Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 82 (2014) 228 – 239

Procedia Engineering

www.elsevier.com/locate/procedia

Optimizing the geometrical accuracy of 2D curvilinear meshes

Jean-François Remacle[a], Jonathan Lambrechts[a,c], Christophe Geuzaine[b], Thomas Toulorge[a,c]

[a] *Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Bâtiment Euler, Avenue Georges Lemaître 4, 1348 Louvain-la-Neuve, Belgium*

[b] *Université de Liège, Department of Electrical Engineering and Computer Science, Montefiore Institute B28, Grande Traverse 10, 4000 Liège, Belgium*

[c] *Fonds National de la Recherche Scientifique, rue d'Egmond, Bruxelles, Belgium*

Abstract

This paper presents a method to generate valid 2D high order meshes with optimized geometrical accuracy. The high order meshing procedure starts with a straight sided mesh. High order points are initially snapped to the real geometry without taking care of the validity of the high order elements. An optimization procedure that both allow to untangle invalid elements and to optimize the geometrical accuracy of the mesh is presented. An area based distance is proposed to compute the geometrical discrepancy between the mesh and its underlying geometry. This area based distance is shown to be strongly related to standard distance measures such as the Hausdorff distance, while being largely faster to compute.

The new distance is minimized through an unconstrained optimization procedure, allowing significant improvements of the geometrical accuracy of high order meshes. A simple computational example shows that geometrically optimized meshes allow smooth CFD solutions.

Peer-review under responsibility of organizing committee of the 23rd International Meshing Roundtable (IMR23)

Keywords: High order meshes, geometrical accuracy

1. Introduction

The development of high-order numerical technologies for engineering analysis has been underway for many years now. For example, Discontinuous Galerkin methods (DGM) have been largely studied in the literature, initially in a theoretical context [6], and now from the application point of view [11]. In many contributions, it is shown that the accuracy of the method strongly depends on the accuracy of the geometrical discretization [3,4,12].

The aim of this paper is to present a method that enables to build geometrically accurate curvilinear meshes in 2 dimensions.

* Jean-Franois Remacle.
E-mail address: jean-francois.remacle@uclouvain.be

Peer-review under responsibility of organizing committee of the 23rd International Meshing Roundtable (IMR23)
doi:10.1016/j.proeng.2014.10.386

Consider a model entity C and the mesh entity C_m that is supposed to approximate C. The first question that arises here is the following one: how do we define a proper distance $d(C, C_m)$ between C and C_m and how do we compute this distance efficiently. Two principal definitions for such distances have been defined in the computational geometry literature, namely the Fréchet distance and the Hausdorff distance.

In this context of curvilinear meshing, distances $d(C, C_m)$ that are computed are usually small in comparison with the typical dimension of either C or C_m. In consequence, $d(C, C_m)$ has to be computed with a high accuracy. In this paper, we show that computing standard distances between the mesh and the geometry may be too expensive for practical computations. Alternative measures of distance are presented that have the advantage to be fast enough to compute and sufficiently accurate. An optimization procedure is developed that allows both to ensure the mesh validity and to drastically reduce the CAD-to-mesh distance.

2. CAD-to-Mesh Distance

2.1. Setup

Consider the following planar parametric curve

$$C \equiv \{\eta \in [\eta_0, \eta_p] \mapsto \boldsymbol{x}(\eta) \in R^2\}$$

and the following $p+1$ successive points on C

$$\boldsymbol{x}_i = \boldsymbol{x}(\eta_i), \quad \text{with} \quad \eta_0 < \eta_1 < \eta_2 \cdots < \eta_{p-1} < \eta_p.$$

A curvilinear mesh edge C_m is defined as the Lagrange approximation of C at order p

$$C_m \equiv \left\{ \xi \in [0,1] \mapsto \boldsymbol{x}_m(\xi) = \sum_{i=0}^{p} \mathcal{L}_i^{(p)}(\xi) \ \in R^2 \right\}. \tag{1}$$

In (1), $\mathcal{L}_i^{(p)}(\xi)$ is the ith Lagrange polynomial of order p.

Curves C and C_m that are both bounded by the vertices $\boldsymbol{x}_0$ and $\boldsymbol{x}_p$ and coincide at least at the $p+1$ Lagrange points $\boldsymbol{x}_i,\ i = 0, \ldots, p$ (see Figure 1).

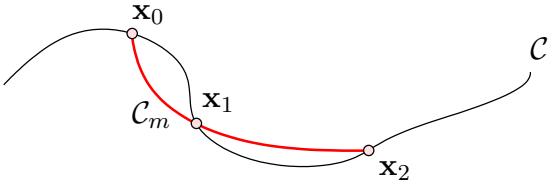

Fig. 1. Typical setup: a model edge C and a quadratic mesh edge C_m.

2.2. Definitions of Distance

Define $\alpha(t)$ (resp. $\beta(t)$) as an arbitrary continuous nondecreasing function from $t \in [0,1]$ onto $\eta \in [\eta_0, \eta_p]$ (resp. $\xi \in [0,1]$). The **Fréchet distance** between C and C_m is defined as

$$d_{\mathrm{F}}(C, C_m) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \|\boldsymbol{x}_m(\alpha(t)) - \boldsymbol{x}(\beta(t))\|.$$

There is a standard interpretation of the Fréchet distance. Consider a man is walking with a dog on a leash. The man is walking on the one curve and the dog on the other one. Both may vary their speed that are represented by

transformations α and β. Backtracking is not allowed which implies α and β to be non-decreasing. Then, the Fréchet distance between the curves is the minimal length of a leash that is necessary.

The **Hausdorff distance** is the longest distance you can be forced to travel by an adversary who chooses a point in one of the two curves. It is defined as

$$d_{\mathrm{H}}(C, C_m) = \max\{ \sup_{\eta\in[\eta_0,\eta_p]} \inf_{\xi\in[0,1]} \|\boldsymbol{x}_m(\eta) - \boldsymbol{x}(\xi)\|, \\ \sup_{\xi\in[0,1]} \inf_{\eta\in[\eta_0,\eta_p]} \|\boldsymbol{x}_m(\eta) - \boldsymbol{x}(\xi)\| \}.$$

Not only $d_H(C, C_m) \leq d_F(C, C_m)$, but the Fréchet distance between two curves can be arbitrarily larger than their Hausdorff distance. The Fréchet distance is usually considered as a more reliable measure of similarity between curves. Figure 2 shows two curves that can be made arbitrary "Hausdorff-close" while being quite dissimilar.

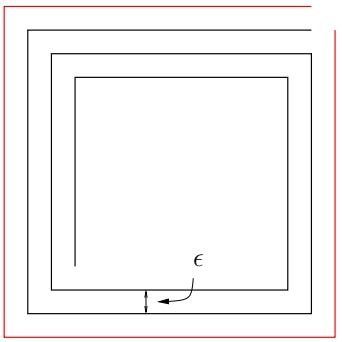

Fig. 2. Two curves that can be made arbitrary close when $\epsilon \to 0$ in terms of their Hausdorff distance (that is exactly 3ϵ) but not in terms of their Fréchet distance that remains finite and equal to the diagonal of the square.

2.3. Optimal sampling of polygonal curves

In computing distances between arbitrary curves one typically approximates the curves by polygonal curves. Assume now $m + 1$ points $\mathbf{p}_i = \boldsymbol{x}_m(\xi_i),\ i \in [0, m]$, that are sampled on C_m. This defines a polygonal curve M formed of m segments for which segment i goes from $\mathbf{p}_i$ to $\mathbf{p}_{i+1}$.

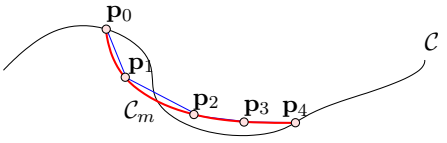

Fig. 3. A model edge C,a mesh edge C_m and a polygonal curve M (in blue).

Let us do the same with C and define a polygonal curve N composed of $n + 1$ points $\mathbf{q}_i,\ i \in [0, n]$.

It is indeed possible to use the well known de Casteljau subdivision scheme to generate optimal samplings i.e. to build polygonal curves that are accurate representations of their continuous counterparts.

Let us find sampling points on the continuous curve in such a way that the distance between the polynomial curve and the continuous curve is less than a threshold distance ϵ.

Consider the Bernstein polynomials:

$$\mathcal{B}_k^{(n)}(\xi) = \binom{n}{k} (1-\xi)^{n-k}\, \xi^k \qquad (\xi \in [0,1]\ ;\ k = 0, ..., n) \tag{2}$$

where $\binom{n}{k} = \frac{n!}{n!(n-k)!}$ is the binomial coefficient. Since Lagrange and Bernstein polynomials span the same function space, we can re-write (1) as the so called Bézier curve

$$\boldsymbol{x}_m(\xi) = \sum_{i=1}^{N_p} \mathcal{B}_k^{(n)}(\xi)\, \boldsymbol{x}_i^b, \quad \xi \in [0,1] \tag{3}$$

where the $\boldsymbol{x}_i^b$'s are the Bézier control points of the curve. Control points $\boldsymbol{x}_i^b$'s are computed from the node locations $\boldsymbol{x}_i$'s using a transformation matrix which computes nodal values from control values:

$$\boldsymbol{T}^{(n)}_{\mathcal{B}\to\mathcal{L}} = \begin{bmatrix} \mathcal{B}_0^{(n)}(\xi_0) & \dots & \mathcal{B}_n^{(n)}(\xi_0) \\ \mathcal{B}_0^{(n)}(\xi_1) & \dots & \mathcal{B}_n^{(n)}(\xi_1) \\ \vdots & \ddots & \vdots \\ \mathcal{B}_0^{(n)}(\xi_n) & \dots & \mathcal{B}_n^{(n)}(\xi_n) \end{bmatrix}.$$

A classical way to optimally sample a Bézier curve is to use de Casteljau's algorithm. A first approximation of the Bézier curve is constructed as the single line segment between $\boldsymbol{x}_0^b$ and $\boldsymbol{x}_{N_p}^b$ (red line segment in Figure 4). The distance d between this single segment and the control polygon is an upper bound of the distance between the curve and the segment because of the convex hull property. The curve is then split into two sub-curves using de Casteljau algorithm,

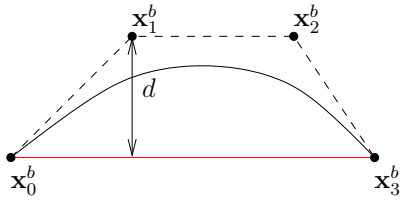

Fig. 4. A cubic Bézier curve, its control polygon (dashed lines) and the rawest approximation of the curve in red.

the two sub-curves being representing exactly the original curve. This argument is applied recursively (see Figure 5) up to the point every sub control polygon is sufficiently flat so that the distance between the control polygon and the approximation is less than ϵ. A similar algorithm can be applied to rational Bézier splines. Most of the CAD entities

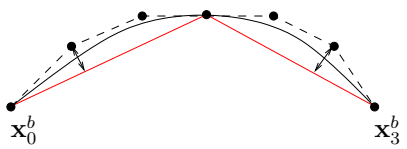

Fig. 5. A first application of de Casteljau's algorithm. The two sub control polygons are getting closer to the Bézier curve.

can be casted as rational Bézier splines so that this optimal subdivision can be applied to most of the curves that are present in CAD models. Nevertheless, a generic recursive sampling algorithm has been developed for curves that have a non standard parametrization.

As an example, Figure 6 presents the subdivision of a boundary using different values of the threshold parameter ϵ.

2.4. Discrete distances between polygonal curves

Discrete Hausdorff and Fréchet distances $\delta_H(M,N)$ and $\delta_F(M,N)$ compute respectively the Hausdorff and the Fréchet distance restricted to discrete points.

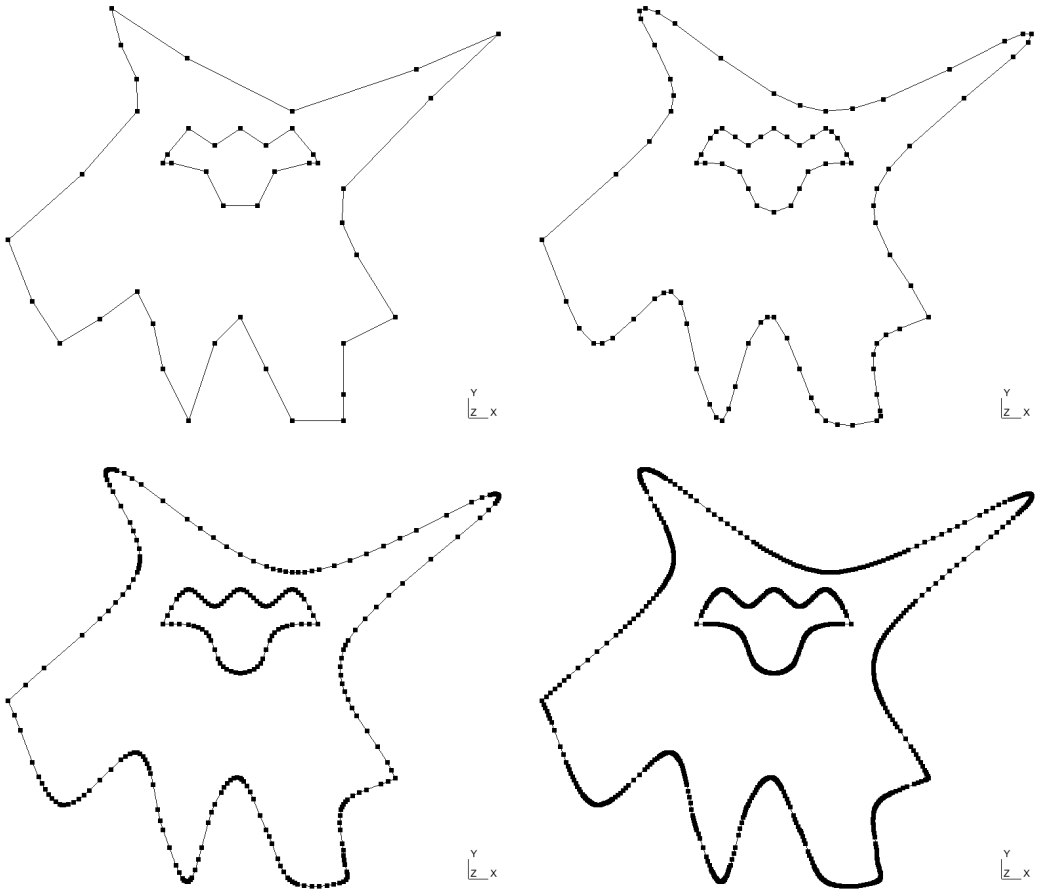

Fig. 6. Adaptive subdivision of a 2D boundary with $\epsilon = 5 \times 10^{-2}$ (88 points), $\epsilon = 10^{-2}$ (196 points), $\epsilon = 10^{-3}$ (554 points), $\epsilon = 10^{-4}$ (858 points)

Computing $\delta_H(M,N)$ consist essentially in computing both Voronoi diagrams of the $\mathbf{p}_i$'s and of the $\mathbf{q}_i$'s and to find Voronoi cells of $\mathbf{p}_i$ that contains $\mathbf{q}_j$ and Voronoi cells of $\mathbf{q}_j$ that contains $\mathbf{p}_i$. Computing $\delta_H(M,N)$ requires $O(m+n)\log(n+m)$ operations. In our implementation, we do not explicitely construct the Voronoi diagram. An approximate nearest neighbor algorithm [2] is used to locate closest points, leading to the same algorithmic complexity. The following result holds

$$d_H(M,N) \leq \delta_H(M,N) \leq d_H(M,N) + \max\{D(M), D(N)\}.$$

where $D(M)$ is the maximum segment length $\|\mathbf{p}_i - \mathbf{p}_{i+1}\|$ and $D(N)$ is the maximum segment length $\|\mathbf{q}_i - \mathbf{q}_{i+1}\|$. In §2.3, the accuracy ϵ of the polygonal representation is supposed to be way smaller than the actual distance between the CAD and the mesh in such a way that

$$|d_H(M,N) - d_H(C,C_m)| = O(\epsilon).$$

In consequense, the discrete distance will only be accurate enough if and only if

$$\max\{D(M), D(N)\} < \epsilon$$

which means that oversampling should be applied on the discrete curve. This is clearly a problem in our context where a relative accuracy of 10^{-6} is often required.

The discrete Fréchet distance $\delta_F(M,N)$ [7] considers only positions of the leash where its endpoints are located at vertices of the two polygonal curves and never in the interior of an edge. The discrete Fréchet distance can be computed in polynomial time i.e. in $O(mn)$ operations using a simple dynamic programming algorithm that is described in [7]. It is very difficult to find a sub-quadratic algorithm that computes the discrete Fréchet distance. We have again that

$$d_F(M,N) \leq \delta_F(M,N) \leq d_F(M,N) + \max\{D(M), D(N)\}.$$

Computing the discrete Fréchet distance may be out of reach if massive oversampling is applied.

2.5. Direct distances between polygonal curves

The computation of the direct Hausdorff distance between two polygonal curves is related to the Voronoi diagram of the line segments. The distance can only occur at points that are either endpoints of line segments or intersection points of the Voronoi diagram of one of the sets with a segment of the other. This observation leads us to the following quadratic algorithm for computing the direct Hausdorff distance $d_H(M,N)$.

- Compute the bissector of all possible pairs of segments of polygonal curve N. Two line segment have a bisector with up to 7 arcs (lines and parabolas). Store all arcs in a list.
- Compute the intersections of each arc with M.
- Compute the distance between those intersection points and N. The one sided Hausdorff distance is the maximum of those distances.

The Voronoi diagram of line segments could be theoretically computed in $O((n+m)\log(n+m))$ operations [1]. Yet, it involves the computation of the whole Voronoi diagram. To our knowledge, few robust implementations of Voronoi diagrams of line segments exist [8] and no extension to higher dimensions than two has been proposed to date.

It is possible to compute the Fréchet distance $d_F(M,N)$ between two polygonal curves in $O(mn\log(mn))$ operations [1]. The algorithm is even more complex that for direct Hausdorff distance.

2.6. An area-based distance

It was mentioned in the previous section that, even thought it was possible to computing both the Frechet distance and the Hausdorff distance with a guaranteed accuracy, the computational time required for computing such distance was not compatible with a mesh generation procedure. Nevertheless, the accurate computation of the Hausdorff distance will serve as a validation for the alternative method that is used in our computations.

It is important to notice that C and C_m coincide at mesh vertices $\boldsymbol{x}_i$, $i = 0,\ldots,p$, i.e. at their end points $\boldsymbol{x}_0$ and $\boldsymbol{x}_p$ and at all internal vertices.

We consider the polygonal curves M and N going from $\boldsymbol{x}_k$ to $\boldsymbol{x}_{k+1}$. For each point of $\mathbf{p}_i$, $i = 1,\ldots,m-1$ of M, one extra point is added to N at the same relative linear abscissa. The same is done for N (see Figure 7). The area A

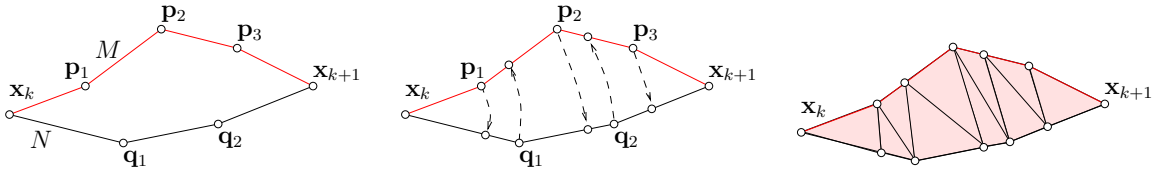

Fig. 7. Two discrete curves (left) where additional sampling points have been added (center) in order to easily compute the area distance (right)

between M and N is then computed as the sum of triangle areas constructed with the points on M and N (see Figure 7). The area-based distance is then computed as the ratio between A and the average length of M and N:

$$d_A = \frac{2A}{|M| + |N|}.$$

The distance d_A measures the average distance between M and N.

2.7. Mesh optimization

In a recent paper [13], a technique that allows to untangle high order/curvilinear meshes is presented. The technique makes use of unconstrained optimization where element Jacobians are constrained to lie in a prescribed range through moving log-barriers. The untangling procedure starts from a possibly invalid curvilinear mesh and moves mesh vertices with the objective of producing elements that all have bounded Jacobians. Bounds on Jacobians are computed using results of papers [9,10]. In what follows, we have extended the objective function $f(\boldsymbol{x}_i^e)$ that is employed in an unconstrained optimization procedure to untangle invalid curved elements. Here $\boldsymbol{x}_i^e$ is the position of the ith node of element e. The objective function

$$f = \mathcal{E} + \mathcal{F} + \mathcal{D}$$

is composed of three parts $\mathcal{E}$, $\mathcal{F}$ and $\mathcal{D}$.

The first part $\mathcal{E}$ relies on the assumption that the method is provided with a straight-sided mesh of high quality. This mesh has potentially been defined to satisfy multiple criteria, such as a predetermined size field, or anisotropic adaptation. The conversion of such meshes to high order is expected to preserve as much as possible all these features. Therefore, the nodes shall be kept as close as possible to their initial location in the straight sided mesh. In this work, the definition of $\mathcal{E}$ is the one of [13] i.e.

$$\mathcal{E}(\boldsymbol{x}_i, K_{\mathcal{E}}) = \frac{K_{\mathcal{E}}}{L^2} \sum_e \sum_i (\boldsymbol{x}_i^e - \boldsymbol{X}_i^e)^2 \tag{4}$$

with $\boldsymbol{X}_i^e$ the straight sided position of the ith node of element e and $K_{\mathcal{E}}$ a non dimensional constant and L is the characteristic size of the problem.

The second part $\mathcal{F}$ of the functional controls the positivity of the Jacobian. A log barrier [13] prevent Jacobians from becoming too small:

$$\mathcal{F}(\boldsymbol{x}_i, \epsilon) = \sum_{e=1} \sum_{l=1} F_l^e(\boldsymbol{x}_i^e, \epsilon)$$

with l iterating on all coefficients B_l^e of Bézier expansion of the jacobian of e and where

$$F_l^e(\boldsymbol{x}_i^e, \epsilon) = \left[\log\left(\frac{B_l^e(\boldsymbol{x}_i^e) - \epsilon J_0^e}{J_0^e - \epsilon J_0^e} \right) \right]^2 \tag{5}$$

is the log barrier function that is defined in such a way that $\mathcal{F}$ blows up when $B_l^e = \epsilon J_0^e$, J_0^e being the constant straight sided jacobian of element e, but still vanishes whenever $B_l^e = J_0^e$.

The third part $\mathcal{D}$ of the functional controls the distance between the curvilinear mesh and the geometrical model. We simply use

$$\mathcal{D}(\boldsymbol{x}_i^e, K_{\mathcal{D}}) = \frac{K_{\mathcal{D}}}{L} \sum_e d_A^e(\boldsymbol{x}_i^e)$$

where d_a^e is the area-distance restricted to element e. Derivatives of d_A with respect to $\boldsymbol{x}_i^e$ are computed using finite differences.

The value of $K_{\mathcal{E}}$ has little influence on results. The presence of $\mathcal{E}$ prevents the problem from being under-determined, and it orients the optimization procedure towards a solution that tends to preserve the straight-sided mesh.

On the other hand, the choice of a suitable value for $K_{\mathcal{D}}$ will definitively have an influence on the solution. A good balance between the control of jacobians and of the distance should be achieved. Assume that l is the typical area distance that we aim to achieve locally ($d_A^e \simeq l$). Assume then that scaled jacobians greater than δ are acceptable. Then, at this point, we should have

$$K_{\mathcal{D}} \gg \frac{L}{l} \left[\log\left(\frac{\delta - \epsilon}{1 - \epsilon} \right) \right]^2 .$$

As an example, if $\epsilon = 0.2$, $\delta = 0.4$, $L = 1$ and $l = 10^{-4}$, then $K_{\mathcal{D}}$ should be much larger than 3624. Here, we choose $K_{\mathcal{D}}$ to be 10 time higher.

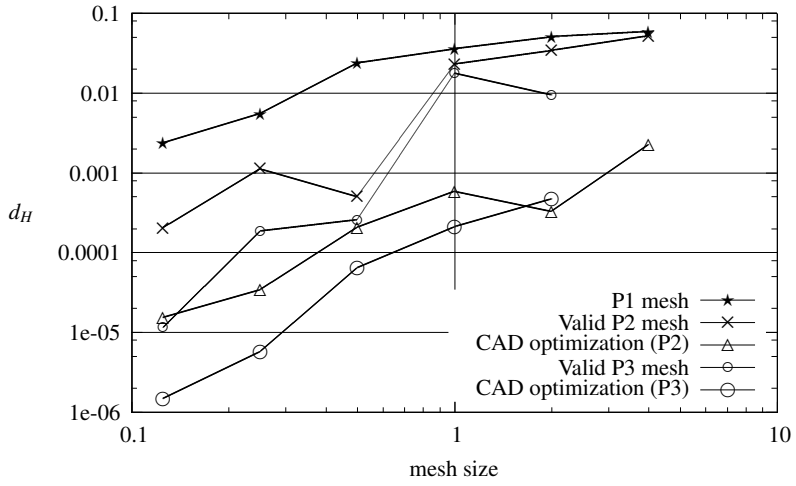

Fig. 8. Hausdorff distance between the geometry of the NACA0012 profile and the meshes.

3. Examples

3.1. NACA0012

We consider the classical geometry of the NACA0012 wing and a sequence of 6 uniform meshes with a mesh size of respectively 4, 2, 1, 1/2, 1/4 and 1/8. The area distance d_A has been used in the optimization process. Figure 8 presents the evolution of the Hausdorff distance d_H between the mesh and the model. Optimizing the CAD-to-mesh distance allows to reduce d_H of at least one order of magnitude in each case. In the coarser meshes, quadratic meshes are not better than straight sided meshes with respect to the CAD-to-mesh distance. Optimizing coarse meshes allow to reduce d_H of two orders of magnitude. Note also that optimized quadratic meshes are usually geometrically more accurate than cubic meshes. This is indeed a good news: it is well known in the finite element community that using curvilinear meshes can harm approximation properties of high order finite elements [5].

Examples of meshes are depicted on Figure 9. In the optimization process, mesh vertices classified on the model are able to move on the CAD curve. The high order vertices of the final optimized mesh (colored in pink on the Figures) are not always located at mid-distance between the corner nodes (colored in black). We have solved the Euler equations on three meshes corresponding to a mesh size of 0.6 with a standard setup corresponding to an outflow at Mach number $Ma = 0.5$ and an angle of attack of three degrees. A high order ($p = 7$ i.e. seventh order polynomials) discontinuous Galerkin scheme has been used for the spatial discretization and a Newton-Krylov iterative scheme with a pseudo time has been used for reaching the steady state. About 15 pseudo-time iterations were necessary to reduce the residual by 8 orders of magnitude.

The results are depicted on Figure 10. The solution on the valid quadratic mesh is not way better than the one on the straight sided mesh while the solution on the CAD optimized mesh is smooth and clean.

It is important to note that curvilinear meshes with high order mappings may not be appropriated for computations: (i) high order mapping requires costly integration procedures to account for the high order jacobians and (ii) the use of high order mappings may harm the convergence of the finite element approximation [5]. Here, we show that quadratic mappings are able to approximate the NACA0012 profile with a very high accuracy. Indeed, the accuracy that is obtained here ($d_H = 2 \times 10^{-4}$ for a wing span of $L = 1$) is lower than manufacturing precision and is obviously sufficient to accomodate very high order polynomials (seventh order here) for approximating the unknown fields.

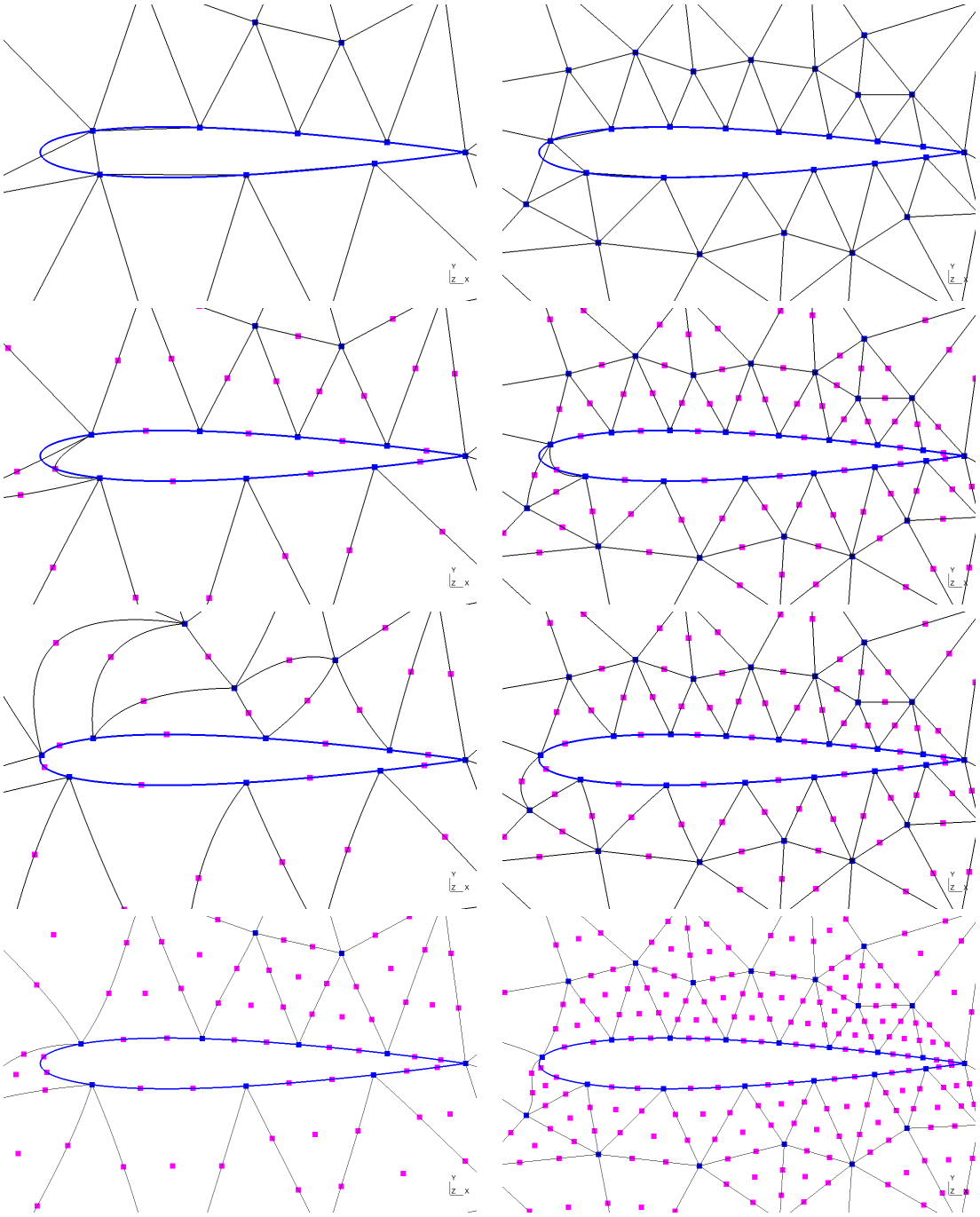

Fig. 9. Two of the 6 meshes of the NACA0012 profile corresponding to mesh sizes 2 (left) and 1 (right). Top figures present the straight side meshes. Second row shows the valid quadratic meshes. Bottom figures show the CAD optimized P2 and P3 meshes.

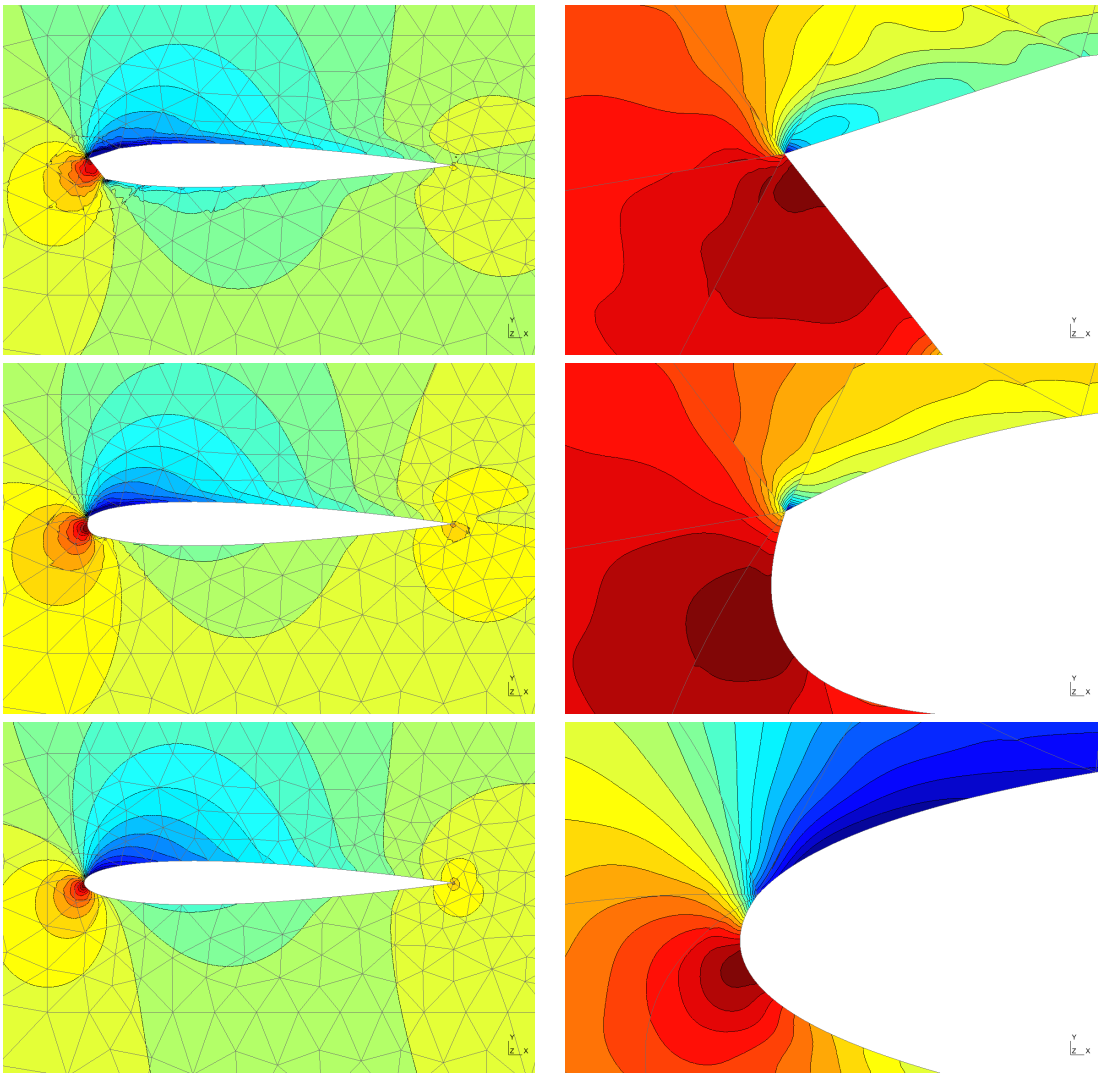

Fig. 10. Density color map for a $Ma = 0.5$ flow around the NACA0012 wing with an angle of attack of three degrees. Top figures present the solution on the straight sided mesh. Middle figures show the results on the valid quadratic meshes. Bottom figures show the solution on the CAD optimized quadratic mesh.

3.2. Three component wing

Figure 11 presents a quadratic mesh of the three component wing. This geometry is used in one of the test cases for the high order workshop (`http://www.as.dlr.de/hiocfd/case_c3.1.html`). It is is aimed at testing high-order methods for a two-dimensional turbulent flow with a complex configuration. It has been investigated previously with low order methods as part of a NASA Langley workshop. We present here preliminary results i.e. grids (no computations). In further work, boundary layer meshes will be generated that will allow RANS computations to be performed.

Close up on two quadratic grids are presented on Figure 12. CAD to mesh distance was reduced by more than one order of magnitude in all cases.

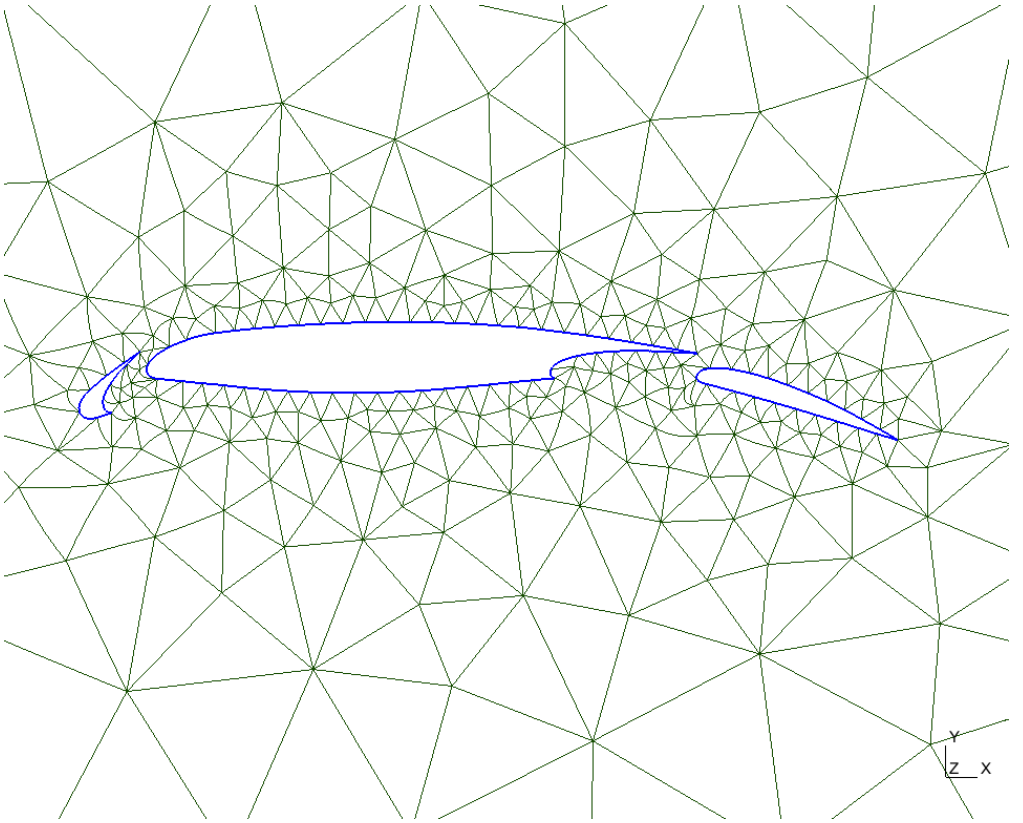

Fig. 11. Quadratic mesh of the three component wing.

4. Conclusions

In this paper, we have shown that CAD to mesh distance can be dramatically reduced using an unconstrained optimization procedure. The optimized quadratic meshes that were generated were sufficiently accurate geometrically so that high order computations could be performed on very coarse meshes. The extension to 3D case is underway.

References

[1] H. Alt and M. Godau. Computing the fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.

[2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.

[3] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138(2):251–285, 1997.

[4] P.-E. Bernard, J.-F. Remacle, and V. Legat. Boundary discretization for high-order discontinuous Galerkin computations of tidal flows around shallow water islands. *International Journal for Numerical Methods in Fluids*, 59(5):535–557, 2009.

[5] L. Botti. Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces. *Journal of Scientific Computing*, 52(3):675–703, 2012.

[6] B. Cockburn, G. Karniadakis, and C.-W. Shu, editors. *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, Berlin, 2000. Springer.

[7] T. Eiter and H. Mannila. Computing discrete frchet distance. Technical report, Technische Universitat Wien, 1994.

[8] M. Held. Vroni: An engineering approach to the reliable and efficient computation of voronoi diagrams of points and line segments. *Computational Geometry*, 18(2):95–123, 2001.

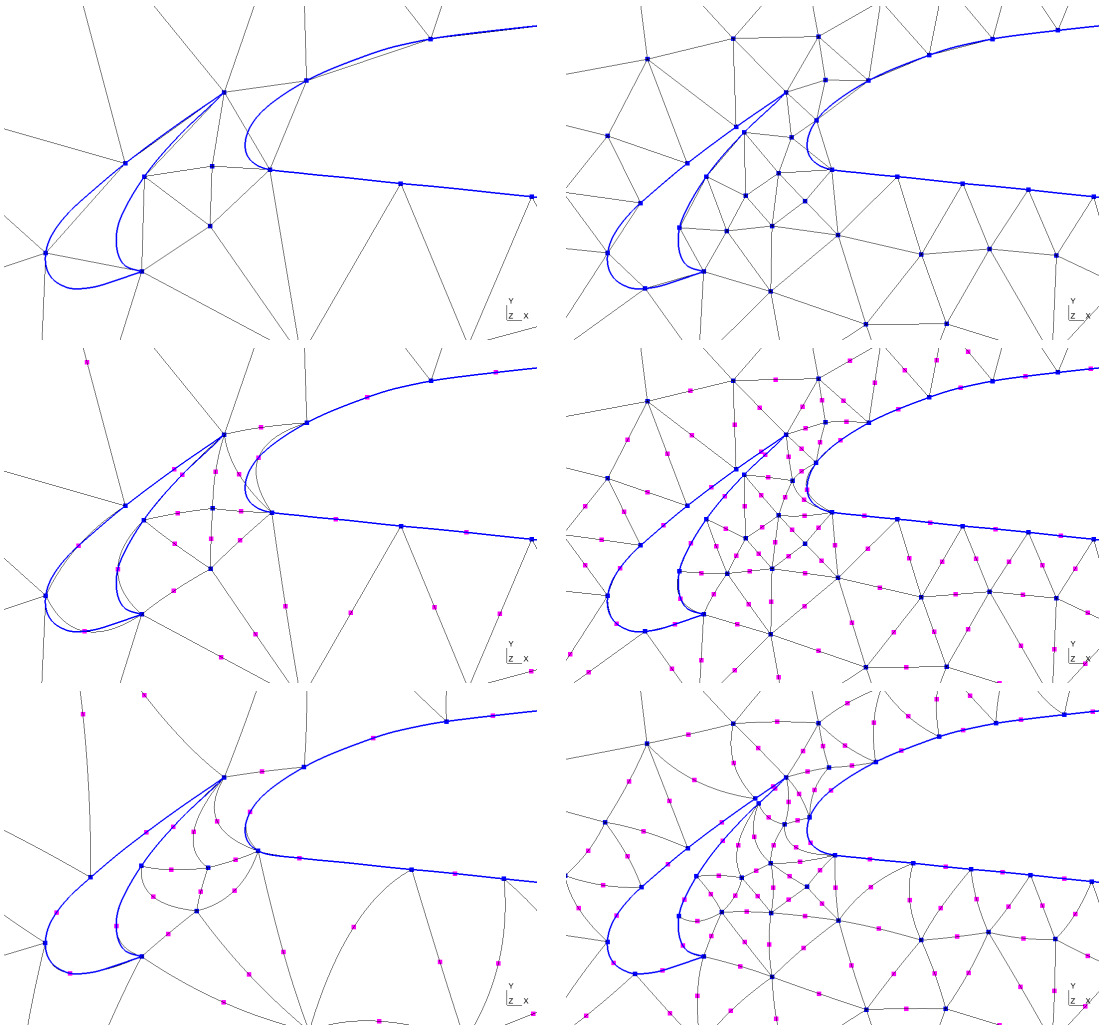

Fig. 12. Close up on the grids for the three component wing.

[9] A. Johnen, J.-F. Remacle, and C. Geuzaine. Geometrical validity of curvilinear finite elements. In W. R. Quadros, editor, *Proceedings of the 20th International Meshing Roundtable*, pages 255–271. Springer Berlin Heidelberg, 2012.

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

[11] N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. Van Der Ven, and K. Sorensen. *ADIGMA – A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Results of a Collaborative Research Project Funded by the European Union, 2006-2009*. Notes on Numerical Fluid Mechanics and Multidisciplinary Design. Springer, 2010.

[12] T. Toulorge and W. Desmet. Curved boundary treatments for the discontinuous Galerkin method applied to aeroacoustic propagation. *AIAA J.*, 48(2):479–489, 2010.

[13] T. Toulorge, C. Geuzaine, J.-F. Remacle, and J. Lambrechts. Robust untangling of curvilinear meshes. *J. Comput. Phys.*, 254:8–26, 2013.