^{\alpha_0}_\alpha(P_{\alpha_0}) \tag{5}$$

This property is crucial. It means that the Jacobian determinant, although a degree 3 polynomial of the control points, does not have cubic nor quadratic terms with repeating variable. This opens the door to linear programming, a field of optimization for which a great deal of work has been done to provide robust and, perhaps most importantly, fast global algorithms. The only restriction is that only one control point (including vertex) can be optimized at once, requiring some iterative procedure be devised. Another restriction appears if we consider the scaled Jacobian instead, which is scaling invariant (therefore a better measure of element distortion) and given by

$$\widetilde{F_K} = F_K/|K|_1 = \sum_{\alpha\in\widehat{K}^q} N_\alpha/|K|_1 B_\alpha$$

where $|K|_1$ is the absolute value of the volume (resp. area) of the P^1 tetrahedron (resp. triangle) defined by the main vertices of K. It follows that linearity of the scaled control coefficients $N_\alpha/|K|_1$ with regards to principal vertices is lost. Since we are interested in enforcing lower bounds on the scaled Jacobian rather than the Jacobian itself, all optimizations will be have to be carried out on edge control points only in the P^2 case or, more generally, on any control points that are not main vertices of the element.

B. Minimum Jacobian maximization as a linear program

We are now in the setting where a control point P_{α_0} is chosen. It belongs to m elements, namely the shell of the edge on which the control point lies, $(K_i)_{1\leq i\leq m}$. Each of these elements K has its own scaled control coefficients $(N^K_\alpha)_{\alpha\in\widehat{K}^q}$ that are seen as functions of P_{α_0}. To simplify, we have not writen the dependency in the volume of the subjacent P^1 element. We denote I the set of indices $\widehat{K}^q \times \{e_1, ..., e_m\}$ where the $(e_i)_{1\leq i\leq m}$ are the canonical basis vectors of $\mathbb{R}^m$. We are left with a set of shell-global control coefficients

$$\forall \alpha \in I,\ \alpha' \in \widehat{K}^q,\ N_\alpha = N^{K_i}_{\alpha'} \text{ iff } \alpha = \alpha' \times e_i$$

The problem we aim to solve is the following

$$\max_{P\in\mathbb{R}^3} \min_{\alpha\in I} N_\alpha(P)$$

which is non linear. To linearize it, we introduce a variable f and solve the equivalent linear problem instead

$$\begin{cases} \max_{f\in\mathbb{R}, P\in\mathbb{R}^3} f \\ \forall \alpha \in I,\ f \leq N_\alpha(P) \end{cases}$$

Since all constraints are linear, as well as the cost function, this is a linear program which can be efficiently solved using, for instance, the simplex algorithm. Indeed, our Fortran implementation is comprised of under a hundred lines of code and typical execution times to optimize an edge shell of P^2 elements takes in the order of 5e-5s on a laptop.

IV. Numerical results

Implementation was carried out in the metric-based mesh adaptation software AMG/feflo.a [43] and visualization on Vizir4 [44, 45]. Full integration remains in progress, preventing full P^2 adaptation. Results provided in previous papers about the P^2 cavity operator [35, 36] have already established the speed of the Riemannian edge length minimization algorithm as well as its ability to curve.

In particular, we have shown to be able to curve over 40000 volume edges per second and, counting the Jacobian correction phase, were able to curve meshes of $20M$ elements in 9min. This was using a very bare-bones untangling procedure, consisting of having edges go to their straight position until Jacobians were made positive. Jacobians were consistently above the decided tolerance. By constructing metric fields from boundary layer elements, we were able to curve boundary layers in a coherent fashion that minimized the amount of straight surface edges. We also illustrated a case of a fully unstructured mesh where the surface could be curved by blending a surface metric field with a small normal size and an isotropic metric field in the volume, and then applying the edge length minimization procedure.

The case presented here is the CRM from the 6th AIAA CFD Drag Prediction Workshop [46] with a mesh adapted using AMG/feflo.a [43] and the Wolf [47] solver. In the first comparison, volume edges are minimized and then the Jacobian is either corrected using the simple relaxation method or the simplex method. Secondly, we compare the results obtained with no volume edge length optimization and both corrections.

Figures 3 and 4 compare executions with and without the simplex algorithm correction and volume curvature. The simplex algorithm can only correct one edge shell at a time. The reason for this is that, as soon as the problem is considered as depending on more than one control point, the cost-function ceases to be linear. Indeed, the Jacobian control coefficients are cubic functions of the control points, with no cubic terms of the same variable (see section III). Therefore, a proper global optimization procedure must be iterative, calling the simplex algorithm as its atomic operation. This is a delicate matter, as the final solution depends on the order in which edges were optimized. Intuition would have it that the worst edges should be optimized first, so that they have the most freedom to improve. But, again, the definition of a bad edge depends on whether we consider only the most negative surrounding control coefficient or a composite function of all negative surrounding control coefficients. Furthermore, tt is very likely that the optimal strategy is a combination of simplex smoothing of worst offenders and Jacobian-based P^2 swaps. For now, we have settled for the following strategy:

1) For each edge, compute ℓ^2 norm of negative control coefficients that depend on edge
2) Sort edges based on this score
3) Loop a set number of iterations, from worst to best edge.

In particular, no re-sorting is done after the first iteration. The ℓ^2 norm is an arbitrary compromise between the average and supremum. Keeping this in mind, the run-times were of 28.5s when the simplex algorithm wasn't used (top figures) versus 44.5s when this strategy was used. In both cases, meshes were valid with Jacobians above the fixed threshold of 1e-6. Qualitatively, the base curvature of the meshes is similar (given by length minimization) with the top meshes (no simplex) being rather straighter. In particular, some coherence of curvature between edges is observed, which is expected from a process that would be capable of recovering the curvature of a solution field. Despite the longer runtimes, this might be an argument in favour of the efficiency of the simplex-based method which owes its current slowness to the global strategy used but wastes less of the volume curvature computed in the first phase. Furthermore, the surface is left straighter when the relaxation method is used (with some outright straight surface edges).

Figures 5 illustrate the results where no volume edges were curved by length minimization but only Jacobian correction. A close view of the reactor shows (bottom) the ability of the simplex-based optimization procedure to correctly keep surface curvature. In the top figure, several straight edges can be seen on the left and top of the reactor which are a consequence of initialy invalid neighbouring volume elements. These elements were succesfuly untangled by the simplex-based algorithm as can be seen in the bottom figure showing a perfectly smooth P^2 surface.

V. Conclusion and future work

High order mesh generation, and adaptation, is one of the key requirements to validate and reach an extreme level of fidelity in complex flow solutions. If many mesh generation techniques exist to curve a linear mesh for a given geometry, less work exists to generate fully adaptive curved meshes. In this paper, we have expanded on the cavity-based framework for anisotropic curved mesh adaptation introduced in [35, 36]. It naturally extends the standard unit-mesh framework used in anisotropic meshes for linear elements. It is based on the idea that a given metric field provides a natural global curvature information that can be used advantageously to curve the mesh everywhere in the domain. The curvature is then not only provided by the geometry but also by the variation of the metric in sizes and orientations. These metric fields account for geometric approximation or some approximation error on the solution.

This high-order mesh generation framework is based on the extension of the linear cavity operator. The main difference is that newly created edges are no longer unique (straight), and their curvature is initialized using the Riemannian edge length minimization algorithm. Furthermore, validity checks no longer involve element volumes but rather their Jacobians which must be ensured positive everywhere. Correcting invalid elements — a procedure sometimes called untangling in the litterature — is a complex process for which no miracle solution has been found. In this paper, we introduced a novel optimization technique based on a property of the Jacobian of Bézier elements that we believe had not been noticed before. Namely, the fact that the Jacobian control coefficients are, when seen as a function of a single control point, linear functions. This enables the use of very efficient algorithms for linear programs, such as the classic simplex algorithm. The minimum of all control coefficients surrounding an edge can be optimized globally in one pass in this manner.

Numerical results focus on illustrating the ability of this new Jacobian optimization procedure's ability to preserve both volume curvature induced by the metric field and surface curvature. Although the global strategy is still being developped, these first results are quite promising.

This paper is a step in developing a fully adaptive curved mesh generation process relying on a faster and more robust P^2 cavity operator. Future work will be directed at implementing all the remaining mesh modification operators such as insertion, collapse and swaps which can be recast within the high-order cavity framework. An additional step will be to extend the local optimization approach to higher order approximation from P^3 to P^5. Other possibilities include extending the log-Euclidean consistent optimization approach to other quantities such as the anisotropic distortion measure. Regarding the Jacobian correction algorithm, more efficient iterative procedures still need to be devised. Namely, a Jacobian-based P^2 generalized swap needs to be implemented.

Acknowledgements

The development of INRIA adapted mesh adaptation AMG/feflo.a is supported by ANR IMPACTS (ANR-18-CE46-0003).

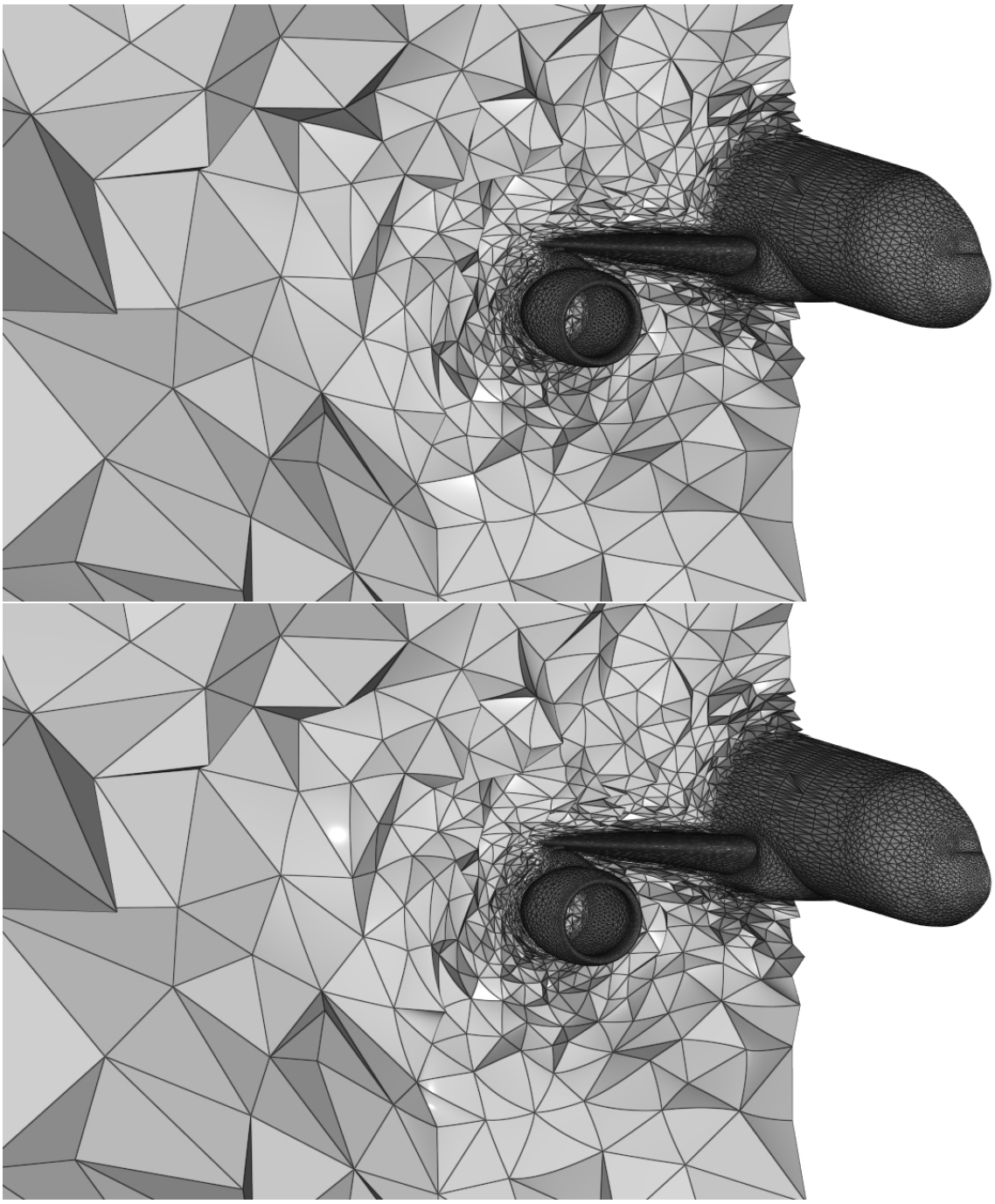

Fig. 3 Front view of the CRM mesh with volume edges optimized for Riemannian length and optimized by relaxation (top) or using the simplex algorithm (bottom). Base curvature is very similar but the top mesh has been straightened out more by the correction.

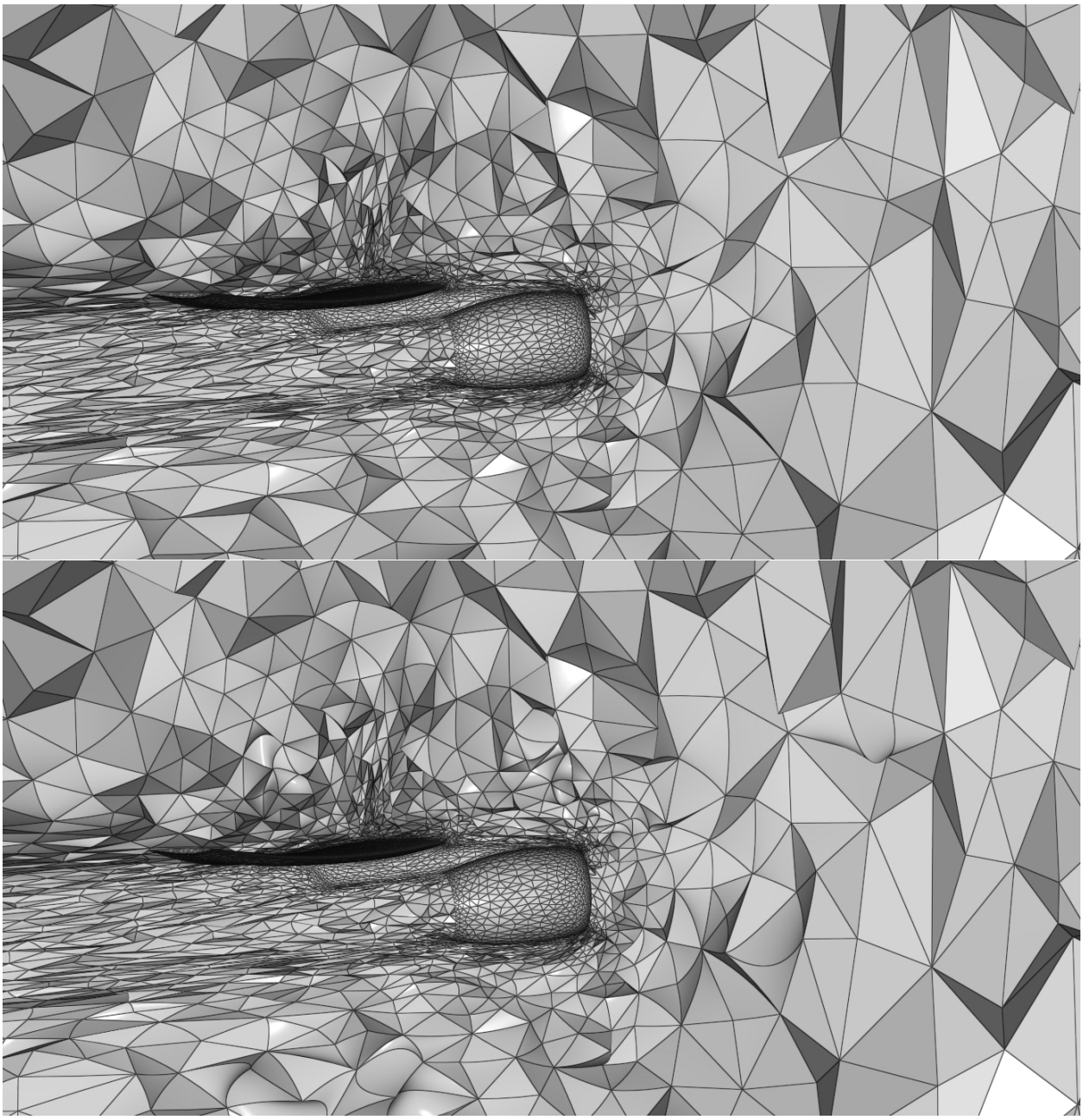

Fig. 4 Reactor side view of the CRM mesh with volume edges optimized for Riemannian length and optimized by relaxation (top) or using the simplex algorithm (bottom). Base curvature is very similar but the top mesh has been straightened out more by the correction.

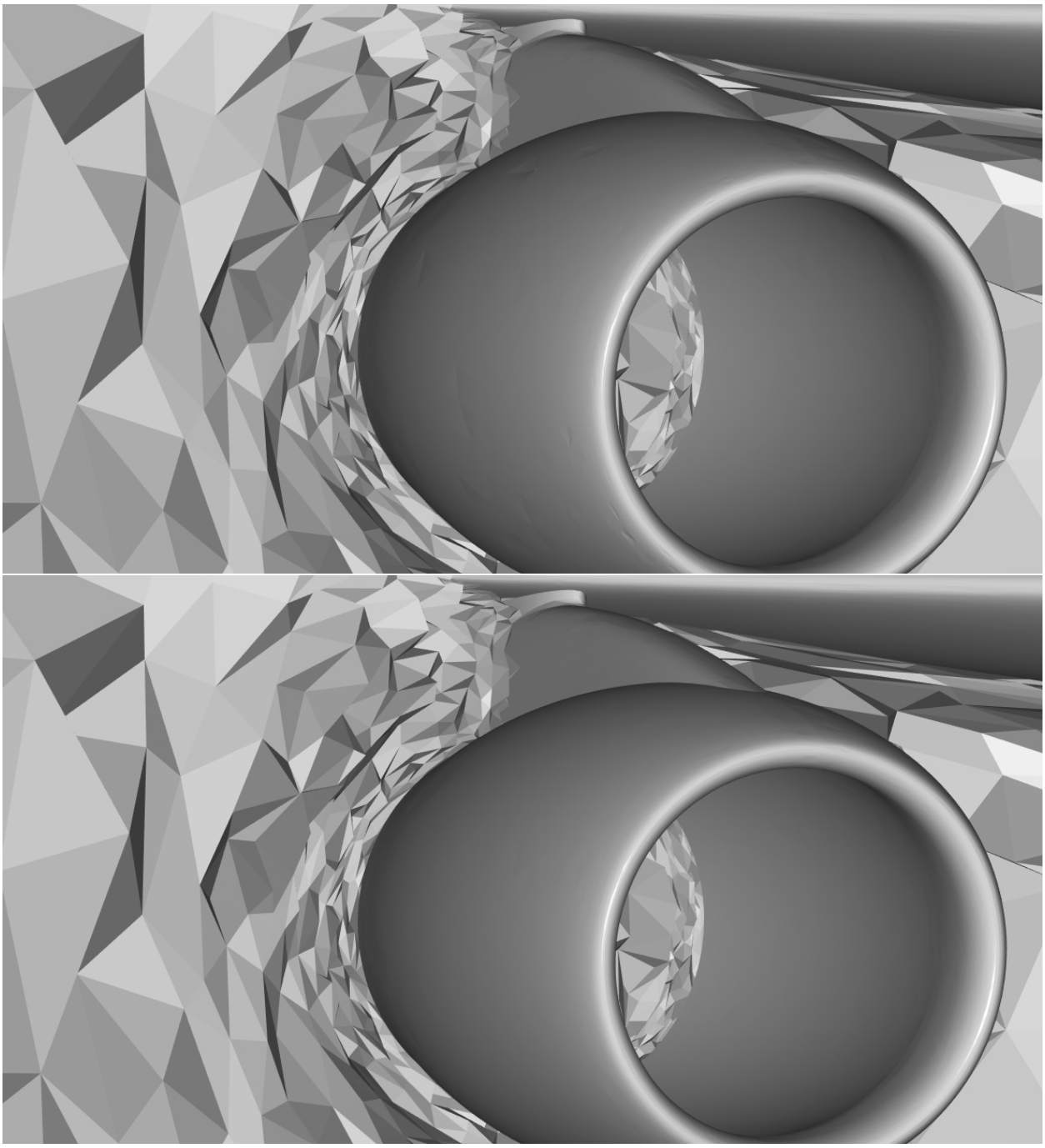

Fig. 5 Close-up on the reactor of the CRM mesh with volume edges optimized for Riemannian length and optimized by relaxation (top) or using the simplex algorithm (bottom). A number of straight faces appear in the top figure whereas the bottom one remains smooth.

References

[1] Huang, W., Kamenski, L., and Lang, J., "A new anisotropic mesh adaptation method based upon hierarchical a posteriori error estimates," *Journal of Computational Physics*, Vol. 229, 2010, pp. 2179–2198.

[2] Babuska, I., Szabo, B. A., and Katz, I. N., "The p-version of the finite element method," *SIAM journal on numerical analysis*, Vol. 18, No. 3, 1981, pp. 515–545.

[3] Berger, M. J., and Oliger, J., "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of computational Physics*, Vol. 53, No. 3, 1984, pp. 484–512.

[4] Loseille, A., and Alauzet, F., "Continuous mesh framework part I: well-posed continuous interpolation error," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 38–60.

[5] Loseille, A., and Alauzet, F., "Continuous mesh framework part II: validations and applications," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 61–86.

[6] Vallet, M.-G., "Génération de maillages éléments finis anisotropes et adaptatifs," Ph.D. Thesis, Université Pierre et Marie Curie - Paris VI, 1992.

[7] Loseille, A., "Anisotropic 3D hessian-based multi-scale and adjoint-based mesh adaptation for Computational fluid dynamics: Application to high fidelity sonic boom prediction." Ph.D. thesis, ????

[8] Alauzet, F., "Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en Mécanique des Fluides," Ph.D. Thesis, Université de Montpellier, Oct. 2003.

[9] Dobrzynski, C., "3D anisotropic mesh adaptation and application for air-conditionning," Ph.D. Thesis, Université Pierre et Marie Curie - Paris VI, Nov. 2005.

[10] Menier, V., "Numerical methods and mesh adaptation for reliable RANS simulations," Ph.D. Thesis, Université Pierre et Marie Curie - Paris VI, Nov. 2015.

[11] Frazza, L., "3D anisotropic mesh adaptation for Reynolds Averaged Navier-Stokes simulations." Ph.D. Thesis, Sorbonne Université , UPMC, Dec. 2018.

[12] Vanharen, J., Feuillet, R., and Alauzet, F., "Mesh adaptation for fluid-structure interaction problems," *AIAA Fluid*, AIAAP 2018-3244, Atlanta, GA, USA, 2018.

[13] Chaillat, S., Groth, S., and Loseille, A., "Metric-based anisotropic mesh adaptation for 3D acoustic boundary element methods ," *Journal of Computational Physics*, Vol. 372, 2018, pp. 473–499.

[14] Borouchaki, H., Grosges, T., and Barchiesi, D., "Improved 3D adaptive remeshing scheme applied in high electromagnetic field gradient computation," *Finite Elements in Analysis and Design*, Vol. 46, No. 1, 2010, pp. 84 – 95.

[15] Amari, T., Canou, A., Aly, J.-J., Delyon, F., and Alauzet, F., "Magnetic cage and rope as the key for solar eruptions," *Nature*, Vol. 554, 2018, pp. 211–215.

[16] Borouchaki, H., Laug, P., Cherouat, A., and Saanouni, K., "Adaptive remeshing in large plastic strain with damage," *International Journal for Numerical Methods in Engineering*, Vol. 63, No. 1, 2005, pp. 1–36.

[17] Loseille, A., "Recent Improvements on Cavity-Based Operators for RANS Mesh Adaptation," *2018 AIAA Aerospace Sciences Meeting*, 2018, p. 0922.

[18] Huerta, A., Angeloski, A., Roca, X., and Peraire, J., "Efficiency of high-order elements for continuous and discontinuous Galerkin methods," *International Journal for numerical methods in Engineering*, Vol. 96, No. 9, 2013, pp. 529–560.

[19] Vanharen, J., "High-order numerical methods for unsteady flows around complex geometries," Ph.D. Thesis, Université de Toulouse, 2017.

[20] Ciarlet, P. G., and Raviart, P.-A., "The combined effect of curved boundaries and numerical integration in isoparametric finite element methods," *The mathematical foundations of the finite element method with applications to partial differential equations*, Elsevier, 1972, pp. 409–474.

[21] Lenoir, M., "Optimal isoparametric finite elements and error estimates for domains involving curved boundaries," *SIAM journal on numerical analysis*, Vol. 23, No. 3, 1986, pp. 562–580.

[22] Bassi, F., and Rebay, S., "High-order accurate discontinuous finite element solution of the 2D Euler equations," *Journal of computational physics*, Vol. 138, No. 2, 1997, pp. 251–285.

[23] Zwanenburg, P., and Nadarajah, S., "On the Necessity of Superparametric Geometry Representation for Discontinuous Galerkin Methods on Domains with Curved Boundaries," *23rd AIAA Computational Fluid Dynamics Conference*, 2017.

[24] Botti, L., "Influence of Reference-to-Physical Frame Mappings on Approximation Properties of Discontinuous Piecewise Polynomial Spaces," *Journal of Scientific Computing*, Vol. 52, 2012. https://doi.org/10.1007/s10915-011-9566-3.

[25] Persson, P.-O., and Peraire, J., "Curved mesh generation and mesh refinement using Lagrangian solid mechanics," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, 2009, p. 949.

[26] Moxey, D., Ekelschot, D., Keskin, Ü., Sherwin, S., and Peirò, J., "High-order curvilinear meshing using a thermo-elastic analogy," *Computer-Aided Design*, Vol. 72, 2016, pp. 130 – 139.

[27] Fortunato, M., and Persson, P.-O., "High-order Unstructured Curved Mesh Generation Using the Winslow Equations," *J. Comput. Phys.*, Vol. 307, 2016, pp. 1–14.

[28] Abgrall, R., Dobrzynski, C., and Froehly, A., "A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems," *International Journal for Numerical Methods in Fluids*, Vol. 76, No. 4, 2014, pp. 246–266.

[29] Hartmann, R., and Leicht, T., "Generation of unstructured curvilinear grids and high-order discontinuous Galerkin discretization applied to a 3D high-lift configuration," *International Journal for Numerical Methods in Fluids*, Vol. 82, No. 6, 2016, pp. 316–333.

[30] Turner, M., Peirò, J., and Moxey, D., "A Variational Framework for High-order Mesh Generation," *Procedia Engineering*, Vol. 163, No. Supplement C, 2016, pp. 340 – 352. 25th International Meshing Roundtable.

[31] Karman, S. L., Erwin, J. T., Glasby, R. S., and Stefanski, D., "High-Order Mesh Curving Using WCN Mesh Optimization," *46th AIAA Fluid Dynamics Conference*, AIAA AVIATION Forum, American Institute of Aeronautics and Astronautics, 2016.

[32] Toulorge, T., Geuzaine, C., Remacle, J.-F., and Lambrechts, J., "Robust untangling of curvilinear meshes," *Journal of Computational Physics*, Vol. 254, 2013, pp. 8 – 26.

[33] "Very High Order Anisotropic Metric-Based Mesh Adaptation in 3D," *Procedia Engineering*, Vol. 163, 2016, pp. 353 – 365. https://doi.org/https://doi.org/10.1016/j.proeng.2016.11.071, 25th International Meshing Roundtable.

[34] "Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations," *Journal of Computational Physics*, Vol. 229, No. 8, 2010, pp. 2866 – 2897. https://doi.org/https://doi.org/10.1016/j.jcp.2009.12.021.

[35] Rochery, L., and Loseille, A., "P2 Cavity Operator with Metric-Based Volume and Surface Curvature," *29th International Meshing Roundtable*, 2021. https://doi.org/10.5281/zenodo.5559055, URL https://doi.org/10.5281/zenodo.5559055.

[36] Rochery, L., and Loseille, A., "P2 cavity operator and Riemannian curved edge length optimization: a path to high-order mesh adaptation," *AIAA Scitech 2021 Forum*, 2021, p. 1781.

[37] George, P.-L., Borouchaki, H., and Barral, N., "Geometric validity (positive jacobian) of high-order Lagrange finite elements, theory and practical guidance," *Engineering with computers*, Vol. 32, No. 3, 2016, pp. 405–424.

[38] George, P. L., "Improvements on Delaunay-based three-dimensional automatic mesh generator," *Finite Elements in Analysis and Design*, Vol. 25, No. 3-4, 1997, pp. 297–317.

[39] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N., "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," *Magnetic Resonance in Medicine*, Vol. 56, No. 2, ????, pp. 411–421. https://doi.org/https://doi.org/10.1002/mrm.20965.

[40] Johnen, A., Remacle, J.-F., and Geuzaine, C., "Geometrical validity of curvilinear finite elements," *Journal of Computational Physics*, Vol. 233, 2013, pp. 359–372.

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

[42] Gargallo-Peiró, A., Roca, X., Peraire, J., and Sarrate, J., "Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes," *International Journal for Numerical Methods in Engineering*, Vol. 103, No. 5, 2015, pp. 342–363.

[43] "Chapter 10 - Unstructured Mesh Generation and Adaptation," *Handbook of Numerical Methods for Hyperbolic Problems*, Handbook of Numerical Analysis, Vol. 18, edited by R. Abgrall and C.-W. Shu, Elsevier, 2017, pp. 263 – 302. https://doi.org/https://doi.org/10.1016/bs.hna.2016.10.004.

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

[45] "On pixel-exact rendering for high-order mesh and solution," *Journal of Computational Physics*, Vol. 424, 2021, p. 109860. https://doi.org/https://doi.org/10.1016/j.jcp.2020.109860.

[46] Vassberg, J., Dehaan, M., Rivers, M., and Wahls, R., "Development of a Common Research Model for Applied CFD Validation Studies," *26th AIAA Applied Aerodynamics Conference*, 2008. https://doi.org/10.2514/6.2008-6919.

[47] Alauzet, F., and Frazza, L., "3D RANS anisotropic mesh adaptation on the high-lift version of NASA's Common Research Model (HL-CRM)," *AIAA Aviation 2019 Forum*, 2019, p. 2947.