ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

HIGH-ORDER CURVED MESH GENERATION BY USING A FINE LINEAR TARGET MESH

Verena S. Schmid[1,2], Hadrien Beriot[1], Onur Atak[1], and Gwenael Gabard[2]

[1]Siemens Industry Software, Simulation and Test Solutions
Interleuvenlaan 68,B-3001,Leuven, Belgium
e-mail: {hadrien.beriot, onur.atak}@siemens.com

[2] Institute of Sound and Vibration Research, University of Southampton
SO17 1BJ, UK
e-mail: v.s.schmid@soton.ac.uk, gabard@southampton.ac.uk

Keywords: Mesh curving, High-order, Lagrange interpolation, Hierarchic shape functions, Projection based interpolation.

Abstract.

This paper examines different mesh boundary curving algorithms in the particular case where no exact geometric description is available. The starting point for the curving is typically a coarse linear mesh, whereas the target geometry is represented by a refined linear mesh. Both can be generated using a classical linear mesh generator.

Two different boundary curving algorithms are examined. The first algorithm is based on Lagrange nodal high-order polynomial interpolation where the interpolation nodes are iteratively moved towards the target curve. Relocation steps are included in each iteration to approximately preserve the original node spacing. The second algorithm is based on hierarchic modal shape functions. In a reference frame, projection based interpolation is applied that minimizes the distance between the interpolating function and the local target function in the H^1-seminorm.

The performance and accuracy of the two methods are evaluated and compared. Thereby, the area between the target and the approximating curve is used as error measure.

In general, both methods exhibit similar levels of performance. A lower bound on the accuracy is observed that depends on the level of refinement of the target mesh. Differences lie in the applicability of the two methods. For the method based on modal interpolation, several initial requirements have to be fulfilled. The method based on nodal interpolation on the other hand is simpler but less robust. Overall, the modal interpolation approach is preferable, where applicable.

1 INTRODUCTION

High-order unstructured methods are gaining popularity for solving large scale engineering problems. These methods generally lead to increased accuracy and lower computational costs when compared to the commonly used low-order methods. While low-order methods typically rely on planar-faced elements, high-order methods call for the introduction of high-order curvilinear finite element meshes, in which the faces and edges of the elements are curved to align with the underlying geometry. The need for an accurate geometric description when employing high-order approximations was shown in the literature both analytically [1] and numerically [2].

Mesh curving algorithms are usually based on a-posteriori curving. A linear mesh is first generated, taking advantage of the robustness of the existing mesh generation algorithms. In a second step these elements are curved to better describe the geometry of the domain, e.g. [3, 4].

If the exact geometry is known, e.g. in the form of a parametrization, this curving step is relatively straightforward. For a nodal based polynomial representation of the geometry, the high-order nodes are defined in the reference space and then mapped onto the exact geometry. The node positions in the reference space influence the resulting curve and can be used to optimize the approximation or to prevent element tangling already during the curving process [3].

Surface reconstruction methods are applied where no or only little additional information is given about the exact geometry. Such problems are for instance encountered in medical imaging, where the definition of the computational domain is given exclusively by scattered data, obtained from data scanning. Examples for surface reconstruction methods are the *Weighted Averaging of Local Fittings (WALF)* and the *Continuous Moving Frame (CMF)* algorithms [5] or the piecewise quadratic mesh extrapolation based on local approximations of the curve Hessian [6].

In this paper an intermediate case is considered. An exact representation is not available, but instead a fine linear boundary mesh is considered, as could be used for a conventional low order simulation. This fine linear mesh forms the target of the curving process. A similar approach can be found in [7] where the fine representation is obtained from scattered scanning data. It is improved by a surface reconstruction based on spherigon patches [8]. The curved mesh is formed of Bézier curves and patches. Here, it is assumed that the target mesh is fine enough such that no further surface reconstruction is required. Two curving algorithms are presented. For both, the curved mesh is represented by piecewise polynomial parametrizations. For the first algorithm, nodal basis functions are used and for the second, hierarchic, modal basis functions.

Note that this paper focuses only on the curving of the boundary elements. Post-processing steps such as element untangling may still be required once the full 2D-mesh is considered.

2 METHODS

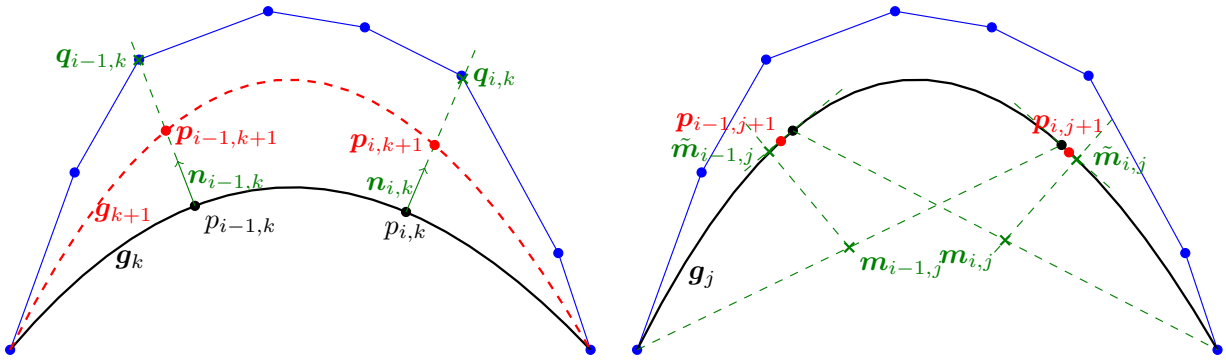
2.1 Notations

In the following, the fine target mesh is named *model mesh* or *Mmesh*. It is denoted $\mathcal{M} := (\mathcal{V}_\mathcal{M}, \mathcal{E}_\mathcal{M})$ with $\mathcal{V}_\mathcal{M} := \left\{\boldsymbol{v}_{M,1}, \boldsymbol{v}_{M,2}, \ldots, \boldsymbol{v}_{M,N^v_\mathcal{M}}\right\}$, $\boldsymbol{v}_{M,i} \in \mathbb{R}^2$ the $N^v_\mathcal{M}$ mesh vertices and $\mathcal{E}_\mathcal{M} := \left\{e_{M,1}, \ldots, e_{M,N^e_\mathcal{M}}\right\}$ the edges that connect them. The number of Mmesh edges is denoted $N^e_\mathcal{M}$.

The coarse mesh that is curved during the algorithm is named *computational mesh* or *Cmesh*. It is denoted $\mathcal{C}$ and the notations $\mathcal{V}_\mathcal{C}$, $\mathcal{E}_\mathcal{C}$, and $N^e_\mathcal{C}$ are defined analogously to $\mathcal{V}_\mathcal{M}$, $\mathcal{E}_\mathcal{M}$, and $N^e_\mathcal{M}$.

The Cmesh geometric order obtained from the mesh curving procedure is denoted q.

2.2 Nodal approach



(a) Curving step (b) Correction step

Figure 1: Schematics of the two different iteration steps in the nodal method. In both figures, the target curve $\mathcal{M}$ is shown in blue, the current $\mathcal{C}$ element at the beginning of the iteration step in black, and the result of the iteration step in red. Helplines and intersection points are drawn in green.

For the nodal method, $\mathcal{C}$ is represented by an elementwise polynomial mapping from the reference element $[-1, 1]$ to the physical space. Lagrange polynomials with equidistant nodes in the reference space form the basis of the mapping. In the physical space, corresponding nodes have to be found in order to elementwise curve the Cmesh.

As a first step, $q - 1$ high-order interpolation nodes are placed equidistantly on each edge of the linear Cmesh, forming a polynomial $\boldsymbol{g}_k : [-1, 1] \rightarrow R^2$ of order q . These nodes are iteratively moved towards the Mmesh during the curving process. The iteration consists of two operations which are implemented in a nested iteration:

1. In an outer loop of n_{iter} iteration steps, the Cmesh nodes are progressively moved towards the target geometry, following the local Cmesh normal direction.

   For the calculation of the new node positions, each node $\boldsymbol{p}_{i,k}$ is treated individually. The normal $\boldsymbol{n}_{i,k}$ of $\boldsymbol{g}_k$ at $\boldsymbol{p}_{i,k}$ is first evaluated. To find the distance to $\mathcal{M}$, the closest intersection point $\boldsymbol{q}_{i,k}$ between $\mathcal{M}$ and the line $\boldsymbol{s}_{i,k}(\alpha) := \boldsymbol{p}_{i,k} + \alpha \boldsymbol{n}_{i,k}$ is determined. The updated node position is then calculated as

   $$\boldsymbol{p}_{i,k+1} \quad = \boldsymbol{p}_{i,k} + \sigma_k \left(\boldsymbol{q}_{i,k} - \boldsymbol{p}_{i,k} \right), \tag{1}$$

   with $\sigma_k = \frac{k}{n_{\text{iter}}}$. An example of a mesh curving iteration step is illustrated in Figure 1(a)).

2. After each mesh curving iteration, several correction steps are applied. The spacing between points is readjusted, aiming for an approximately equidistant spacing in the physical space (see Figure 1(b)).

   The tangent of $\boldsymbol{g}_j$ in a node $\boldsymbol{p}_{i,j}$ is denoted $\boldsymbol{t}_{i,j}$. For each point $\boldsymbol{p}_{i,j}$, a midpoint is computed between its two neighboring points and denoted $\boldsymbol{m}_{i,j} := \frac{\boldsymbol{p}_{i-1,j} + \boldsymbol{p}_{i+1,j}}{2}$. The midpoint is then projected onto the tangent line $\boldsymbol{t}_{i,j}$. The resulting point $\tilde{\boldsymbol{m}}_{i,j}$ represents the target point towards which the node $\boldsymbol{p}_{i,j}$ is moved. All nodes are updated at the same time. In order to reduce the likelihood to overshoot, the node is updated by

   $$\boldsymbol{p}_{i,j+1} = \boldsymbol{p}_{i,j} + \tau \left(\boldsymbol{p}_{i,j} - \tilde{\boldsymbol{m}}_{i,j} \right), \tag{2}$$

   with $\tau \in (0, 1)$. In this study, $\tau = 0.4$ is chosen.

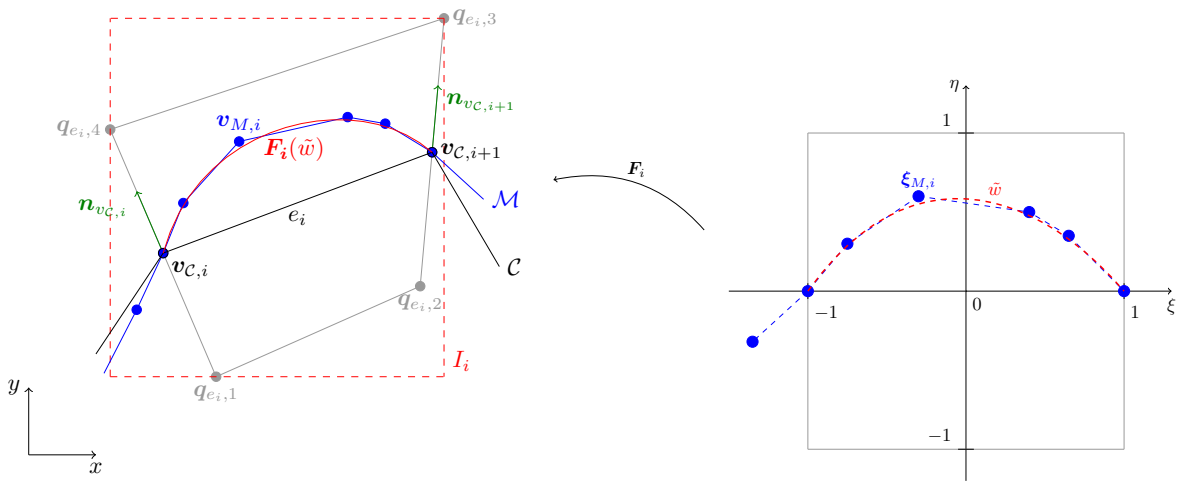


Figure 2: Schematic of the modal method. On the left, the physical space is shown with $\mathcal{M}$ in blue, $\mathcal{C}$ with edge e_i in black, the quadrangle spanned by e_i in grey, and the bounding interval I_i in red. On the right, the reference space is shown with the reference quadrangle in grey, the local target function w_i in blue and the polynomial interpolation $\tilde{w}_i$ in red.

After the last iteration step, one additional mesh deformation step is performed with $\sigma_{n_{\text{iter}}+1} = 1$ to move the nodes exactly onto the Mmesh.

The number of iteration steps for the two iteration loops are increased adaptively until a convergence of the method is observed. For both iteration loops the number of iteration steps are increased simultaneously, but those of the outer loop is increased in bigger steps. As criterion for convergence, the distance between nodes obtained with the current number of iteration steps and those obtained with the previous number of iteration steps is evaluated. Once the maximal distance is smaller than a predefined nodal tolerance of 10^{-14}, it is decided that the curving method converged. The algorithm breaks down if a predefined upper limit of iteration numbers is hit without reaching convergence. In this study, the limit is set to 200 iteration steps for the outer loop.

2.3 Modal approach

The modal method is based on a polynomial fitting in a reference space, using a hierarchic, modal polynomial basis. A schematic of the method is provided in Figure 2.

The algorithm begins as follows. In the physical space, the vertices of each Cmesh element $e_i \in \mathcal{E}_\mathcal{C}$ are extruded by an extrusion length λ_i in positive and negative edge normal direction. Connecting these extruded nodes, a quadrangle element $\mathcal{Q}_{e_i} = (\boldsymbol{q}_{e_i,1}, \boldsymbol{q}_{e_i,2}, \boldsymbol{q}_{e_i,3}, \boldsymbol{q}_{e_i,4})$ is constructed around e_i. Note that the union $\bigcup_{e_i \in \mathcal{E}_\mathcal{C}} \mathcal{Q}_{e_i}$ forms a neighborhood of $\mathcal{C}$. Furthermore, it is required that $\mathcal{M} \in \bigcup_{e_i \in \mathcal{E}_\mathcal{C}} \mathcal{Q}_{e_i}$ and $\mathcal{Q}_{e_i} \cap \mathcal{Q}_{e_j} = \emptyset$ for $i \neq j$. For each quadrangle $\mathcal{Q}_{e_i}$ the mapping $\boldsymbol{F}_i : \mathbb{R}^2 \to \mathbb{R}^2$, $\mathcal{Q}_{ref} \mapsto \mathcal{Q}_{e_i}$ is defined by

$$\begin{aligned} \boldsymbol{F}_i(\xi,\eta) = \; & \tfrac{1}{4}\left((\xi-1)(\eta-1)\boldsymbol{q}_{e_i,1} - (\xi+1)(\eta-1)\boldsymbol{q}_{e_i,2}\right. \\ & \left. +(\xi+1)(\eta+1)\boldsymbol{q}_{e_i,3} - (\xi-1)(\eta+1)\boldsymbol{q}_{e_i,4}\right), \end{aligned} \tag{3}$$

where $\mathcal{Q}_{ref}$ designates the reference quadrangle element. For all $\boldsymbol{v}_{M,i} \in \mathcal{V}_\mathcal{M}$ that lie within the bounding interval

$$I_i := \left[\min_j \left(\boldsymbol{q}_{e_i,j}\right)_x, \max_j \left(\boldsymbol{q}_{e_i,j}\right)_x\right] \times \left[\min_j \left(\boldsymbol{q}_{e_i,j}\right)_y, \max_j \left(\boldsymbol{q}_{e_i,j}\right)_y\right], \tag{4}$$

the corresponding points in the reference space

$$\Xi_{I_i} := \{\boldsymbol{\xi} = (\xi, \eta) \,|\boldsymbol{F}_i(\boldsymbol{\xi}) \in \mathcal{V}_{\mathcal{M}} \cap I_i\} \tag{5}$$

are determined. Since $\boldsymbol{F}_i$ is generally not linear, no closed-form expression could be found for the inverse mapping. Instead, the objective function $g(\xi, \eta) := |\boldsymbol{x} - \boldsymbol{F}_i(\xi, \eta)|$ is minimized to find the inverse mapping of the fixed point $\boldsymbol{x}$. The points in Ξ_{I_i} are then piecewise linearly interpolated by $w_i : \mathbb{R} \mapsto \mathbb{R}, w_i(\xi_{I_i,j}) = \eta_{I_i,j}$ for $(\xi_{I_i,j}, \eta_{I_i,j}) \in \Xi_{I_i}$. Note that this linear interpolation forms an approximation. In general, the image $\boldsymbol{F}_i(w_i)$ of w_i does not lie on $\mathcal{M}$ because of the non-linearity of $\boldsymbol{F}_i$.

The function w_i is approximated by a polynomial interpolation

$$\tilde{w}_i = \sum_{j=0}^{q} \alpha_{i,j} l_j. \tag{6}$$

The basis of the interpolation consists of Lobatto shape functions [9]

$$\begin{aligned} l_0(\xi) &= \tfrac{1-\xi}{2}, \\ l_1(\xi) &= \tfrac{\xi+1}{2}, \quad \text{and} \\ l_k(\xi) &= \tfrac{1}{\|L_{k-1}\|_2} \int_{-1}^{\xi} L_{k-1}(x)\,\mathrm{d}x, \ \text{for } 2 \le k,\ \xi \in [-1, 1]\,, \end{aligned} \tag{7}$$

with $L_i(\xi)$ the i^{th} Legendre polynomial. From the orthogonality of the Legendre polynomials in L^2, the Lobatto shape functions inherit the property

$$\int_{-1}^{1} l_i'(\xi) l_j'(\xi)\,\mathrm{d}\xi = \delta_{ij}, \ \text{with } \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases}. \tag{8}$$

This property can be used to obtain the coefficients $\alpha_{i,j}$, $j = 2, \ldots, q$ in (6) that minimize the distance between the $\tilde{w}_i$ and w_i in the H^1-seminorm [9]

$$\left|w_i - \tilde{w}_i^v - \tilde{w}_i^b\right|_{H^1} \to \min\,. \tag{9}$$

The function $\tilde{w}_i$ is thereby splitted into the linear part $\tilde{w}_i^v = \sum_{j=0}^{1} \alpha_{i,j} l_j$ and the high-order part $\tilde{w}_i^b = \sum_{j=2}^{q} \alpha_{i,j} l_j$. Note that $\alpha_{i,1}$ and $\alpha_{i,2}$ are uniquely defined by $w_i(-1)$ and $w_i(1)$. Therefore, the optimization only considers $\alpha_{i,2}, \ldots, \alpha_{i,q}$.

The minimization (9) is equivalent to

$$\int_{-1}^{1} \left(w_i - \tilde{w}_i^v - \tilde{w}_i^b\right)' l_j'\,\mathrm{d}\xi = 0, \ \text{for } j = 2, \ldots, q, \tag{10}$$

which can be reformulated as

$$\alpha_{i,j} = \int_{-1}^{1} w_i'(\xi) l_j'(\xi)\,\mathrm{d}\xi. \tag{11}$$

This approach is referred to as projection based interpolation [9]. Lobatto shape functions are also hierarchic, meaning that $\mathcal{B}^{n-1} \subset \mathcal{B}^n$, with $\mathcal{B}^n$ the Lobatto basis of order n. This is beneficial for adaptive schemes, because it implies that a change in the order q does not require a recalculation of the shape functions l_j or coefficients α_j.

3 NUMERICAL TESTS

3.1 Geometry and Meshes

The test case geometry is based on an ellipse with a high aspect ratio. An additional cosine term is included to introduce bumps along the curve:

$$\begin{pmatrix} x(\theta) \\ y(\theta) \end{pmatrix} = \begin{pmatrix} (r_x + h_b \cos(\theta n_b)) \cos(\theta) \\ (r_y + h_b \cos(\theta n_b)) \sin(\theta) \end{pmatrix} \quad \text{for } \theta \in [0, 2\pi]\,, \tag{12}$$

where n_b is the number of bumps and h_b controls their height. The parameters were set to $r_x = 0.5$m, $r_y = 5$m, $h_b = 0.1$m, and $n_b = 10$. A plot of the geometry can be seen in Figure 3(a).

In a first step, n_v equidistant nodes $\tilde{\theta}_i \in [0, \pi]$ are defined in the reference space. In a second step, perturbed nodes are introduced, $\theta_i = \tilde{\theta}_i + \gamma_i$, with γ_i uniformly distributed pseudo random values in the interval $[-\frac{1}{3n_v}, \frac{1}{3n_v}]$ and $\gamma_0 = \gamma_{n_v} = 0$.

As a last step before the actual curving, the Cmesh vertices are snapped onto the closest Mmesh vertices. This ensures that the assumption $\mathcal{V}_\mathcal{C} \subset \mathcal{V}_\mathcal{M}$ holds.

3.2 Evaluation Method

To assess the accuracy of the two algorithms, the area between the final high-order curve and the target curve $\mathcal{M}$ needs to be evaluated. The area between the two curves is divided in small quadrangles $Q_i = (\boldsymbol{q}_{i,1}, \boldsymbol{q}_{i,2}, \boldsymbol{q}_{i,3}, \boldsymbol{q}_{i,4})$. For each quadrangle, the area is calculated with the formula

$$A_{Q_i} = \frac{1}{2} \left| (\boldsymbol{q}_{i,2} - \boldsymbol{q}_{i,4}) \times (\boldsymbol{q}_{i,1} - \boldsymbol{q}_{i,3}) \right| \tag{13}$$

for convex quadrangles. The size of the quadrangles is adapted in each case to ensure an accurate evaluation of the area. Restrictions on the node positions are applied to reduce the number of non-convex quadrangles. The area based error measure is defined as

$$A_Q := \frac{\sum_i A_{Q_i}}{l_\mathcal{M}}, \tag{14}$$

with $l_\mathcal{M}$ the total length of $\mathcal{M}$. This corresponds to the measure suggested in [4].

3.3 Fixed number of Mmesh elements

The first study examines the influence of the interpolation order q and the size of $\mathcal{C}$ on the results of the curving process. The target curve $\mathcal{M}$ is fixed with $N^e_\mathcal{M} = 998$.

The curving procedures are performed for varying $N^e_\mathcal{C} \in \{6, 10, \ldots, 78\}$ and $q \in \{1, \ldots 8\}$ for the nodal approach and $N^e_\mathcal{C} \in \{10, 14, \ldots, 78\}$ and $q \in \{1, \ldots 15\}$ for the modal approach. Figures 3(b)–3(f) provide an example of the initial $\mathcal{C}$ and the final curved mesh with both methods for $N^e_\mathcal{C} = 10$ and $q = 4$,

Note that the different bounds of the parameter show the limitations of the two methods. The nodal curving algorithm failed to converge within iteration boundaries for some cases of $N^e_\mathcal{C}$ for $q = 8$. Calculations for $q = 9$ did not converge for any of the tested $N^e_\mathcal{C}$. The cases that did not converge are not included in the upcoming plots.

The modal curving approach on the other hand could be applied up to $q = 15$, with no indication that even higher orders would not be calculable. The limitation for the modal method

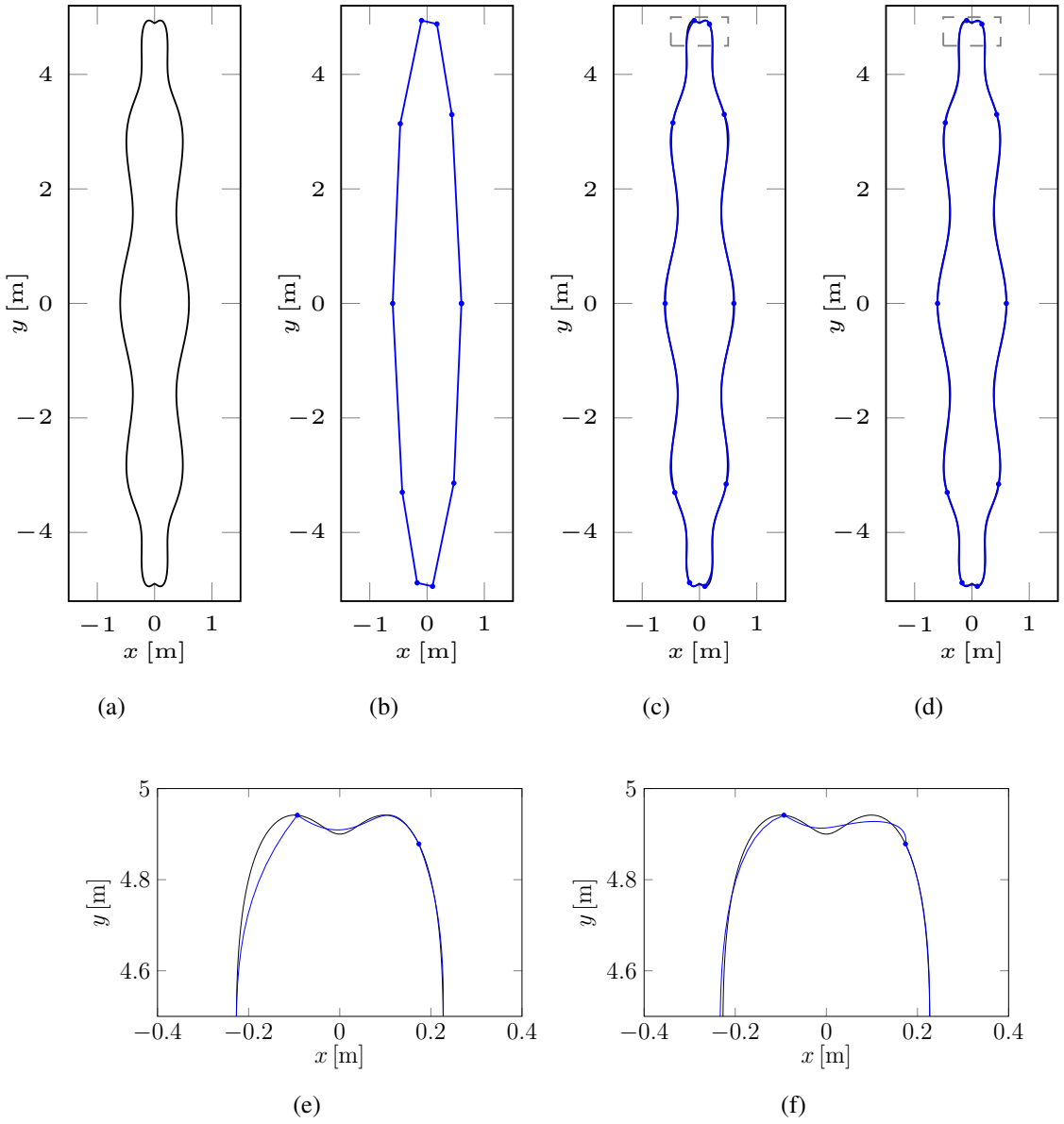
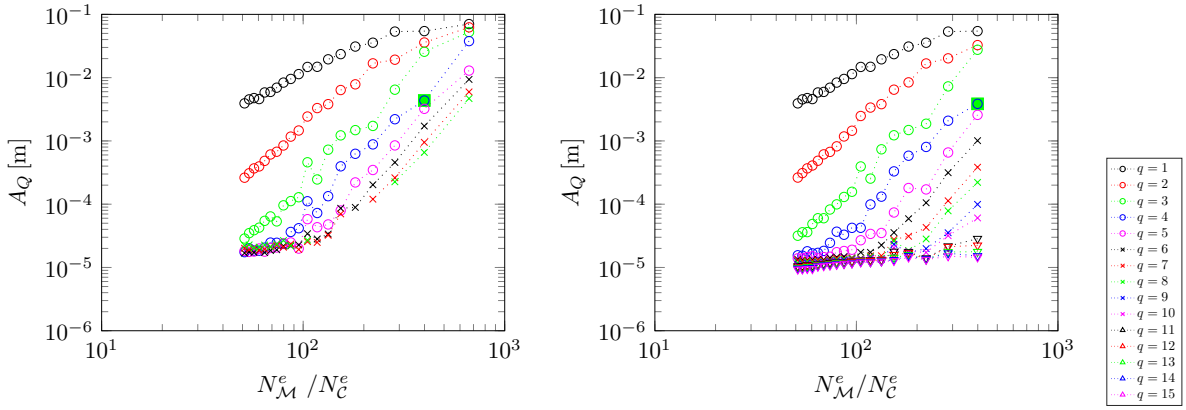
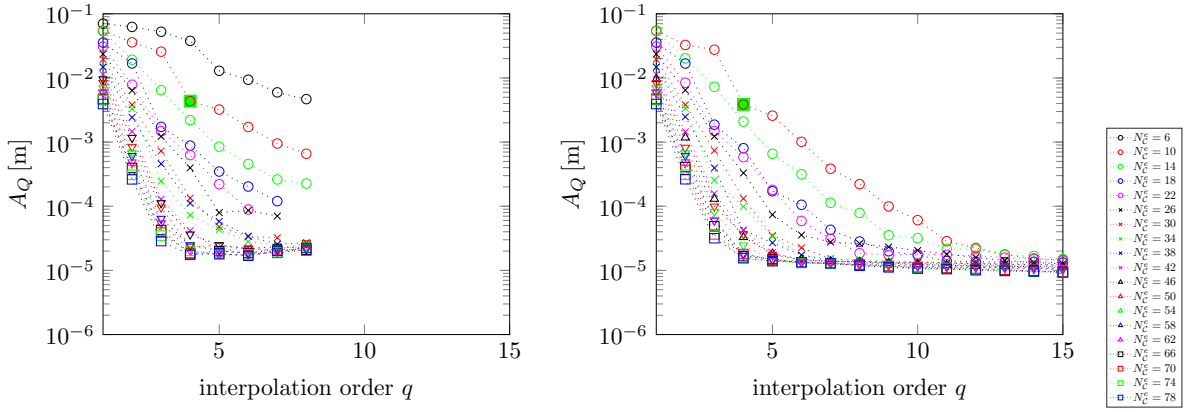


Figure 3: The distorted ellipse test case. (a) geometry, (b) Cmesh with $N_{\mathcal{C}}^{e} = 10$. The resulting curves for $q = 4$ after applying (c) the nodal curving approach or (d) the modal curving approach are shown in blue. The grey dashed boxes mark the zoom-in region that is shown in (e) for the nodal curving approach and in (f) for the modal curving approach.



(a) (b)

Figure 4: $N^e_{\mathcal{M}}/N^e_{\mathcal{C}}$-convergence for (a) the nodal method and (b) the modal method, with fixed $N^e_{\mathcal{M}} = 998$.



(a) (b)

Figure 5: q-convergence for (a) the nodal method and (b) the modal method, with fixed $N^e_{\mathcal{M}} = 998$.

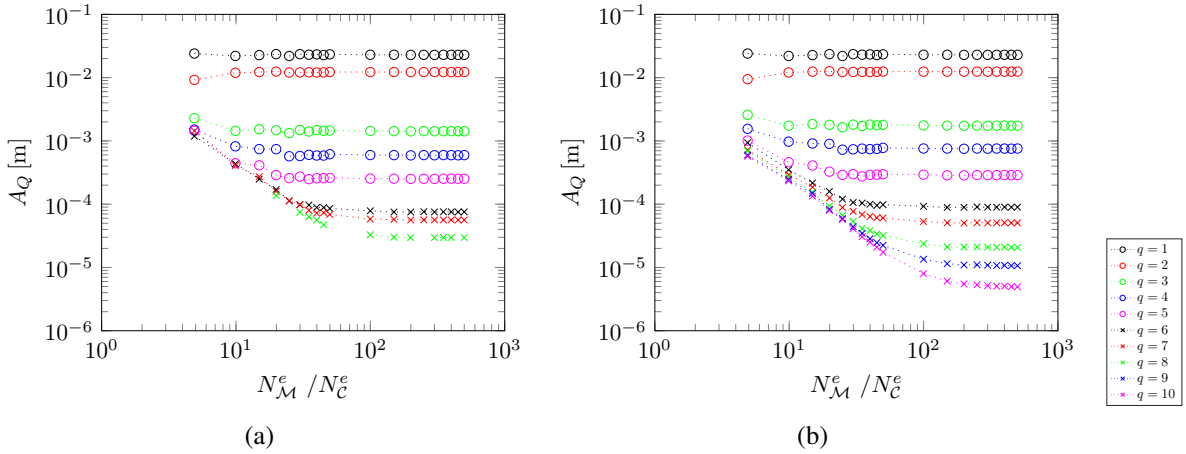
lies in the construction of the quadrangles $\mathcal{Q}_{e_i}$. For the case $N^e_{\mathcal{C}} = 6$, no set of extrusion lengths $\Lambda = \{\lambda_1, \ldots, \lambda_{N^e_{\mathcal{C}}}\}$ could be found such that the requirements $\mathcal{M} \in \bigcup_{e_i \in \mathcal{E}_{\mathcal{C}}} \mathcal{Q}_{e_i}$ and $\mathcal{Q}_{e_i} \cap \mathcal{Q}_{e_j} = \emptyset$ for $i \neq j$ were fulfilled.

In Figure 4, the $N^e_{\mathcal{M}}/N^e_{\mathcal{C}}$ convergence of the area based error measure is shown for different interpolation orders q. Each data point corresponds to one parameter set $(N^e_{\mathcal{C}}, q)$. The example case from Figures 3(b)–3(f) is highlighted by a green square.

The results indicate that both methods provide a similar behavior. For a fixed order q, the $\mathcal{C}$ approximation convergence increases with q following roughly $\mathcal{O}(q)$ up to $q = 3$. For higher orders q, the convergence order lies between 3 and 4. However, all approximations are limited in accuracy, and stagnate around $A_Q = 10^{-5}$.

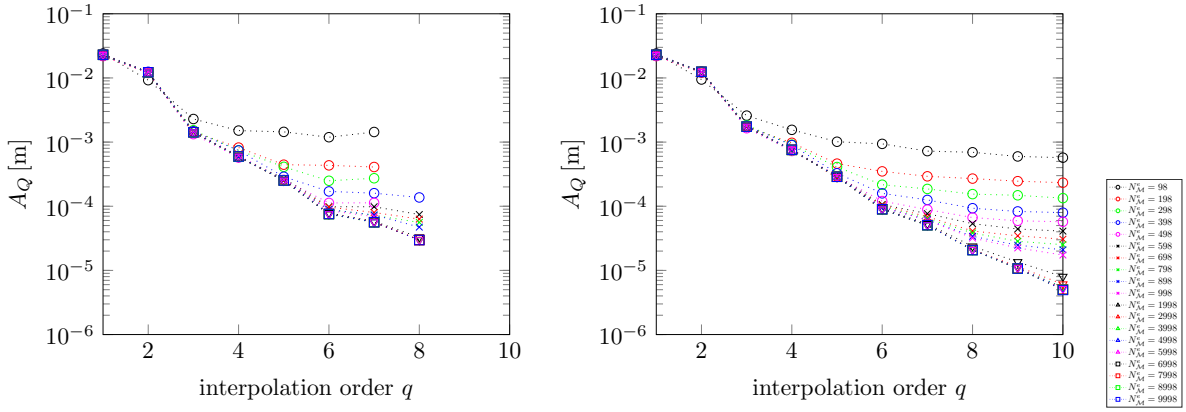
The corresponding q-convergence curves for different $N^e_{\mathcal{C}}$ are shown in Figure 5. As in the previous figure, the example case from Figures 3(b)–3(d) is highlighted by a green square.

With increasing q the error decreases until it stagnates. The observed bound is about the same for both methods. This seems to indicate that the limit in accuracy is caused by the choice of the $\mathcal{M}$ representation, more specifically by the fact that $\mathcal{M}$ is only C^0 continuous. This typically limits the accuracy up to which $\mathcal{M}$ can be polynomially interpolated.



(a) (b)

Figure 6: $^{N^e_{\mathcal{M}}}/_{N^e_{\mathcal{C}}}$-convergence for (a) the nodal method and (b) the modal method, with fixed $N^e_{\mathcal{C}} = 20$.



(a) (b)

Figure 7: q-convergence for (a) the nodal method and (b) the modal method, with fixed $N^e_{\mathcal{C}} = 20$.

3.4 Fixed number of Cmesh elements

A second study is performed in order to better understand the influence of $N^e_{\mathcal{M}}$ on the error of the approximation. The size of the Cmesh is now fixed with $N^e_{\mathcal{C}} = 20$, and the target Mmesh is varied. The parameter sets are $q \in \{1, \dots, 8\}$ for the nodal and $q \in \{1, \dots, 10\}$ for the modal algorithm, and $N^e_{\mathcal{M}} \in \{98, 198, \dots, 998, 1998, \dots, 9998\}$ for both methods. Again, the lower q limit for the nodal method as well as missing data points are caused by convergence issues.

Figure 6 shows the $^{N^e_{\mathcal{M}}}/_{N^e_{\mathcal{C}}}$-convergence for different interpolation orders q. Expectedly, as $\mathcal{M}$ is refined, the error first decreases and then stagnates. The stagnation levels decrease as q increases. It shows that with a fixed order q, $\mathcal{M}$ refinement is useful only up to a certain point.

Figure 7 shows the corresponding q-convergence curves for different $N^e_{\mathcal{M}}$. For low interpolation orders q, the $\mathcal{M}$-refinement has no impact on the performance of the mesh curving algorithm. But for larger values of q, the error stagnates for the coarser Mmeshes. The lower $N^e_{\mathcal{M}}$, the higher the stagnation level. This confirms the assumption from above that the stagnation is introduced by the C^0 continuity of $\mathcal{M}$. A finer $\mathcal{M}$ provides a better representation of the exact geometry and is in a colloquial sense smoother than a coarse $\mathcal{M}$. This highlights the fact that the order of approximation q should always be consistent with the quality of the Mmesh. After a certain point, since the information at hand on the target geometry is not satisfactory

enough, increasing the order and/or refining the Cmesh does not allow to improve geometric representation.

4 DISCUSSION AND CONCLUSIONS

In this paper, high-order mesh curving algorithms were considered in the case of limited knowledge of the original geometry, more precisely in the case where the original geometry is only given in the form of a refined linear mesh. Two approaches were proposed and tested on a distorted ellipse. The results for both methods indicate that, besides the number of edges $N_{\mathcal{C}}^e$ and the interpolation order q of the curved mesh, also the level of refinement of the target curve $\mathcal{M}$ provides a limit to the accuracy of the methods. The bound introduced by $N_{\mathcal{M}}^e$ is explained by the C^0-continuity of $\mathcal{M}$ that, even though it never becomes C^1 continuous, becomes smoother in a colloquial sense as $N_{\mathcal{M}}^e$ is increased.

In general, the two methods provided similar results, but both showed disadvantages as compared to the other. For the nodal curving algorithm, convergence issues were observed for high q values. Other node spacings such as Legendre or Chebyshev nodes are currently considered to test if these would provide better convergence behavior. The modal approach, on the other hand, is not applicable in some cases, due to the difficulty to construct a consistent reference frame. This might be overcome by small changes in $\mathcal{C}$ such as shifting of vertices or refinement of the affected elements. Concerning possible applications to more complex geometries as well as the extension to 3D, the modal curving algorithm seems more promising.

In general, the two discussed curving methods consider only boundary curves. Problems such as element tangling that arise when domain elements are included, were not examined in this study. Furthermore, geometric feature detection is required before the methods could be applied to real life geometries.

5 ACKNOWLEDGMENT

This work was performed as part of the CRANE project (Community and Ramp Aircraft NoisE, www.crane-eid.eu, GA: 606844) funded by the European Union under the Framework Programme 7. The authors acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

REFERENCES

[1] S. Dey, M.S. Shephard, J.E. Flaherty, Geometry representation issues associated with p-version finite element computations. *Computer Methods in Applied Mechanics and Engineering*, **150(1)**, 39–55, 1997.

[2] X.J. Luo, M.A. Shephard, J.-F. Remacle, The influence of geometric approximation on the accuracy of high order methods. *Rensselaer SCOREC report*, **1**, 2001.

[3] Z. Xie, R. Sevilla, O. Hassan, K. Morgen, The generation of arbitrary order curved meshes for 3D finite element analysis. *Computational Mechanics*, **51(3)**,361-374, 2013.

[4] J.-F. Remacle, J. Lambrechts, C. Geuzaine, and T. Toulorge, Optimizing the geometrical accuracy of 2D curvilinear meshes. *Procedia Engineering*, **82**, 228–239, 2014

[5] X. Jiao, D. Wang, Reconstructing high-order surfaces for meshing. *Engineering with Computers*, **28(4)**, 361–373, 2012.

[6] Y.V. Vassilevski, V.G. Dyadechko, K.N. Lipnikov, Hessian-based anisotropic mesh adaptation in domains with discrete boundaries. *Russian Journal of Numerical Analysis and Mathematical Modelling*, **20(4)**, 391–402, 2005.

[7] K. Bock, J. Stiller, Generation of High-Order Polynomial Patches from Scattered Data.Spectral and High Order *Methods for Partial Differential Equations - ICOSAHOM 2012, Lecture Notes in Computational Science and Engineering*, **95**, 157–167, 2014.

[8] P. Volino, N.M. Thalmann, The SPHERIGON: a simple polygon patch for smoothing quickly your polygonal meshes. *Computer Animation 98. Proceedings*, 72–78, 1998.

[9] P. Šolín, K. Segeth, I. Doležel, *Higher order finite element methods*. Chapman & Hall / CRC, ISBN 1-58488-438-X, 2004.