Fast practical untangling of simplicial P2 and P3 curvilinear meshes

Guillaume Coiffier * Amaury Johnen * Jean-François Remacle *

Abstract

The aim of this article is to couple the ideas proposed by [8, 6] and [4] to unfold/untangle high-order meshes. What is proposed here is to systematically reduce the untangling of any high-order elements to the problem of untangling simplices (triangles in 2D and tetrahedra in 3D). First, we present a general way of expressing the validity of a high-order element by calculating linear combinations of areas of well-chosen simplices. We then show how to adapt[4] to these linear combinations of simplices. Examples of 2D boundary layer untangling are presented with P2 and P3 elements. The algorithm is then adapted to P2 tetrahedra.

1 Introduction

A decade ago, robust estimators for validating high-order meshes were proposed [8]. The idea is that the space-varying determinant of the Jacobian matrix $|\mathbf{J}|$ of an element can be expressed in bases of positive Bézier-type functions. If the coefficients of $|\mathbf{J}|$ are all positive as well, then this is a sufficient condition to demonstrate that the mapping is injective and that the high-order element is therefore valid. In fact, it is possible to interpret these Bézier coefficients geometrically as areas of triangles or linear combinations of areas of triangles whose vertices are the corners of the Bézier polygon of the element [6]. Our idea is to use what is proposed in [4] to ensure the positivity of these Bézier coefficients. This makes for an optimization algorithm that is easy to implement and parallelize at the expense of slightly less deformation freedom. We present three case studies: P2 and P3 triangles and P2 tetrahedra. General formulas are given for simplices of arbitrary order, and the method can be generalized to quads and hexes of any order in a straightforward way.

2 Related work

The problem of ensuring the validity of a high-order mesh has been well studied in the last decade. A common approach consists of first generating a straight first-order mesh, before adding degrees of freedom and deforming the mesh into a high-order one. Quality metrics for deformation include signed corner angles [10], the Winslow functional [3], elasticity-based energies [2, 1] or various quantities built from an element's Jacobian determinant, like its $\mathcal{L}^2$-norm [5] or a bound on its minimum computed recursively [11]. But to our knowledge, no method combine the three following properties: optimizing for a necessary and sufficient validity condition, being computable without quadrature, subdivision or approximation, and requiring a single continuous optimization pass. In our case, we relax the first property and only consider a subset of all valid elements that satisfy simple *sufficient, but not necessary* conditions.

*Institute of Mechanics, Materials and Civil Engineering, Université catholique de Louvain, Belgium

3 Bézier triangles

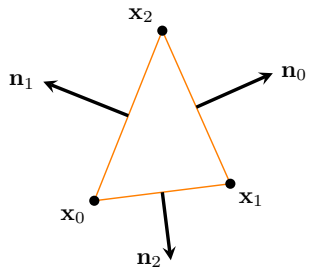

Figure 1: A straight-sided triangle and its edge normals.

3.1 Barycentric coordinates. Consider the linear triangle $T_{012} = \{\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2\}$ (see Figure 1). Any point $\mathbf{x}$ of T_{012} corresponds to one given triplet of barycentric coordinates λ_i

$$\mathbf{x} = \lambda_0\mathbf{x}_0 + \lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_2, \tag{3.1}$$

with $\lambda_0+\lambda_1+\lambda_2 = 1$. The (signed) area of T_{012} is found computing the box product:

$$A = \frac{1}{2}\big[(\mathbf{x}_1 - \mathbf{x}_0) \times (\mathbf{x}_2 - \mathbf{x}_0)\big] \cdot \mathbf{e}_z.$$

The Jacobian determinant, $J = |\mathbf{J}|$ can be obtained from the barycentric expression of $\mathbf{x}$ (Eq. (3.1)) provided that we express λ_0 as a function of the two others coordinates such that λ_1 and λ_2 become two indepen-

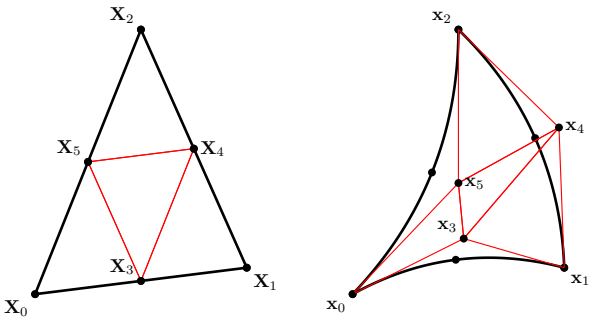

Figure 2: A straight-sided triangle (left) that is the ideal triangle for a second order triangle (right). Red triangles are Bézier triangles.

dent variables.

$$J = \left(\frac{\partial \mathbf{x}}{\partial \lambda_1} \times \frac{\partial \mathbf{x}}{\partial \lambda_2} \right) \cdot \mathbf{e}_z, \quad \lambda_0 = 1 - \lambda_1 - \lambda_2. \tag{3.2}$$

Computing the partial derivatives leads to the well-known result that the Jacobian determinant of a linear triangle is simply two times its area.

In what follows, it will be useful to calculate the derivative of this area with respect to the motion of, say, point $\mathbf{x}_1$ of T_{012}. We have

$$\frac{\partial A}{\partial \mathbf{x}_1} = \frac{1}{2}(\mathbf{x}_2 - \mathbf{x}_0) \times \mathbf{e}_z = \mathbf{n}_1. \tag{3.3}$$

Here, $\mathbf{n}_1$ is a vector orthogonal to $\mathbf{x}_2 - \mathbf{x}_0$ and of half the size. Using finite element vocabulary, $\mathbf{n}_1$ is the gradient of Lagrange shape function associated to $\mathbf{x}_1$.

The barycentric mapping of Eq. (3.1) can be generalized to order p with the Bézier triangular polynomials [8]:

$$\mathcal{T}^{(p)}_{i,j,k}(\lambda_0, \lambda_1, \lambda_2) = \frac{p!}{i!j!k!}\, \lambda_0^i\, \lambda_1^j\, \lambda_2^k, \quad i + j + k = p.$$

To each $\mathcal{T}^{(p)}_{i,j,k}$ is associated a point $\mathbf{x}_{ijk}$ such that the mapping of the element is given by

$$\mathbf{x}(\lambda_0, \lambda_1, \lambda_2) = \sum_{i,j,k}^{p} \mathcal{T}^{(p)}_{i,j,k}\, \mathbf{x}_{ijk}, \tag{3.4}$$

with the convention that $\sum^p_{i,j,k} = \sum_{i,j,k \geq 0,\ i+j+k=p}$. There is a relation between the barycentric indices and the index convention that we use in this paper. For example, point $\mathbf{x}_{p00}$ is point $\mathbf{x}_1$ and $\mathbf{x}_{0p0}$ is point $\mathbf{x}_2$.

3.2 Quadratic Bézier triangles. Let us first consider the case of the quadratic triangle. Using notations of Figure 2, the general interpolation formula of Equation (3.4) can be instantiated as:

$$\begin{aligned} \mathbf{x}(\lambda_0, \lambda_1, \lambda_2) = {} & \lambda_0^2\, \mathbf{x}_0 + 2\lambda_0\lambda_1\, \mathbf{x}_3 + \\ & \lambda_1^2\, \mathbf{x}_1 + 2\lambda_1\lambda_2\, \mathbf{x}_4 + \\ & \lambda_2^2\, \mathbf{x}_2 + 2\lambda_2\lambda_0\, \mathbf{x}_5. \end{aligned} \tag{3.5}$$

By setting $\lambda_0 = 1 - \lambda_1 + \lambda_2$, we can compute the partial derivatives of Equation (3.2):

$$\begin{aligned} \frac{\partial \mathbf{x}}{\partial \lambda_1} &= 2\,(\mathbf{y}_1 - \mathbf{y}_0) \\ \frac{\partial \mathbf{x}}{\partial \lambda_2} &= 2\,(\mathbf{y}_2 - \mathbf{y}_0) \end{aligned}$$

where we have introduced three new points $\mathbf{y}_0$, $\mathbf{y}_1$, $\mathbf{y}_2$ that are located inside triangles T_{035}, T_{314} and T_{542} respectively:

$$\begin{aligned} \mathbf{y}_0 &= \lambda_0\mathbf{x}_0 + \lambda_1\mathbf{x}_3 + \lambda_2\mathbf{x}_5 \\ \mathbf{y}_1 &= \lambda_0\mathbf{x}_3 + \lambda_1\mathbf{x}_1 + \lambda_2\mathbf{x}_4 \\ \mathbf{y}_2 &= \lambda_0\mathbf{x}_5 + \lambda_1\mathbf{x}_4 + \lambda_2\mathbf{x}_2. \end{aligned}$$

The Jacobian determinant of the P2 triangle at any point $\mathbf{x}(\lambda_0, \lambda_1, \lambda_2)$ is thus equal to

$$\boxed{J(\lambda_0, \lambda_1, \lambda_2) = 8A_{\mathbf{y}_0\,\mathbf{y}_1\,\mathbf{y}_2}} \tag{3.6}$$

where $A_{\mathbf{y}_0\,\mathbf{y}_1\,\mathbf{y}_2}$ is the area of triangle formed by points $\mathbf{y}_0, \mathbf{y}_1$ and $\mathbf{y}_2$. On the other hand, the Jacobian determinant of a P2 triangle is a P2 function [8], and thus, similarly to Equation 3.5, we can write:

$$\begin{aligned} J(\lambda_0, \lambda_1, \lambda_2) = {} & \lambda_0^2\, J_0 + 2\lambda_0\lambda_1\, J_3 + \\ & \lambda_1^2\, J_1 + 2\lambda_1\lambda_2\, J_4 + \\ & \lambda_2^2\, J_2 + 2\lambda_2\lambda_0\, J_5. \end{aligned} \tag{3.7}$$

Those two last equations allow to establish the relation between the coefficients of the Jacobian determinant J_i and triangles areas. Now, a sufficient condition for $J > 0$ is that all Bézier coefficients $J_i > 0$, $i = 0, \ldots, 5$ in Equation (3.7) are positive.

For coefficients corresponding to triangle corners ($i = 0, 1, 2$), we have

$$\begin{aligned} J(1,0,0) &= J_0 = 8A_{035} > 0, \\ J(0,1,0) &= J_1 = 8A_{143} > 0, \\ J(0,0,1) &= J_2 = 8A_{254} > 0. \end{aligned}$$

meaning that all three *corner jacobians* should be strictly positive.

For coefficient at the middle of edges ($i = 3, 4, 5$), the computation is slightly more complicated. Taking the case of $i = 3$, Equation 3.7 allows us to write:

$$J_3 = 2\,J\left(\frac{1}{2},\frac{1}{2},0\right) - \frac{1}{2}\left(J_0 + J_1\right),$$

while substituting $\lambda_0 = \lambda_1 = \frac{1}{2}$ in Equation 3.6 yields:

$$\begin{aligned}\frac{1}{2}\,J\left(\frac{1}{2},\frac{1}{2},0\right) &= 4\left[\frac{\mathbf{x}_1-\mathbf{x}_0}{2}\times\left(\frac{\mathbf{x}_5+\mathbf{x}_4}{2}-\frac{\mathbf{x}_3+\mathbf{x}_0}{2}\right)\right]\\ &= (\mathbf{x}_1-\mathbf{x}_0)\times(\mathbf{x}_5-\mathbf{x}_0)+\\ &\quad (\mathbf{x}_1-\mathbf{x}_0)\times(\mathbf{x}_4-\mathbf{x}_3)\\ &= A_{015}+A_{014}-A_{013}.\end{aligned}$$

Combining the two, we obtain

$$\begin{aligned}\frac{1}{4}\,J_3 &= A_{015}+A_{014}-A_{013}-A_{035}-A_{143}\\ &= A_{034}+A_{153}+A_{013}\end{aligned}$$

A similar computation gives expressions for J_4 and J_5.

To summarize, there exist six sufficient conditions for the P2 element to be valid [8] and those conditions can be interpreted geometrically using *area of triangles whose corners are Bézier points*:

$$\begin{aligned}&J_0>0 \;\rightarrow\; A_{035}>0,\\ &J_1>0 \;\rightarrow\; A_{143}>0,\\ &J_2>0 \;\rightarrow\; A_{254}>0,\\ &J_3>0 \;\rightarrow\; A_{034}+A_{153}+A_{013}>0,\\ &J_4>0 \;\rightarrow\; A_{145}+A_{234}+A_{124}>0,\\ &J_5>0 \;\rightarrow\; A_{253}+A_{045}+A_{205}>0.\end{aligned} \tag{3.8}$$

Furthermore, if we consider the Jacobian matrix itself and not only its determinant, we have

$$\frac{1}{4}\,\mathbf{J} = (\mathbf{y}_2-\mathbf{y}_0, \mathbf{y}_3-\mathbf{y}_0).$$

For example,

$$\frac{1}{4}\,\mathbf{J}(1,0,0) = (\mathbf{x}_3-\mathbf{x}_0, \mathbf{x}_5-\mathbf{x}_0)$$

which means that the Jacobian of the three corners is equal to that of the corner Bézier triangle. On top of guaranteeing validity of the element, we can therefore also control its corner angles by controlling the P1 corner triangles of the Bézier expansion. In our experiments, we only use these corner Jacobians to control the shape/angles of the P2 triangle, but it is of course possible to add the other Jacobians.

3.3 Generalization to PN Bézier triangles. To obtain an expression of the Bézier coefficients of the Jacobian determinant as triangle areas, we have to input the interpolation of the triangle given at Equation (3.4) into Formula (3.2) and rearrange the equation as a sum of Bézier functions.

Let us first use a symbol for the coefficients that lie in the Bézier basis functions: $C_{ijk} = \frac{(i+j+k))!}{i!j!k!}$ and let us highlight the following relation:

$$i\;C_{ijk} = \frac{(i+j+k)!}{(i-1)!j!k!} = (i+j+k)\;C_{(i-1)jk}. \tag{3.9}$$

Now, the derivative of the interpolation with respect to λ_1 reads

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial \lambda_1} = &\sum_{\substack{i,j,k\\ j\geq 1}}^{p} C_{ijk}\big[j\;\lambda_0^i\lambda_1^{j-1}\lambda_2^k\;\mathbf{x}_{ijk}\big]\\ &-\sum_{\substack{i,j,k\\ i\geq 1}}^{p} C_{ijk}\big[i\;\lambda_0^{i-1}\lambda_1^{j}\lambda_2^k\;\mathbf{x}_{ijk}\big].\end{aligned}$$

Note that if $j \geq 1$, then $i,k \leq p-1$. By transforming $j \rightarrow j'+1$ for the first term and $i \rightarrow i'+1$ for the second and using the the relation (3.9), we obtain

$$\begin{aligned}\frac{\partial \mathbf{x}}{\partial \lambda_1} &= \sum_{i,j',k}^{p-1}(j'+1)\;C_{i(j'+1)k}\big[\lambda_0^i\lambda_1^{j'}\lambda_2^k\;\mathbf{x}_{i(j'+1)k}\big]\\ &\quad -\sum_{i',j,k}^{p-1}(i'+1)\;C_{(i'+1)jk}\big[\lambda_0^{i'}\lambda_1^{j}\lambda_2^k\;\mathbf{x}_{(i'+1)jk}\big]\\ &= p\sum_{i,j,k}^{p-1}\mathcal{T}_{i,j,k}^{(p-1)}\;\big[\mathbf{x}_{i(j+1)k}-\mathbf{x}_{(i+1)jk}\big].\end{aligned}$$

We see that the partial derivative of the interpolation with respect to λ_1 is a sum of order p–1 Bézier functions. By symmetry, we obtain the same result for λ_2. For the sake of simplicity, let us introduce a notation for the vectors that have been obtained: $\mathbf{v}_{ijk}^{lmn} = \mathbf{x}_{lmn} - \mathbf{x}_{ijk}$. The cross product of the partial derivative gives

$$\begin{aligned}&\frac{\partial \mathbf{x}}{\partial \lambda_1}\times\frac{\partial \mathbf{x}}{\partial \lambda_2} =\\ &p^2\sum_{i,j,k}^{p-1}\sum_{l,m,n}^{p-1}\mathcal{T}_{i,j,k}^{(p-1)}\mathcal{T}_{l,m,n}^{(p-1)}\big[\mathbf{v}_{(i+1)jk}^{i(j+1)k}\times\mathbf{v}_{(l+1)mn}^{lm(n+1)}\big].\end{aligned}$$

Now, two steps remains before achieving the desired expression. The first step consists in writing the two vectors as a sum of triangle areas. For instance, we can write the first of the two vectors as in Figure 3:

$$\mathbf{v}_{(i+1)jk}^{i(j+1)k} = \mathbf{v}_{(i+1)jk}^{(l+1)mn} + \mathbf{v}_{(l+1)mn}^{i(j+1)k}.$$

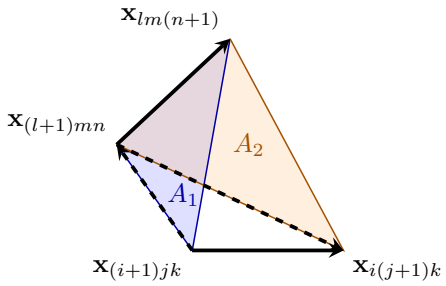

Figure 3: The bottom vector is decomposed in order to express the cross product as a difference of triangles area.

The cross product then becomes

$$\begin{aligned}\mathbf{v}^{i(j+1)k}_{(i+1)jk} \times \mathbf{v}^{lm(n+1)}_{(l+1)mn} &= A\big(\mathbf{x}_{(l+1)mn}, \mathbf{x}_{lm(n+1)}, \mathbf{x}_{(i+1)jk}\big) \\ &\quad + A\big(\mathbf{x}_{(l+1)mn}, \mathbf{x}_{i(j+1)k}, \mathbf{x}_{lm(n+1)}\big) \\ &= A_2 - A_1\end{aligned}$$

The second step consists in transforming the rest of the expression :

$$\begin{aligned}\mathcal{T}^{(p-1)}_{i,j,k}\mathcal{T}^{(p-1)}_{l,m,n} &= C_{ijk}C_{lmn}\ \lambda_0^{i+l}\lambda_1^{j+m}\lambda_2^{k+n} \\ &= \frac{C_{ijk}C_{lmn}}{C_{(i+l)(j+m)(l+n)}}\ \mathcal{T}^{(2p-2)}_{i+l,j+m,k+n}.\end{aligned}$$

A coefficient $J_{\alpha,\beta,\gamma}$ of the Jacobian determinant is thus given by the sum of all the terms for which $i+l=\alpha$, $j+m=\beta$ and $k+n=\gamma$. Writing the final expression for the PN triangle by combining the two steps is straightforward but would be too tedious and not useful for the comprehension of this article. We thus let the interested reader derive it on their own.

Note that this computation leads to a number of triangle areas that is not optimal. Some simplifications are possible to reduce this number, for instance by splitting big triangles into smaller ones.

3.4 Cubic Bézier triangles. Let us apply the generalization described in the previous section to the cubic triangle. Given notations of Figure 4, the interpolation of a P3 triangle in terms of Bézier polynomials is

$$\begin{aligned}\mathbf{x}(\lambda_0,\lambda_1,\lambda_2) = {} & \lambda_0^3\mathbf{x}_0 + 3\lambda_0^2\lambda_1\mathbf{x}_3 + 3\lambda_0\lambda_1^2\mathbf{x}_4 + \\ & \lambda_1^3\mathbf{x}_1 + 3\lambda_1^2\lambda_2\mathbf{x}_5 + 3\lambda_1\lambda_2^2\mathbf{x}_6 + \\ & \lambda_2^3\mathbf{x}_2 + 3\lambda_2^2\lambda_0\mathbf{x}_7 + 3\lambda_2\lambda_0^2\mathbf{x}_8 + \\ & 6\lambda_0\lambda_1\lambda_2\mathbf{x}_9.\end{aligned} \tag{3.10}$$

Similarly to the P2 case (Equation 3.6), it is possible to show, and this is a general statement that applies to any high-order triangle, that the Jacobian of P3 triangle can also be expressed as the area of one P1 triangle :

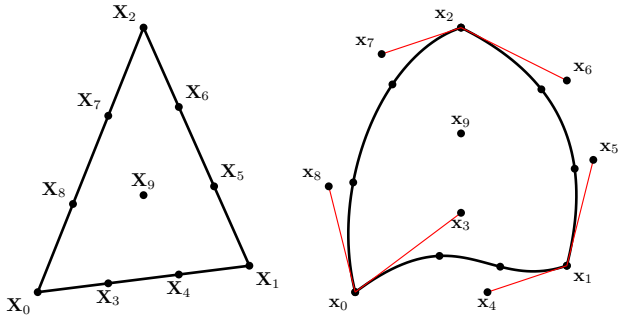

Figure 4: A straight-sided triangle (left) that is the ideal triangle for a third order Bezier triangle (right).

$$\boxed{J(\lambda_0,\lambda_1,\lambda_2) = 8A_{\mathbf{y}_0\,\mathbf{y}_1\,\mathbf{y}_2}} \tag{3.11}$$

where the three corners $\mathbf{y}_0$, $\mathbf{y}_1$ and $\mathbf{y}_2$ are located in three P2 (in general, P(N-1)) triangles whose Bézier control points are three subsets of the Bézier points of the P3 triangle:

$$\begin{aligned}\mathbf{y}_0 &= 3(\lambda_0^2\mathbf{x}_1 + \lambda_1^2\mathbf{x}_4 + \lambda_2^2\mathbf{x}_7 + \\ &\quad 2\lambda_0\lambda_1\mathbf{x}_3 + 2\lambda_0\lambda_2\mathbf{x}_8 + 2\lambda_1\lambda_2\mathbf{x}_9) \\ \mathbf{y}_1 &= 3(\lambda_0^2\mathbf{x}_3 + \lambda_1^2\mathbf{x}_1 + \lambda_2^2\mathbf{x}_6 + \\ &\quad 2\lambda_0\lambda_1\mathbf{x}_4 + 2\lambda_0\lambda_2\mathbf{x}_9 + 2\lambda_1\lambda_2\mathbf{x}_5) \\ \mathbf{y}_2 &= 3(\lambda_0^2\mathbf{x}_8 + \lambda_1^2\mathbf{x}_5 + \lambda_2^2\mathbf{x}_2 + \\ &\quad 2\lambda_0\lambda_1\mathbf{x}_9 + 2\lambda_0\lambda_2\mathbf{x}_7 + 2\lambda_1\lambda_2\mathbf{x}_6).\end{aligned}$$

From Equation (3.10), computing the partial derivatives is straightforward if we keep in mind that $\lambda_0 = 1-\lambda_1-\lambda_2$. They are vector valued functions that appear to be expressed as the interpolation of 6 "control vectors" $\mathbf{v}_{ij} = \mathbf{x}_j - \mathbf{x}_i$ (as defined in the previous Section) with quadratic Bézier triangular polynomials:

$$\begin{aligned}\frac{\partial\mathbf{x}}{\partial\lambda_1} &= 3\left[\lambda_0^2\mathbf{v}_{03} + \lambda_1^2\mathbf{v}_{41} + \lambda_2^2\mathbf{v}_{76}\right. \\ &\quad \left. + 2\lambda_0\lambda_1\mathbf{v}_{34} + 2\lambda_1\lambda_2\mathbf{v}_{95} + 2\lambda_0\lambda_2\mathbf{v}_{89}\right] \\ &= 2(\mathbf{y}_1 - \mathbf{y}_0)\end{aligned}$$

$$\begin{aligned}\frac{\partial\mathbf{x}}{\partial\lambda_2} &= 3\left[\lambda_0^2\mathbf{v}_{08} + \lambda_1^2\mathbf{v}_{45} + \lambda_2^2\mathbf{v}_{72}\right. \\ &\quad \left. + 2\lambda_0\lambda_1\mathbf{v}_{39} + 2\lambda_1\lambda_2\mathbf{v}_{96} + +2\lambda_0\lambda_2\mathbf{v}_{87}\right] \\ &= 2(\mathbf{y}_2 - \mathbf{y}_0)\end{aligned}$$

As the Jacobian determinant is the product of those two partial derivatives (Eq. 3.2), it is thus a quartic polynomial and possesses 15 coefficients. Those coefficients are to be expressed as linear combination of triangle areas whose corners are the 10 Bézier points of the P3 triangle. The final steps to obtain this, as described in the previous section, is to gather the terms in identical polynomial, transform each cross product of vectors as triangle areas and simplify when possible.

As we explained above, it's not easy to find the smallest possible number of triangles to express the validity conditions of a PN triangle. We know here that 15 conditions are needed to express the validity of the P3 triangle, and we think we have found the optimal expression for these 15 conditions, though we do not provide a proof. We believe it is useful to give the 15 P3 conditions *in extenso*:

$$\begin{aligned}
J_0 &= 18\,A_{038}\\
J_1 &= 18\,A_{415}\\
J_2 &= 18\,A_{762}
\end{aligned}$$

$$\begin{aligned}
J_3 &= 9\left(A_{039}+A_{348}+A_{043}\right)\\
J_5 &= 9\left(A_{419}+A_{345}+A_{143}\right)\\
J_6 &= 9\left(A_{159}+A_{564}+A_{165}\right)\\
J_8 &= 9\left(A_{629}+A_{567}+A_{265}\right)\\
J_9 &= 9\left(A_{279}+A_{786}+A_{287}\right)\\
J_{11} &= 9\left(A_{809}+A_{783}+A_{087}\right)
\end{aligned}$$

$$\begin{aligned}
J_4 &= 3\left(4\,A_{349}+A_{035}+A_{418}+A_{043}+A_{014}\right)\\
J_7 &= 3\left(4\,A_{569}+A_{157}+A_{624}+A_{165}+A_{126}\right)\\
J_{10} &= 3\left(4\,A_{789}+A_{273}+A_{806}+A_{287}+A_{208}\right)
\end{aligned}$$

$$\begin{aligned}
J_{12} &= 3\big(2\,A_{478}-2\,A_{378}+2A_{398}\\
&\qquad +A_{036}+A_{058}-A_{038}\big)\\
J_{13} &= 3\big(2\,A_{634}-2\,A_{534}+2\,A_{594}\\
&\qquad +A_{158}+A_{174}-A_{154}\big)\\
J_{14} &= 3\big(2\,A_{856}-2\,A_{756}+2\,A_{796}\\
&\qquad +A_{274}+A_{236}-A_{276}\big)
\end{aligned}$$

If all 15 coefficients J_i are positive, then the P3 triangle is provably valid. Let us point out that the number triangle areas given here is optimal. Indeed, the Bézier points on which depends each coefficient cannot be changed, and the number of triangle cannot be less than the number of point minus 2.

3.5 The quadratic tetrahedron. The analysis we propose remains very similar when we go up in dimension and consider the case of tetrahedra instead of triangles. For a quadratic tetrahedron, the interpolation is written

$$\mathbf{x}(\lambda_0,\lambda_1,\lambda_2,\lambda_3)=\sum_{i,j,k,l}^{2}\mathcal{T}^{(2)}_{i,j,k,l}\;\mathbf{x}_{ijkl},$$

where the corresponding Bézier functions are

$$\mathcal{T}^{(2)}_{i,j,k,l}=\frac{p!}{i!j!k!l!}\;\lambda_0^i\,\lambda_1^j\,\lambda_2^k\,\lambda_3^l,\quad i+j+k+l=2.$$

We order the point as in Figure 5, so that only one index is necessary.

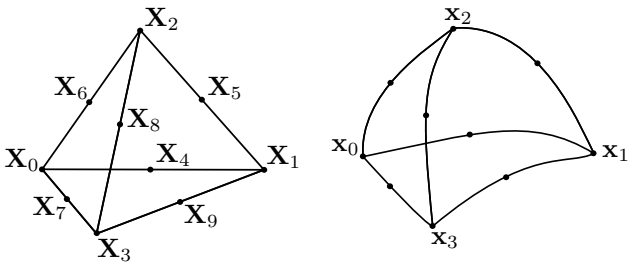

Figure 5: An ideal tetrahedra (left) and an order 2 tetrahedral element (right). For clarity, Bézier points that do not correspond to a vertex (i.e. points $\mathbf{x}_3$ to $\mathbf{x}_9$) are not shown.

Computing the the Jacobian determinant is done by the box product of the *three* partial derivatives

$$J=\left(\frac{\partial\mathbf{x}}{\partial\lambda_1}\times\frac{\partial\mathbf{x}}{\partial\lambda_2}\right)\cdot\frac{\partial\mathbf{x}}{\partial\lambda_3},\quad \lambda_0=1-\lambda_1-\lambda_2-\lambda_3,$$

which are given by

$$\begin{aligned}
\frac{\partial\mathbf{x}}{\partial\lambda_1}&=2\left[\lambda_0\mathbf{v}_{04}+\lambda_1\mathbf{v}_{41}+\lambda_2\mathbf{v}_{65}+\lambda_3\mathbf{v}_{79}\right]=2(\mathbf{y}_1-\mathbf{y}_0)\\
\frac{\partial\mathbf{x}}{\partial\lambda_2}&=2\left[\lambda_0\mathbf{v}_{06}+\lambda_1\mathbf{v}_{45}+\lambda_2\mathbf{v}_{62}+\lambda_3\mathbf{v}_{78}\right]=2(\mathbf{y}_2-\mathbf{y}_0)\\
\frac{\partial\mathbf{x}}{\partial\lambda_3}&=2\left[\lambda_0\mathbf{v}_{07}+\lambda_1\mathbf{v}_{49}+\lambda_2\mathbf{v}_{68}+\lambda_3\mathbf{v}_{73}\right]=2(\mathbf{y}_3-\mathbf{y}_0),
\end{aligned}$$

where $\mathbf{v}_{ij}=\mathbf{x}_j-\mathbf{x}_i$.

The Jacobian determinant is thus in this case a cubic polynomial in dimension 3 and the number of coefficients is 20. Each coefficient is associated to a Bézier function in $\lambda_0^i\,\lambda_1^j\,\lambda_2^k\,\lambda_3^l$ for a given quadruplet $(i,j,k,l),\ i+j+k+l=3$. To obtain an expression of those coefficients as volumes of tetrahedra, one has to develop the box product of the three partial derivative and gather the terms in identical $\lambda_0^i\,\lambda_1^j\,\lambda_2^k\,\lambda_3^l$. As an example, the coefficient J_0 which is associated to λ_0^3 is given by

$$J_0=8\left(\mathbf{v}_{04}\times\mathbf{v}_{06}\right)\cdot\mathbf{v}_{07}=48\,V_{0467}$$

Listing all 20 coefficients would take too much place in this article. Here, we give the expression for one of each type which correspond to the position of the point in the P3 tetrahedron: on a corner, on an edge or on a face. The corner coefficient is J_0 and the two other are respectively

$$\begin{aligned} J_4 &= 16\left(V_{0469} + V_{0457} + V_{0167} - V_{0467}\right) \\ J_{16} &= 8\left(V_{0458} - V_{0456} + V_{0429} - V_{0469}\right. \\ &\quad + V_{0467} - V_{0468} + V_{0127} - V_{0427} \\ &\quad \left. + V_{0168} - V_{0167} + V_{0569} + V_{4567}\right). \end{aligned}$$

It can be proved that the edge coefficient cannot be computed with less than 4 volumes. On the other hand, the 12 volumes of face coefficients obtained here is suboptimal. One simplification that we have found is

$$V_{0465} + V_{0458} + V_{0486} = V_{0658} + V_{4658}$$

but others may exist.

4 Untangling PN elements

Let us now turn our attention to the practical untangling of high-order triangles and tetrahedra. In the previous section, we have established a list of sufficient conditions for a PN element to be valid. These conditions take the form of either P1 determinants (for corners), or linear combinations of P1 determinants, that need to stay positive. To achieve these conditions in practice, we propose to use the method of Garanzha et al.[4]. In this work, author indeed propose a methodology for untangling simplicial meshes in a very robust fashion. As our formulation for a PN element is formulated only as conditions on P1 simplices, the adaptation of the untangling method to our case will be straightforward.

4.1 Untangling simplices. Let us first recall the principles of the approach proposed in [4]. Consider the task of untangling a mesh of triangles composed of n nodes. The goal is to optimize the position of the nodes so that each triangle is correctly oriented and exhibits low distortion with relation to an ideal triangular shape. However, distortion cannot be completely eliminated in practice. One can only hope for a mapping that preserves angles (conformal) or areas (authalic), but not both at the same time. What is proposed in [4] is to achieve a compromise between the two.

Let $\mathcal{T}$ be a triangulation of a domain Ω of the plane (the principle trivially extends to tetrahedrizations in space). Each triangle $t = (\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ of $\mathcal{T}$ is associated with a triangle $(\mathbf{X}_c, \mathbf{X}_b, \mathbf{X}_c)$ of *ideal shape*. This idea is quite similar to that used in anisotropic mesh adaptation, where a metric field is used to define angles and areas and where the ideal element is an equilateral or unit triangle with respect to the given metric.

The jacobian matrix $\mathbf{J} = \frac{d\mathbf{x}}{d\mathbf{X}}$ can easily be computed as

$$\mathbf{J} = \left(\mathbf{X}_b - \mathbf{X}_a, \mathbf{X}_c - \mathbf{X}_a\right)^{-1} \left(\mathbf{x}_b - \mathbf{x}_a, \mathbf{x}_c - \mathbf{x}_a\right).$$

The determinant of $\mathbf{J}$ plays of course an important role in this work. We write

$$\det(\mathbf{J}) = J = \frac{(\mathbf{x}_b - \mathbf{x}_a) \times (\mathbf{x}_c - \mathbf{x}_a) \cdot \mathbf{e}_z}{(\mathbf{X}_b - \mathbf{X}_a) \times (\mathbf{X}_c - \mathbf{X}_a) \cdot \mathbf{e}_z}.$$

In [4], authors propose to minimize

$$\lim_{\epsilon \to 0} \sum_{t \in \mathcal{T}} f_\epsilon(\mathbf{J}_t, \epsilon) + \lambda g_\epsilon(\mathbf{J}_t, \epsilon) \tag{4.12}$$

where

$$f_\epsilon(\mathbf{J}) = \frac{\mathrm{tr}(\mathbf{J}^T\mathbf{J})}{\chi(J,\epsilon)} \quad \text{and} \quad g_\epsilon(\mathbf{J}) = \frac{J^2+1}{\chi(J,\epsilon)}.$$

Here, $\chi(x,\epsilon) = \frac{x+\sqrt{\epsilon^2+x^2}}{2}$ is a regularization function that tends to the ramp function when $\epsilon \to 0$.

The term f_ϵ is minimal when the mapping preserve the angles of the triangles while minimizing g_ϵ amounts at preserving their area. Parameter $\lambda > 0$ controls the balance between the two and thus controls the tradeoff. To minimize Eq. (4.12), we rely on the same quasi-Newton scheme proposed in [4]. In particular, we do not need to compute the Hessian of the function but only the derivatives of J with respect to coordinates $\mathbf{x}_{a,b,c}$. The formula is the same as Eq.(3.3):

$$\frac{\partial J}{\partial \mathbf{x}_{a,b,c}} = \frac{(\mathbf{x}_{c,a,b} - \mathbf{x}_{b,c,a}) \times \mathbf{e}_z}{(\mathbf{X}_b - \mathbf{X}_a) \times (\mathbf{X}_c - \mathbf{X}_a) \cdot \mathbf{e}_z}. \tag{4.13}$$

One of the advantages of this approach over [11], for example, is the ease with which the gradient of f_ϵ can be calculated. Another advantage to use the regularization proposed in [4] is that f_ϵ is convex so the solution that is found is unique and a simple LBFGS allows to converge in every case.

4.2 Adapting the untangler for P2 triangles. In the case of a P2 mesh, we will first assume that it is made from a valid, good-quality P1 mesh. As already proposed in the litterature (see *e.g.* [5]), we choose this straight-sided P1 element as the ideal shape of the P2 triangle. This has two advantages. Firstly, it allows the algorithm to favor an unchanged mesh because unchanged triangles will have a unit Jacobian. Secondly, it enables the untangling of highly anisotropic meshes such as boundary layer meshes while preserving the anisotropy.

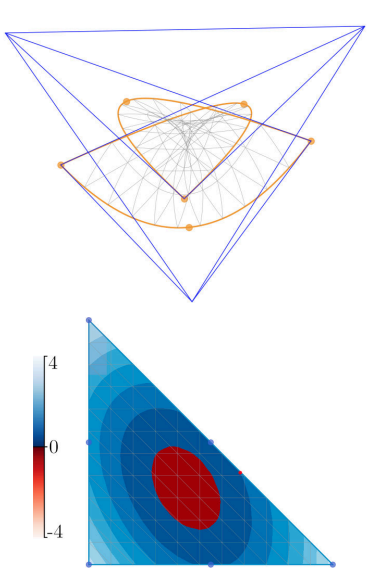

Figure 6: Top Figure shows an example of an invalid P2 triangle with positive Jacobian on all of its boundary. (In orange are the edges and in blue is the Bézier control polygon.) Bottom Figure shows the Jacobian value in the reference triangle.

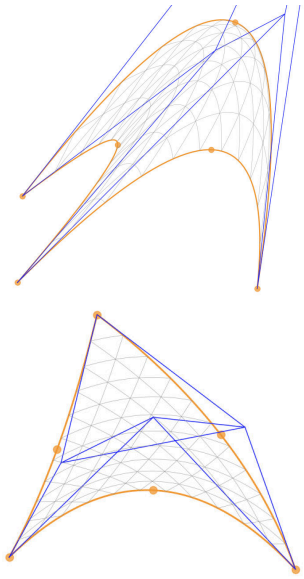

Figure 7: Top Figure shows an example of an invalid P2 triangle for which all its Bézier triangles are valid. Bottom Figure shows an example of a P2 triangle that is valid even though it has an invalid central Bézier triangle.

The Jacobian matrix is this time varying over the element and the straightforward generalization of the minimization problem (4.12) would necessitate to evaluate f_ϵ and g_ϵ everywhere inside the element. In practice, this is of course unfeasible, which is why we use conditions (3.8) which we established to be sufficient for the P2 elements to be valid. The first three conditions are simple P1 triangles between Bezier control points whose Jacobian should remain positive. These correspond to the three corner Jacobian of the P2 triangle, that is to say the mappings between triangles $(\mathbf{X}_0, \mathbf{X}_3, \mathbf{X}_5) \to (\mathbf{x}_0, \mathbf{x}_3, \mathbf{x}_5)$, $(\mathbf{X}_1, \mathbf{X}_4, \mathbf{X}_3) \to (\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_3)$ and $(\mathbf{X}_2, \mathbf{X}_5, \mathbf{X}_4) \to (\mathbf{x}_2, \mathbf{x}_5, \mathbf{x}_4)$.

Untangling these three triangles for each P2 element is necessary for global validity but is unfortunately not a sufficient solution, as there exist invalid P2 triangles whose Jacobian is positive at all three corners. Figure 6 shows an example of such a configuration.

The most intuitive idea for improving the procedure would be to add the central triangle $(\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5)$. On the example of Figure 6, it is indeed invalid. This idea have been proposed in previous works [1] and effectively makes the process more robust. It is indeed quite complicated to find an invalid P2 triangle whose four Bézier triangles are valid, though such configurations exist (see Figure 7,top). On the other hand, it's very easy to find a perfectly valid and very well conditioned P2 triangle where the central Bézier triangle is invalid (see Figure 7,bottom). Thus, the positivity of this fourth triangle is neither necessary nor sufficient for the validity in the P2 case and should not be used for untangling P2 meshes.

Instead, we adapt the untangling method to the last three conditions of (3.8), which are sufficient for validity. In the f_ϵ term of Eq. (4.12), we only consider the three corner Jacobians as before. The other term g_ϵ is modified to take into account the three other conditions: as the function acts as a barrier for negative values for any ϵ, we add to the energy the linear combination of determinants that needs to stay positive.

More precisely, let us consider a combination of N triangle areas whose corners are the $\mathbf{x}_j$ of a given PN element. Assume that the indices of the triangle vertices are encoded in a $N \times 3$ matrix C_{ij}. Then the condition boils down to:

$$\sum_{i=1}^{N} \det\left(\mathbf{x}_{C_{i,2}} - \mathbf{x}_{C_{i,1}}, \mathbf{x}_{C_{i,3}} - \mathbf{x}_{C_{i,1}}\right) > 0.$$

In the case of the P2 triangle, $N = 3$ (see Equation (3.8)) and there are three 3×3 matrices

$$C^{(3)} = \begin{pmatrix} 0 & 3 & 4 \\ 1 & 5 & 3 \\ 0 & 1 & 3 \end{pmatrix}, \quad C^{(4)} = \begin{pmatrix} 1 & 4 & 5 \\ 2 & 3 & 4 \\ 1 & 2 & 4 \end{pmatrix}, \quad C^{(5)} = \begin{pmatrix} 2 & 5 & 3 \\ 0 & 4 & 5 \\ 2 & 0 & 5 \end{pmatrix}.$$

In order for the condition to be invariant by scaling,

we normalize it by its initial value and write

$$S(\mathbf{x},C) = \frac{\sum_{i=1}^{N} \det\left(\mathbf{x}_{C_{i,2}} - \mathbf{x}_{C_{i,1}}, \mathbf{x}_{C_{i,3}} - \mathbf{x}_{C_{i,1}}\right)}{\sum_{i=1}^{N} \det\left(\mathbf{X}_{C_{i,2}} - \mathbf{X}_{C_{i,1}}, \mathbf{X}_{C_{i,3}} - \mathbf{X}_{C_{i,1}}\right)} > 0.$$

Overall, the final problem we minimize is the following expression:

$$\underbrace{\sum_{i=0}^{2} \left(f_\epsilon(\mathbf{J}_i) + \lambda g_\epsilon(\mathbf{J}_i)\right)}_{\text{Jacobians at corners}} + \underbrace{\lambda \sum_j g_\epsilon(S(\mathbf{x},C^{(j)}))}_{\text{Barriers on linear combinations}}$$

summed over all elements in the mesh. As computing the gradient of g_ϵ is no more complicated for a linear combination of triangles than for a single triangle, evaluating and differentiating this expression is not harder than for the original untangler of [4].

Generalizing this approach for PN triangles and tetrahedra is also straightforward, as we only need to build the matrices C. However, their size and number grows quickly with the order of the element, which increases the cost of the optimization. For a P3 triangle for instance, there are 12 of them that link 3 to 6 triangles together (see Eqs. (3.12)).

5 Application: P2 and P3 untangling of boundary layers.

Let's now try to apply the methodology proposed above to a few boundary layer meshes.

5.1 Ellipsis. The first example is a simple elliptical body (excentricity being equal to 4) around which we have generated (using Gmsh [7]) a boundary layer mesh of thickness 0.1, a first element of size $h_0 = 0.001$ and a geometric progression of reason $r = 1.3$. The ith point of the boundary layer is therefore positioned at a distance $h_i = h_0(1+r)^{i-1}$ to the wall. Figure 8 shows not only the ellipse that describes the geometry, but also the other ellipse that forms the outer boundary of the boundary layer. The interest of our proposed methodology is twofold. Firstly, we have transformed the problem of untangling P2 elements into a simpler problem: untangling P1 elements. Secondly, the tradeoff parameter λ allows us to balance control over the area of the elements (and the thickness of the curved boundary layer) and over the orthogonality of the elements in the boundary layer. As both quantities are impossible to preserve perfectly simultaneously, the value of λ in our method specifies which one to favor.

Figure 9 shows untangled P2 boundary layer meshes around the ellipse for three values of λ. Note that the outer ellipse drawn in the various Figures of 9 is not part of the CAD, but represents the outer boundary of the boundary layer as prescribed in the Gmsh file. Another important note: the positioning of the P2 nodes on the ellipse is not equidistant. It has been optimized to make the 1D quadratic mesh as $\mathcal{C}^1$ as possible [11].

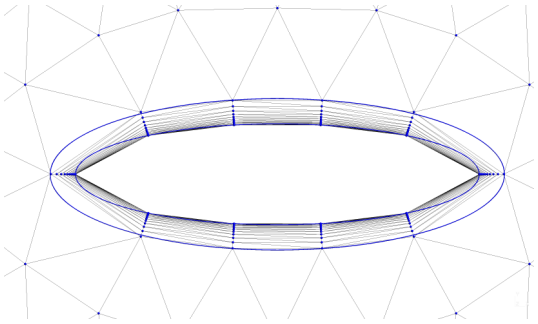

Figure 8: A straight-sided boundary layer *O-mesh* around an ellipsis.

With a parameter of $\lambda = 1/100$, the mesh is untangled, but at the cost of a mild extension of the boundary layer's thickness. If we choose $\lambda = 1$, the thickness of the boundary layer is this time relatively spared, while maintaining good mesh orthogonality. Finally, if we use $\lambda = 100$, i.e. if we resolutely prefer to keep the size of the elements identical, orthogonality is lost and the algorithm introduces curvature in the direction orthogonal to the wall. We believe that choosing $\lambda = 1$ is a good compromise, and this is the value we set for all the following experiments.

5.2 Compressor blade. A second example is briefly presented here on Figure 10, a compressor blade that is part of a cascade (vertical periodic domain). The geometry contains both concave and convex boundaries. Here, optimizing the mesh to conform to both an ideal size and an ideal shape allows to generated meshes that are elegantly curved all along the boundary.

5.3 Three component wing This third example is a classical: the three component wing. (see Figure 11). Again, our method enables the generation of a boundary layer that gracefully follows the boundary on all components.

5.4 P3 boundary layers We used the same 3 test cases (ellipse, blade and wing, see Figure 12), with the same number of elements, but using P3 triangles instead of P2. The number of simplexes needed to apply the sufficient positivity condition rises from 12 in P2 to 54 in P3. The time for untangling the three component wing (§5.3) is increased from a few seconds to just under a minute. It is clearly possible to

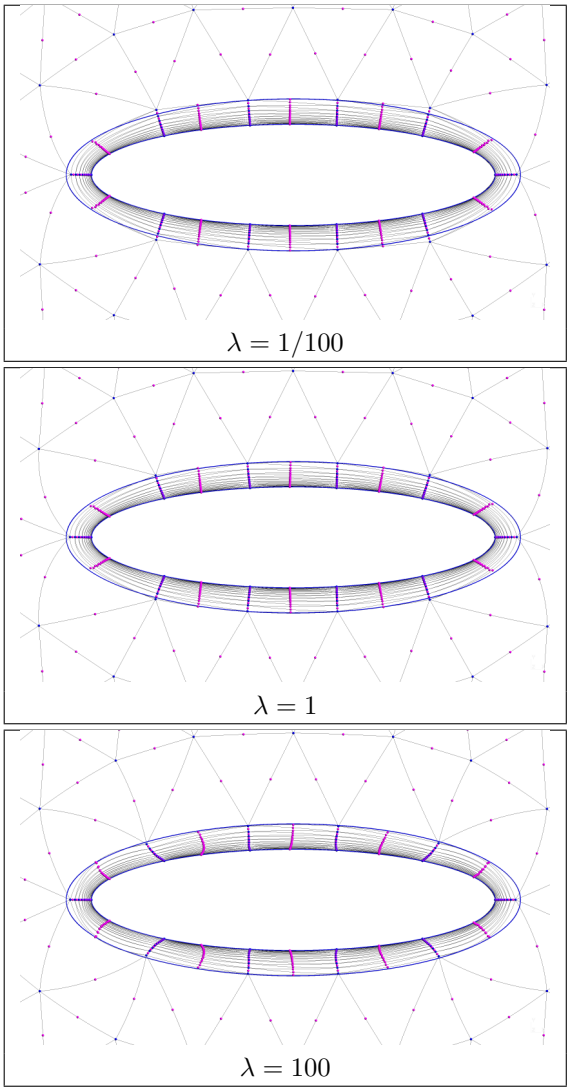

Figure 9: Valid P2 boundary layer mesh around an ellipsis with various values of λ. The case $\lambda = 1$ seems to be the best compromise, with orthogonal meshes and a boundary layer thickness that is not very affected by the curving process The case $\lambda = 1/100$ shows some mild thickening of the boundary layer while the case $\lambda = 100$, we find major disturbances in the regularity of the elements.

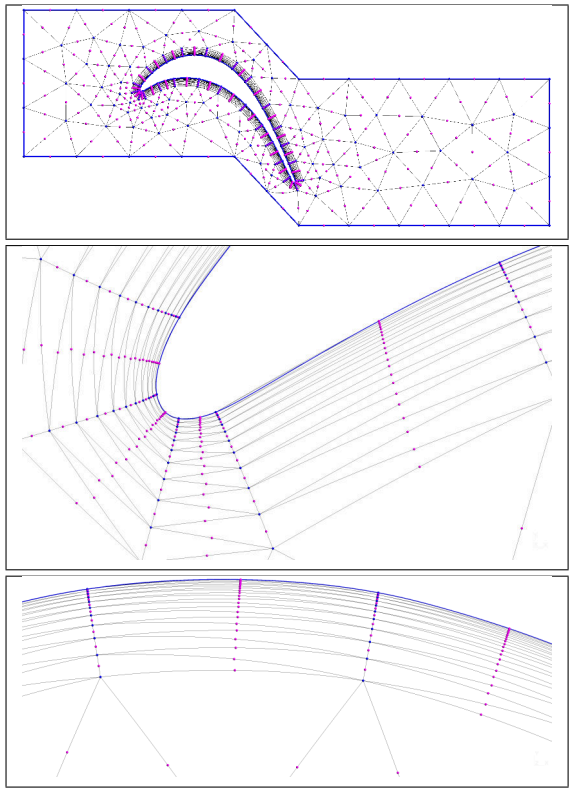

Figure 10: Valid P2 boundary layer mesh around an compressor blade ($\lambda = 1$).

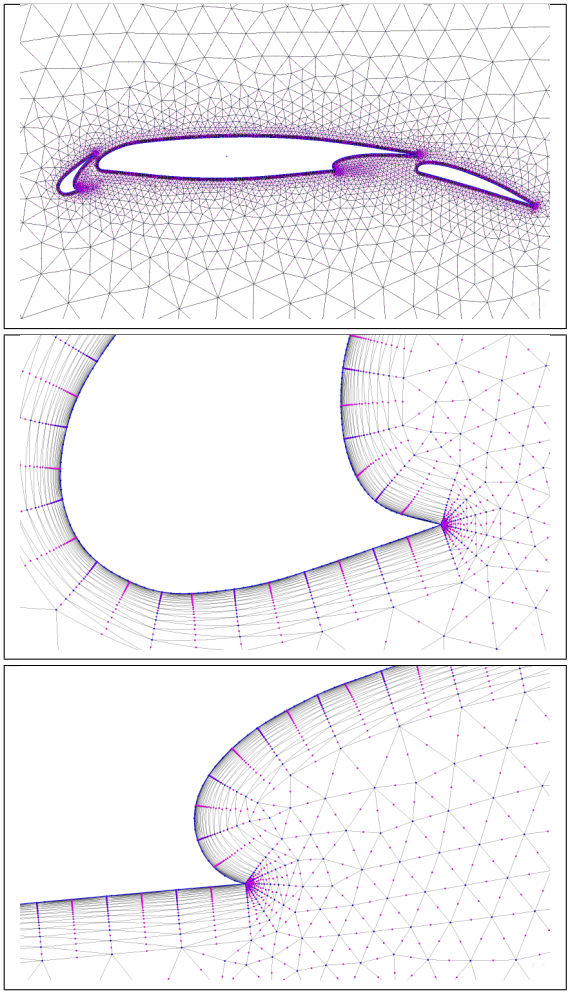

Figure 11: Valid P2 boundary layer mesh of the three component wing ($\lambda = 1$).

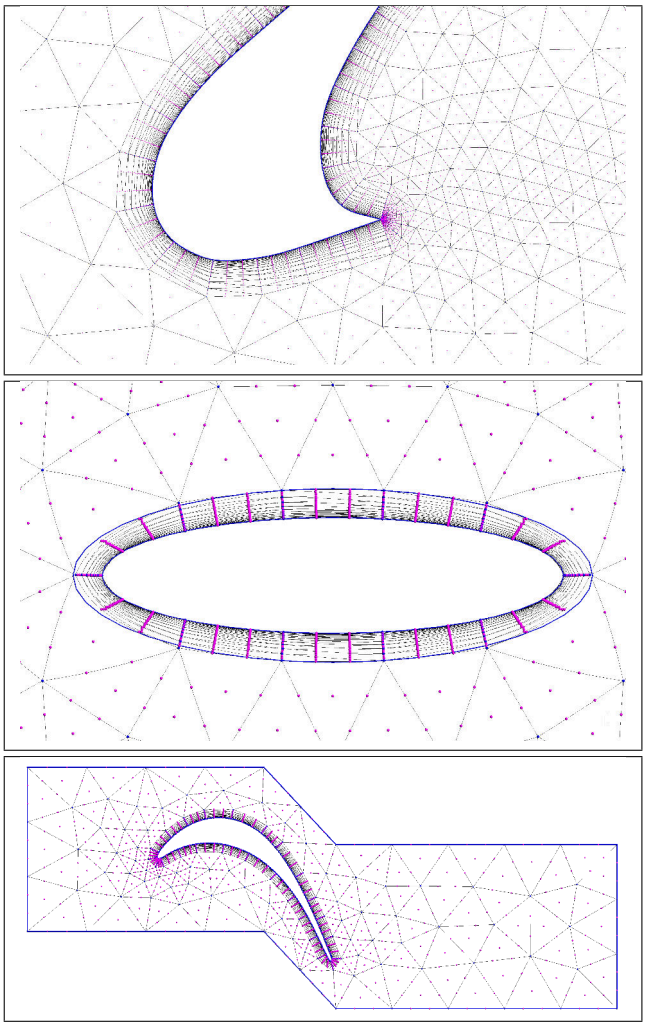

Figure 12: Valid P3 boundary layer meshes.

improve the performance of our algorithm, for example, by parallelizing the calculation of f_ϵ and its gradient, but for us, P3 will remain on the order of 5 times more expensive than P2, and using higher orders will drastically increase the computational cost. For P4, for example, the Jacobian is of order 6 and we will need 28 conditions and 154 triangle areas to compute. For a given order p, we found that the maximum number of triangle needed to compute a single coefficient is bounded by

$$\frac{p^2}{3} + 5\,\frac{p}{3} - \frac{7}{9},$$

which means that the total number of triangles increases at worst as p^4.

5.5 Tetrahedral sphere Finally, we consider the 3D case of untangling P2 tetrahedra. On Figure 13, we test

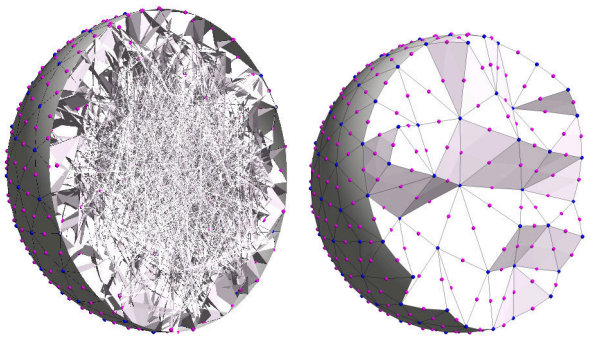

Figure 13: Stress test of the untangler on a P2 sphere. Left: initial sphere where all P1 and P2 interior nodes have been assigned a random position. Right: untangled result.

our approach on the simple case of a sphere where all interior points have been assigned a random position. Taking as an ideal shape the P1 untangled sphere, optimizing for our conditions is sufficient to recover a good quality P2 mesh.

6 Initialization and Convergence

In our first experiments, the optimization algorithm – L-BFGS [9] in this case – was initialized using the straight-sided mesh data as an initial condition, except on the boundary where the high-order points are moved on the CAD with the aim of minimizing the geometric error (see Figure 14). The optimization still converged in all the cases we tested, but this convergence was very slow. Indeed, at the start of the optimization process, only triangles with an edge on the domain boundary are tangled, all others matching their ideal shape. The algorithm propagates information layer by layer, and the number of layers of elements in a boundary layer is potentially high (> 20).

It is actually quite simple to speed up the optimization process by propagating the movement of boundary nodes throughout their respective columns in the boundary layer. This mesh is potentially invalid (see Figure 14), but L-BFGS can now act locally.

With this good initialization, it takes 2 seconds on a standard laptop to untangle the three component wing of Figure 11 (13,068 triangles, most of them in the boundary layer). Figure 15 shows the state of the mesh after 2 seconds of optimization in the case of the simple initial condition. Obtaining the convergence takes more than 30 seconds in this case.

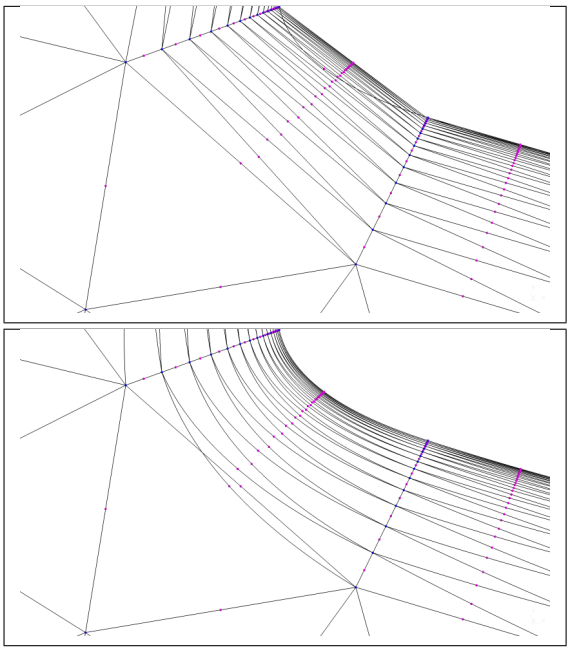

Figure 14: Initial conditions for the optimization problem: only boundary nodes are initially positioned on the CAD (top) and displacement of the boundary node is propagated on the whole boundary layer column (bottom). Both setups may present invalid elements, though their untangling is faster in the latter.

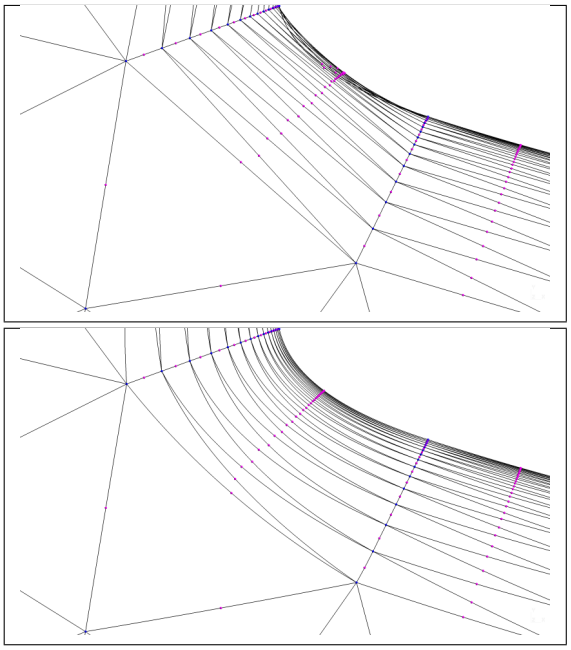

Figure 15: Resulting meshes after 2 seconds of optimization for different initial conditions. When boundary nodes only are initially positioned on the CAD (top Figure), the mesh is still tangled while when boundary nodes are initially propagated on the whole boundary layer (bottom Figure), the optimization procedure is converged.

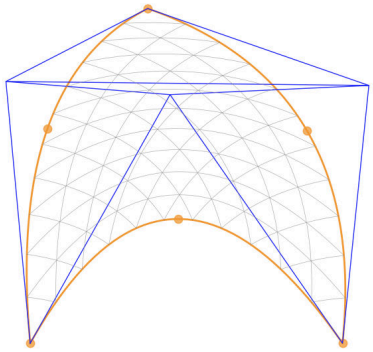

Figure 16: A valid element that violates sufficient conditions (3.8).

7 Conclusions and Future Works

We have introduced an optimization method for fast untangling of order 2 and 3 meshes, that rely on a simple set of sufficient validity conditions. The conditions given in Eq.(3.8) are reasonably sharp[8] yet they sometimes provide *false negative* results. Figure 16 shows an example of a valid P2 triangle that violates one of the 6 conditions (3.8). Although these configurations were not needed in practice to achieve valid meshes, relaxing our constraints to include these false negatives is a path for future work. An idea would be to utilize the methodology proposed in [8], which is based on a recursive division of the Bézier triangulation using De Casteljau's algorithm. The sharp bound it provides could be used in our setup along a subdivision scheme for a progressively stiffening scheme.

While we showed that the same methodology could be applied to any order, computation costs quickly grow. More research is needed to determine a more efficient approach for orders higher than 3.

References

[1] R. Abgrall, C. Dobrzynski, and A. Froehly, *A method for computing curved meshes via the linear elasticity analogy, application to fluid dynamics problems*, International Journal for Numerical Methods in Fluids, 76 (2014), pp. 246–266.

[2] C. Dobrzynski and G. El Jannoun, *High order mesh untangling for complex curved geometries*, PhD thesis, INRIA Bordeaux, équipe CARDAMOM, 2017.

[3] M. Fortunato and P.-O. Persson, *High-order unstructured curved mesh generation using the winslow equations*, Journal of Computational Physics, 307 (2016), pp. 1–14.

[4] V. Garanzha, I. Kaporin, L. Kudryavtseva, F. Protais, N. Ray, and D. Sokolov, *Foldover-free*

maps in 50 lines of code, ACM Transactions on Graphics (TOG), 40 (2021), pp. 1–16.

[5] A. Gargallo-Peiró, X. Roca, J. Peraire, and J. Sarrate, *Distortion and quality measures for validating and generating high-order tetrahedral meshes*, Engineering with Computers, (2014), pp. 1–15.

[6] P.-L. George, H. Borouchaki, and P. Laug, *Construction de maillages de degré 2-Partie 1: Triangle P2*, PhD thesis, INRIA, 2011.

[7] C. Geuzaine and J.-F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities*, International journal for numerical methods in engineering, 79 (2009), pp. 1309–1331.

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

[9] D. C. Liu and J. Nocedal, *On the limited memory BFGS method for large scale optimization*, Mathematical Programming, 45 (1989), pp. 503–528.

[10] M. Stees, M. Dotzel, and S. M. Shontz, *Untangling high-order meshes based on signed angles*, Proceedings of the 28th international meshing roundtable, (2020).

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