

Imposing boundary conditions to match a CAD virtual geometry for the mesh curving problem

Eloi Ruiz-Gironés* and Xevi Roca

Abstract We present a high-order mesh curving method where the mesh boundary is enforced to match a target virtual geometry. Our method has the unique capability to allow curved elements to span and slide on top of several CAD entities during the mesh curving process. The main advantage is that small angles or small patches of the CAD model do not compromise the topology, quality and size of the boundary elements. We associate each high-order boundary node to a unique group of either curves (virtual wires) or surfaces (virtual shell). Then, we deform the volume elements to accommodate the boundary curvature, while the boundary condition is enforced with a penalty method. At each iteration of the penalty method, the boundary condition is updated by projecting the boundary interpolative nodes of the previous iteration on top of the corresponding virtual entities. The method is suitable to curve meshes featuring non-uniform isotropic and highly stretched elements while matching a given virtual geometry.

1 Introduction

Curved high-order meshes are required for unstructured high-order methods in order to keep their advantages [1–5]. These advantages come in the form of geometrical flexibility, high accuracy, and low numerical dissipation and dispersion. High-order methods feature exponential converge rates and therefore, they have been proved to

**Corresponding author*

Eloi Ruiz-Gironés

Computer Applications in Science and Engineering, Barcelona Supercomputing Center, 08034 Barcelona, Spain. e-mail: eloi.ruizgirones@bsc.es

Xevi Roca

Computer Applications in Science and Engineering, Barcelona Supercomputing Center, 08034 Barcelona, Spain.

be faster than low-order methods in several applications [6–14], especially in those problems where an implicit solver is required [15].

Usually, to generate a curved high-order mesh an *a posteriori* approach is used [16–25]. First, a linear mesh with elements of the desired shape and size is generated and then, the mesh boundary is curved to match the target geometry. This step may introduce low-quality and inverted elements that have to be repaired using a high-order mesh curving technique. There are several manners to formulate the mesh curving problem: PDE-based methods like solid mechanics analogies [21,24,26–29] or the Winslow equation [25], and optimization-based methods [23,30–32]

One of the key points of all these methods is the imposition of the boundary displacement, because it drives the insertion of invalid elements that might hamper the convergence and robustness of the mesh curving algorithm. In the most simple scenario, the boundary nodes are directly projected onto the target geometry to interpolate it. In other approaches, after the nodal projection, the nodes can slide along the single geometric entity they belong to [30,33–36]. Thus, the boundary condition is introduced into the problem formulation by means of the parametric coordinates of the nodes. Although inverted and low-quality elements may still appear in the first stages of the optimization process, the additional freedom of the boundary nodes allows obtaining a *better* mesh that interpolates the target geometry.

Instead of directly moving the boundary nodes to the target position, it is possible to introduce the Dirichlet condition in an incremental manner [21,25,27,28]. In this way, it is possible to mitigate the insertion of inverted elements during the optimization process and therefore, we increase the practical robustness of the mesh curving method. The key ingredient is to define the boundary trajectory and the number of sub-steps to obtain a valid boundary mesh during the curving process.

Alternatively, it is possible to pose the mesh curving as a constrained optimization problem. The boundary condition is introduced into the target function by means of a penalty method or an augmented Lagrangian formulation. The constraint of the problem can be introduced by imposing an interpolation condition [37], or by approximating the target geometry in a weak sense [38,39].

It is important to point out that standard curved mesh generators do not consider to span the curved elements on top of several CAD entities. Singularly, this option has been explored in the context of the NURBS-enhanced finite element method, where the curved boundary elements are fixed to span on top of several patches of the boundary representation [29]. Nevertheless, the option of sliding the curved elements on top several CAD entities has not been explored yet.

The main contribution of this work is to propose a mesh curving method that allows curved elements to span and slide on top of several CAD entities during the optimization process. Therefore, the curved mesh is not constrained by the topology of the geometric model. This is especially important when the model contains tangent curves with small inner angles that hinder the element quality, or small geometric entities that limit the element size. To accomplish this, we need to group the curves and surfaces of the model into virtual wires and shells using a virtual geometry kernel. Although we manually group the different geometric entities into virtual

wires and shells, we can perform this process in an automatic manner by checking the angles between adjacent entities.

To set up the virtual curves and wires, the element sizes, and the initial linear mesh we have used Pointwise [40]. When the distribution of high-order nodes is inserted in the mesh, each boundary node is associated to a virtual wire or virtual shell. Then, we project the boundary nodes onto the corresponding virtual entity, and the boundary condition for the mesh curving problem is defined as the interpolation of the target geometry through the projected points.

Once the boundary condition is obtained, we pose the mesh curving problem as a constrained optimization problem in which we minimize the mesh distortion [23] subject to the boundary condition. To solve this problem, we use a penalty method to transform the mesh curving problem into an unconstrained optimization problem. Note that the boundary condition depends non-linearly on the curved high-order mesh, since we project the boundary nodes. To solve this non-linear problem, we perform a fix-point iteration in which the boundary condition is defined in terms of the curved high-order mesh of the previous iteration. Thus, the boundary condition is not known *a priori*, and it is updated during the mesh curving process.

In the proposed method, we explicitly minimize the mesh distortion, and we penalize inverted elements in order to preserve the mesh validity during the whole optimization process. Thus, even if the current boundary condition defines an invalid boundary mesh, we obtain a valid volume mesh. Nevertheless, since we update the boundary condition at each iteration of the penalty method, in the presented applications we obtain a valid boundary condition at some point of the curving process.

Our previous curving formulations do not allow to deal with virtual geometry [36, 39]. We first proposed to slide the nodes along the geometric entities by expressing the position of the boundary nodes in terms of the corresponding parametric coordinates [36]. However, since the parametric coordinates are not continuous between different adjacent entities the nodes are not able to jump between adjacent entities. Then, we proposed to morph meshes by solving a constrained minimization problem where the boundary deformation is known and fixed during the whole process [39]. Accordingly, the latter approach does not allow updating the boundary condition in terms of the current iteration of the mesh curving process and therefore, it cannot deal with virtual geometry. On the contrary, in this work we weakly approximate an interpolative condition that depends non-linearly on the high-order mesh. Thus, it is the first time that we show a mesh curving formulation that can deal with virtual geometry.

The rest of the paper is structured as follows. Section 2 introduces several definitions related to the presented work. Section 3 presents the formulation of the proposed high order mesh curving methodology. Section 4 presents several examples to show the capabilities of the proposed formulation. Finally, Section 5 details the conclusions and the future work.

2 Preliminaries

2.1 Geometric model and mesh discretization

In this work, we use a geometric model, Ω , composed of several surfaces and curves, $\{\Omega_i^f\}_{i=1,\dots,n_f}$ and $\{\Omega_i^e\}_{i=1,\dots,n_e}$, respectively. Nevertheless, the given decomposition of the geometric model into these surfaces and curves may lead to low-quality meshes. This is the case when the interior angle between two adjacent curves is small, or when small entities limit the maximum element size. For this reason, we group the curves and surfaces of the geometric model into virtual wires and shells, respectively. That is,

$$\Omega_i^w = \bigcup_{j=1}^{n_i} \Omega_j^e \quad \Omega_i^s = \bigcup_{j=1}^{n_i} \Omega_j^f,$$

where Ω_i^w and Ω_i^s are the wires and shells, respectively. Note that after grouping the surfaces into shells, the shell interior curves are not used to define the wires. That is, not all the curves of the model are used to define the new wires. This *gluing* operation is common when dealing with virtual geometry engines [41, 42]. Currently, we use Pointwise [40] to generate the initial meshes since it can deal with such virtual geometry operation.

We consider that a geometric model, Ω , is defined as the union of its wires and shells in the following manner:

$$\Omega = \bigcup_{i=1}^N \Omega_i.$$

Figure 1a shows a CAD model of a sphere with surfaces that limit the maximum element quality. Note that the maximum element quality is bounded by the minimum angle between the curves, and for general CAD models, this angle can be arbitrarily small. To address this issue, we group the top and the bottom surfaces into two different shells, and the four equatorial curves that form the boundary of the shells into one wire, see Figure 1b. By joining the upper surfaces into a single shell, we are able to generate a high-quality surface element that spans several surfaces of the original model.

The geometric model, Ω , is discretized using an initial linear mesh, \mathcal{M}^l , composed of elements of the desired shape and size. The discretization is performed in such a way that

$$\mathcal{M}^l = \bigcup_{i=1}^N \mathcal{M}_i^l,$$

where \mathcal{M}_i^l is a discretization of the geometric entity Ω_i . Figure 1a shows a high-order triangle that approximates the surface of a sphere. Moreover, two edges of the triangle are approximating two different curves of the model. In this case, the

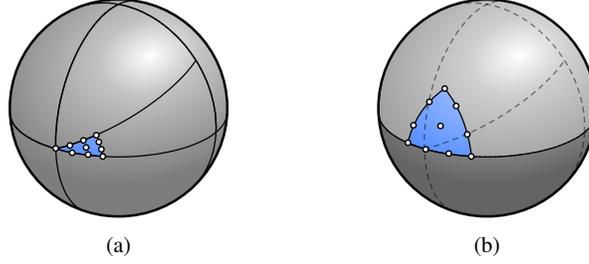


Fig. 1: High order element approximating a geometric model of a sphere: (a) with surfaces limiting the element quality; and (b) with virtual shells and wires.

triangles of the mesh are restricted to belong to a single geometric entity. In contrast, Figure 1b shows a high-order triangle that approximates a shell of the model. Note that there are high-order nodes of the triangle located in different surfaces of the geometric model. The main advantage of this procedure is that the mesh is no longer constrained to the topology of the geometric model. That is, the nodes can freely slide through different surfaces of the geometry.

2.2 Point projection algorithm

The main ingredient to perform the proposed high-order mesh curving technique is the point-projection onto the geometric model. Given a point \mathbf{x} and an entity of the geometric model, Ω_i , we want to compute the orthogonal projection of \mathbf{x} onto Ω_i that satisfies

$$\Pi_{\Omega_i}(\mathbf{x}) = \arg \min_{\mathbf{y} \in \Omega_i} \|\mathbf{x} - \mathbf{y}\|^2. \quad (1)$$

Since Ω_i is defined as a union of sub-entities of the geometric model, to compute the projection $\Pi_{\Omega_i}(\mathbf{x})$ we need to compute the projection in all the contained sub-entities and take the minimum. That is,

$$\Pi_{\Omega_i}(\mathbf{x}) = \arg \min_{\Omega_i^j \in \Omega_i} \left\{ \min_{\mathbf{y} \in \Omega_i^j} \|\mathbf{x} - \mathbf{y}\|^2 \right\} = \arg \min_{\Omega_i^j \in \Omega_i} \left\{ \Pi_{\Omega_i^j}(\mathbf{x}) \right\} \quad (2)$$

3 Formulation of the mesh curving problem

In this section, we pose the mesh curving problem as a constrained optimization problem. Then, we define the boundary condition using the geometric model, and show the proposed high-order mesh curving technique.

3.1 Constrained minimization of the distortion measure

Given an initial linear mesh, \mathcal{M}_I , we want to characterize a curved high-order one, \mathcal{M}_p , in terms of a diffeomorphism ϕ^* [36, 43]. This diffeomorphism should present optimal point-wise distortion, and has to satisfy a prescribed boundary condition. That is,

$$\begin{aligned} M\phi^* &= 1, & \forall \mathbf{y} \in \mathcal{M}_I, \\ \mathbf{T}\phi^* &= \mathbf{g}_D(\mathbf{T}\phi^*), & \forall \mathbf{y} \in \partial\mathcal{M}_I, \end{aligned} \quad (3)$$

where \mathbf{T} is the trace operator, $\mathbf{g}_D(\mathbf{T}\phi^*)$ is a non-linear Dirichlet boundary condition on $\partial\mathcal{M}_I$ that depends on the values of ϕ^* , and

$$M\phi^*(\mathbf{y}) = \eta(\mathbf{D}\phi^*(\mathbf{y})) = \frac{\|\mathbf{D}\phi^*(\mathbf{y})\|^2}{n\sigma(\mathbf{D}\phi^*(\mathbf{y}))^{2/n}}$$

is a point-wise distortion measure defined in terms of the shape distortion measure [44] for linear simplices, where $\|\cdot\|$ and $\sigma(\cdot)$ are the Frobenius norm and the determinant, respectively.

Nevertheless, the shape distortion measure presents finite values when the determinant is negative, and this can potentially lead to meshes with inverted elements. To solve this issue, we propose to regularize the shape distortion measure as

$$\eta(\mathbf{D}\phi^*) = \frac{|\mathbf{D}\phi^*|^2}{n\sigma_0(\mathbf{D}\phi^*)^{2/n}}, \quad \text{where } \sigma_0 = \frac{1}{2}(\sigma + |\sigma|). \quad (4)$$

In this manner, when the determinant is non-positive, the point-wise distortion takes a value of infinity, and when the determinant is positive, the point-wise distortion takes a finite value.

In order to solve the problem in Equation (3), we first rewrite it as a constrained optimization one in the following manner

$$\begin{aligned} \min_{\phi \in \mathcal{V}} E(\phi) &= \|M\phi\|^2 \\ \text{subject to:} & \\ \mathbf{T}\phi &= \mathbf{g}_D(\mathbf{T}\phi), \end{aligned} \quad (5)$$

where

$$\mathcal{V} = \left\{ \mathbf{u} \in [\mathcal{C}^0(\mathcal{M}_I)]^n \text{ such that } \mathbf{u}|_{e_I} \in [\mathcal{P}^p(e_I)]^n \quad \forall e_I \in \mathcal{M}_I \right\}.$$

Being $\mathcal{P}^p(e_I)$ the space of polynomials of degree at most p over the element e_I . Thus,

$$\phi = \sum_{i=1}^N \mathbf{x}_i N_i,$$

being $\{N_i\}_{i=1,\dots,N}$ a Lagrangian basis of element-wise polynomials continuous at the element interfaces. Note that ϕ depends on the nodal positions and therefore, so does the functional in Equation (5).

We have introduced a merit function to measure the distortion of the mapping that transforms the linear mesh into a curved high-order one, and the boundary condition, \mathbf{g}_D is a constraint of the optimization problem. To solve the constrained optimization problem in (5), we use a penalty approach, see [45], in which we introduce the boundary constraint into the objective function in a weak sense as follows

$$\min_{\phi \in \mathcal{V}} E_\mu(\phi) = \frac{E(\phi)}{\|1\|_{\mathcal{M}_I}^2} + \mu \frac{\|\mathbf{T}\phi - \mathbf{g}_D(\mathbf{T}\phi)\|_{\partial\mathcal{M}_I}^2}{\|1\|_{\partial\mathcal{M}_I}^2}, \quad (6)$$

where μ is a penalty parameter that enforces the validity of the constraint when it tends to infinity. We have introduced the measures of the initial mesh and its boundary in order to balance the two contributions of the new functional.

3.2 Definition of the boundary condition

In order to solve the problem in Equation (6), we need to define the non-linear Dirichlet boundary condition $\mathbf{g}_D(\mathbf{T}\phi)$. The boundary condition takes into account the geometric model Ω , and we define it by means of the point projection algorithm presented in Section 2.2, and the current value of ϕ . Thus, we define the boundary condition as

$$\mathbf{g}_D(\mathbf{T}\phi) = \sum_{i=1}^{N_b} \Pi_{\Omega_{\mathbf{x}_i}}(\mathbf{x}_i) N_i^b = \Pi_\Omega(\mathbf{T}\phi), \quad (7)$$

where $\Omega_{\mathbf{x}_i}$ denotes the entity where the boundary node \mathbf{x}_i belongs to, and $\{N_i\}_{i=1,\dots,N_b}$ is a Lagrangian basis of shape functions continuous between adjacent boundary faces. Thus, we generate a function $\mathbf{g}_D \in \mathcal{V}_b$, where

$$\mathcal{V}_b = \{\mathbf{u} \in \mathcal{C}^0 \text{ such that } \mathbf{u}|_{f_I} \in [\mathcal{P}^p(f_I)]^n \forall f_I \in \partial\mathcal{M}_I\},$$

being $\mathcal{P}^p(f_I)$ the space of polynomials of degree at most p over the boundary face f_I . Note that the function \mathbf{g}_D is generated as a linear combination of a Lagrangian basis of shape functions. The coefficients of the linear combination are the projection of the boundary high-order nodes onto the geometric entities they belong to. Thus, the boundary condition is defined as the interpolation of the geometric model into the function space \mathcal{V}_b . The interpolation points are the projection of the high-order nodes of the curved mesh.

Since we have grouped the curves and surfaces into wires and shells, the interpolation points are free to jump between curves and surfaces of the wire and shell they belong to. The only constraint for the interpolation points is that they need to be associated to the same wire or shell during the whole optimization process.

3.3 High-order mesh curving

To obtain a curved high-order mesh, we optimize the functional in Equation (6) with an increasing penalty parameter. Nevertheless, as shown in Section 3.2, the boundary condition depends on the actual solution of the problem. Thus, we propose to apply a fix-point iteration in which

$$\mathbf{g}_D^k = \Pi_\Omega(\mathbf{T}\boldsymbol{\phi}^k), \quad \boldsymbol{\phi}^{k+1} = \arg \min_{\boldsymbol{\phi} \in \mathcal{V}} E_{\mu^k}(\boldsymbol{\phi}; \mathbf{g}_D^k)$$

In this manner, we are able to deal with the non-linearity of the boundary constraint. Algorithm 1 describes the proposed penalty method for high-order mesh curving. The inputs of the algorithm are an initial linear mesh, \mathcal{M}_I , a geometric model, Ω , and the tolerances for the non-linear problem and the constraint norm, ω^* and ε^* , respectively. The algorithm stops when a solution is found that satisfies

$$\left\| \nabla E_{\mu^k}(\boldsymbol{\phi}^k; \mathbf{g}_D^k) \right\| < \omega^* \quad \text{and} \quad \left\| \mathbf{T}(\boldsymbol{\phi}^k) - \mathbf{g}_D^k \right\|_{\partial\Omega_I}^2 < \varepsilon^*.$$

The output of the algorithm is a valid curved high-order mesh, \mathcal{M}_p , that approximates the target geometric domain. In Line 2 we initialize $\boldsymbol{\phi}^0$ to the identity mapping, \mathbf{Id} . Note that the identity mapping is optimal with respect to the distortion measure. However, it does not satisfy the boundary constraint. In Line 3, we initialize the boundary condition using the projection of the boundary high-order nodes, according to Equation (7). The initial penalty parameter is initialized to 10, and the initial tolerance for the non-linear solver is initialized to the norm of the objective function gradient over 10. Lines 6–15 define the main loop of the proposed penalty method. In Line 7, we optimize the proposed functional. Then, if the norm of the constraint is too large, we increase the penalty parameter and tighten the tolerance of the non-linear solver, Lines 9 and 10. On the contrary, if the norm of the constraint is low enough, we keep the current value of the penalty parameter and we set the tolerance of the non-linear problem to the prescribed tolerance of the mesh curving algorithm. Finally, we update the boundary condition for the next non-linear problem, and iterate the main loop until convergence is achieved.

To optimize each non-linear problem of the proposed penalty method, we use a backtracking line-search method in which the advancing direction is computed using Newton's method and the step-length is set using the Wolfe conditions, see [45] for more details. To solve the linear systems that arise during Newton's method, we use a generalized minimum residual method (GMRES) with a relative tolerance of 10^{-9} , preconditioned with a successive over-relaxation method. The stopping criteria for this optimization process is $\left\| \nabla E_{\mu^k}(\boldsymbol{\phi}^k; \mathbf{g}_D^k) \right\| < \omega^k$.

One of the advantages of the proposed high-order mesh curving algorithm is that it maintains a valid mesh during the whole process. The main reason is that the proposed functional detects invalid meshes by taking infinite values. Thus, when an invalid configuration is detected, the backtracking line-search reduces the step

Algorithm 1 Penalty method for high-order mesh curving.**Input:** Mesh \mathcal{M}_I , GeometricModel Ω , Real ω^* , Real ε^* **Output:** CurvedHighOrderMesh \mathcal{M}_P

```

1: function highOrderMeshCurving
2:    $\phi^0 \leftarrow \mathbf{Id}$ 
3:    $\mathbf{g}_D^0 \leftarrow \Pi_\Omega(\phi^0)$ 
4:    $\mu^0 \leftarrow 10$ 
5:    $\omega^0 \leftarrow \|\nabla E_{\mu^0}(\phi^0; \mathbf{g}_D^0)\| / 10$ 
6:   while  $\|\mathbf{T}\phi^k - \mathbf{g}_D^k\|_{\partial\mathcal{M}_I} > \varepsilon^*$  and  $\|\nabla E_{\mu^k}(\phi^k; \mathbf{g}_D^k)\| > \omega^*$  do
7:      $\phi^{k+1} \leftarrow \text{optimizeFunction}(E_{\mu^k}(\phi^k; \mathbf{g}_D^k), \omega^k)$ 
8:     if  $\|\mathbf{T}(\phi^k) - \mathbf{g}_D^k\|_{\partial\mathcal{M}_I} > \varepsilon^*$  then
9:        $\mu^{k+1} \leftarrow 10\mu^k$ 
10:       $\omega^{k+1} \leftarrow \omega^k / 10$ 
11:     else
12:        $\omega^{k+1} \leftarrow \omega^*$ 
13:     end if
14:      $\mathbf{g}_D^{k+1} \leftarrow \Pi_\Omega(\phi^{k+1})$ 
15:   end while
16:    $\mathcal{M}_P \leftarrow \phi(\mathcal{M}_I)$ 
17: end function

```

length until a valid mesh is obtained for the next iteration. Moreover, we do not need to ensure that the boundary condition defines a valid boundary mesh at each iteration of the penalty method. Since we ensure the volume mesh validity and we re-compute the boundary condition at each step, in practical situations we obtain a valid boundary condition at some point of the mesh curving process.

To increase the practical robustness of the mesh curving method, we apply a p -continuation technique in order. In this manner, the optimal configuration of a given polynomial degree is used as the initial condition on to optimize the mesh for the next polynomial degree. This p -continuation technique has allowed us to obtain curved high-order meshes with stretched elements for complex geometries.

4 Examples

This section presents several examples that show the capabilities of the presented high-order mesh curving method. Specifically, we show four three-dimensional examples in which we present isotropic and stretched meshes for two different geometric models.

To generate the initial linear meshes, we have used Pointwise [40]. The mesh curving framework has been implemented in Python [46] using the FEniCS [47] and the petsc4py [48] libraries. To project the boundary high-order nodes we have used both the geode [49] and the Open CASCADE [50] libraries interfaced with a python wrapper using swig [51].

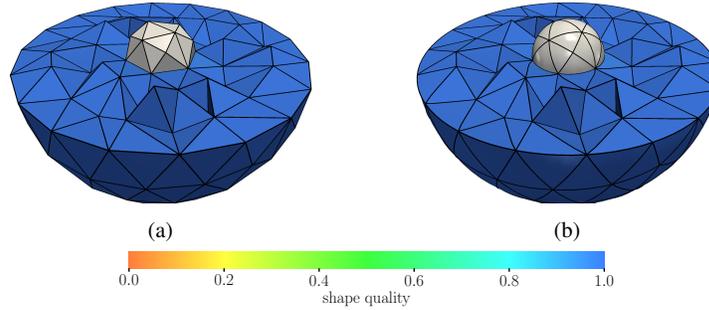


Fig. 2: Isotropic meshes generated around a sphere: (a) initial linear mesh; and (b) curved high-order mesh of polynomial degree five.

The optimization process has been performed in the MareNostrum4 super-computer located at the Barcelona Supercomputing Center. It is composed of 3456 nodes, connected using an Intel Omni-Path network. Each node contains two Intel Xeon Platinum 8160 CPU with 24 cores, each at 2.10 GHz, and 96 GB of RAM memory.

In all the examples, we color the elements of the mesh according to its element quality relative to the initial meshes [23]

$$q_{e_l} = \frac{1}{\eta_{e_l}}, \quad \text{where} \quad \eta_{e_l} = \left(\frac{\int_{e_l} (M\phi)^2 d\Omega}{\int_{e_l} 1 d\Omega} \right)^{1/2}.$$

The element quality takes values between zero and one, being zero for inverted elements, and one for ideally deformed elements.

4.1 Isotropic mesh around a sphere

In this example we show the generation of an isotropic curved high-order mesh generated for the exterior domain of a sphere. The inner and outer spheres radius are one and 21 units, respectively. The initial linear mesh, see Figure 2a, contains isotropic elements of size 1.0. Figure 2b shows a curved high-order mesh of polynomial degree five, obtained after applying the proposed mesh curving technique. The mesh contains 2048 elements and 38167 nodes. The whole optimization process has been performed using 24 cores and took 1261 seconds distributed in seventeen iterations of the penalty method. The norm of the constraint at the last iteration is of the order of 10^{-9} , and the minimum element quality of the whole mesh is 0.972. Figures 6a and 6b show a detailed view of the initial linear mesh and the curved high-order one

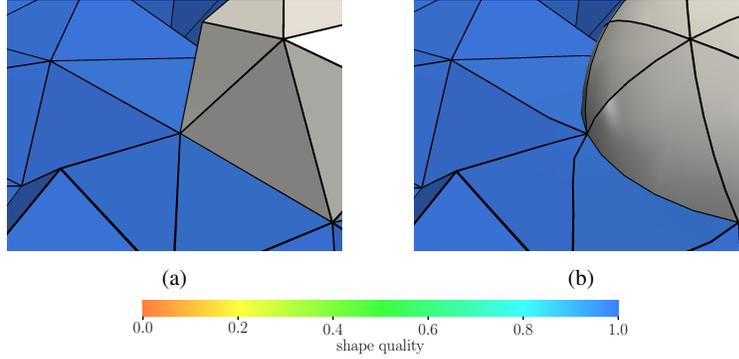


Fig. 3: Detail of the isotropic meshes generated around a sphere: (a) initial linear mesh; and (b) curved high-order mesh of polynomial degree five.

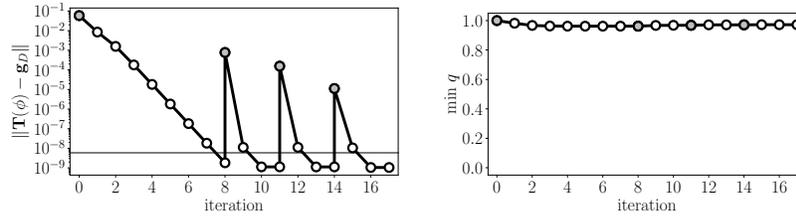


Fig. 4: Evolution of the constraint norm (left) and minimum element quality (right) over the iterations of the penalty method. Thin line denotes the tolerance of the constraint norm, and gray dots denote the first iteration of a new polynomial degree.

near the inner sphere. Note that the curved high-order mesh represents the spherical boundary with a high geometric precision without hampering the element quality.

Figure 4 shows the evolution of the norm of the boundary condition and the minimum element quality of the mesh, against the iterations of the proposed penalty method. In Figure 2, we depict with gray dots the initial iteration of each polynomial degree. The first gray dot denotes the starting iteration of the mesh curving for the mesh of polynomial degree two. Note that the norm of the constraint decreases linearly in logarithmic scale with the iterations of the penalty method. At each iteration, the boundary condition norm is reduced one order of magnitude, and we converge it in eight iterations. Then, we increase the polynomial degree of the mesh to three. Since the boundary condition is also expressed in terms of piece-wise polynomials of degree three, it is able to capture more features of the geometric model. For this reason, the norm of the boundary condition increases. Nevertheless, in three iterations of the penalty method, we are able to obtain a mesh of polynomial degree three. In the next iterations, this behavior is repeated to obtain the meshes of polynomial degree four and five. Note that during the whole process, the mesh remains

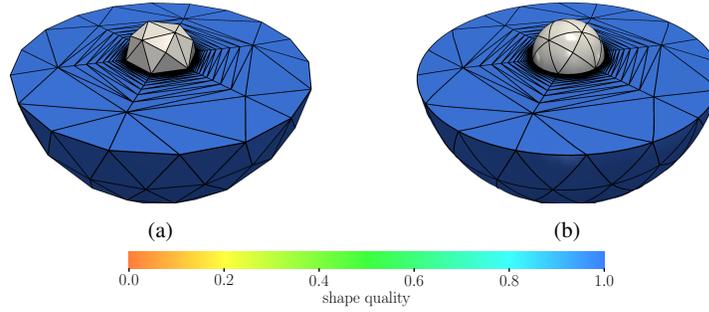


Fig. 5: Stretched meshes generated around a sphere: (a) initial linear mesh; and (b) curved high-order mesh.

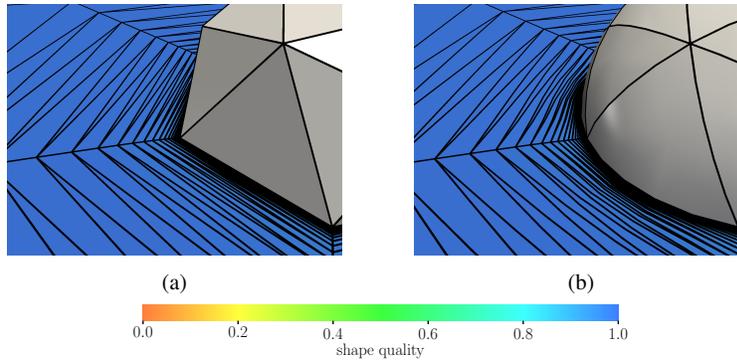


Fig. 6: Detail of the stretched meshes generated around a sphere: (a) initial linear mesh; and (b) curved high-order mesh.

valid, since we are enforcing in the proposed penalty method the validity of the mesh.

4.2 Stretched elements mesh around a sphere

In this example we present the mesh generated for the exterior domain of a sphere with high stretched elements. The geometric model and the element sizes are the same as the ones in Example 4.1. Nevertheless, we introduce a boundary layer around the inner sphere, see Figure 5a. The boundary layer is defined by a wall distance of 10^{-3} , a growing factor of 1.3, and 24 layers of elements. The maximum stretching of this boundary layer is 10^3 . In Figure 5b we show the optimized curved high-order mesh of polynomial degree five, composed of 3616 elements and 71072

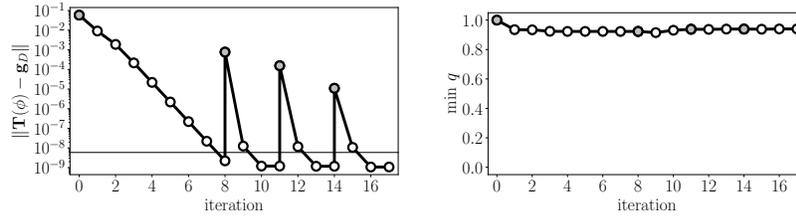


Fig. 7: Evolution of the constraint norm (left) and minimum element quality (right) over the iterations of the penalty method. Thin line denotes the tolerance of the constraint norm, and gray dots denote the first iteration of a new polynomial degree.

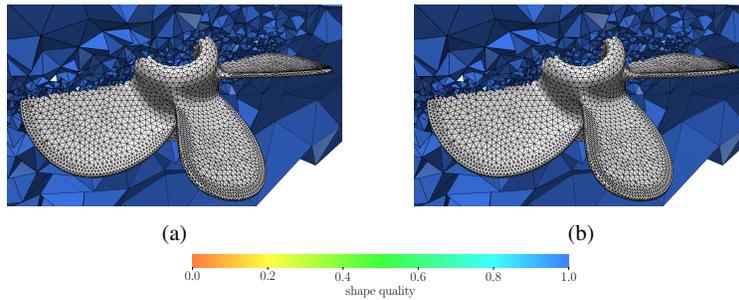


Fig. 8: Isotropic meshes generated around a propeller: (a) initial linear mesh; and (b) curved high-order mesh.

nodes. The optimization process is performed using 24 cores. The whole process is performed in seventeen iterations and takes 2121 seconds. At the last iteration, the norm of the constraint is of the order of 10^{-9} , and the minimum element quality is 0.94.

The evolution of the constraint norm through the iterations of the penalty method, see Figure 7, presents a similar behavior than the evolution of the inviscid mesh. Nevertheless, although the presence of highly stretched elements, the minimum element quality is only slightly reduced compared with the isotropic case.

4.3 Isotropic mesh around a propeller

This example shows the generation of a curved high-order mesh for a complex geometry. In this example, we generate an initial linear mesh around a propeller, see Figure 8a. We apply the proposed technique to obtain the high-order mesh of polynomial degree three shown in Figure 8b. It is composed of 227214 elements and 980992 nodes. The optimization process is performed using 768 cores, and

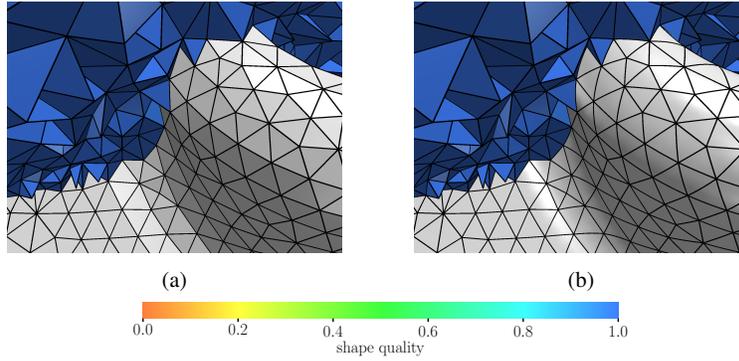


Fig. 9: Detail of the isotropic meshes generated around a propeller: (a) initial linear mesh; and (b) curved high-order mesh.

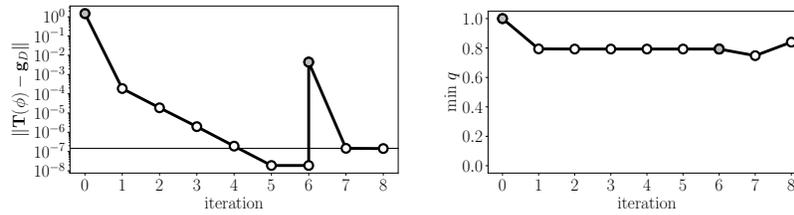


Fig. 10: Evolution of the constraint norm (left) and minimum element quality (right) over the iterations of the penalty method. Thin line denotes the tolerance of the constraint norm, and gray dots denote the first iteration of a new polynomial degree.

converges in 8 iterations of the penalty method, and takes 431 seconds. At the last iteration, the norm of the boundary condition is of the order of 10^{-7} , and the minimum element quality is 0.839.

In Figure 10, we show the evolution of the boundary condition norm and the minimum element quality along the iterations of the penalty method. In the first iterations, the boundary condition norm is reduced linearly in logarithmic scale. In six iterations we obtain the converged quadratic mesh, and in two more iterations we obtain the final curved high-order mesh of polynomial degree three.

4.4 Stretched elements mesh around a propeller

In this example we deal with the generation of a mesh with stretched elements for the exterior domain of a propeller. The boundary layer around the propeller contains 25 layers of elements, a wall distance of 0.002, and a growing ratio of 1.3. This leads

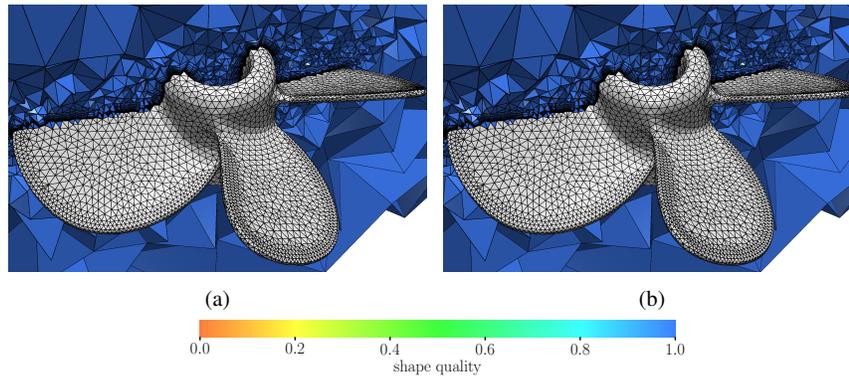


Fig. 11: Stretched meshes generated around a propeller: (a) initial linear mesh; and (b) curved high-order mesh.

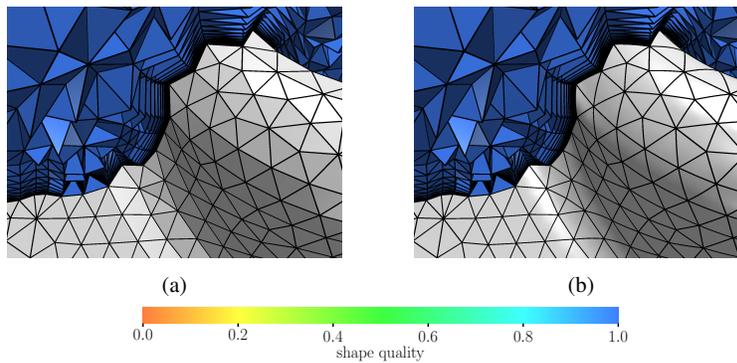


Fig. 12: Detail of the stretched meshes generated around a propeller: (a) initial linear mesh; and (b) curved high-order mesh.

to maximum stretching factor of 750. The objective of this example is to show that we are able to generate a high-order mesh with high stretched elements around a complex geometry. Figure 11a shows the initial linear mesh, and Figure 11b shows the final curved high-order mesh of polynomial degree three. This mesh is obtained in eight iterations of the penalty method, and takes 5452 seconds (1.5 hours) to optimize it using 768 processors. The mesh is composed of 1648596 elements and 6863825 nodes. At the last iteration, the norm of the boundary constraint is of the order of 10^{-7} , and the minimum element quality is 0.524.

The evolution of the norm of the constraint, see Figure 13, shows a similar behavior than in the inviscid case, even when a boundary layer around the propeller is present. In addition, the complexity of the geometric model and the highly stretched elements induce a lower minimum quality with respect to the isotropic case.

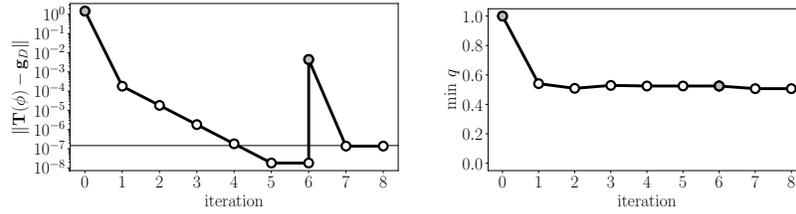


Fig. 13: Evolution of the constraint norm (left) and minimum element quality (right) over the iterations of the penalty method. Thin line denotes the tolerance of the constraint norm, and gray dots denote the first iteration of a new polynomial degree optimization.

5 Conclusions

We have presented a novel high-order mesh curving technique in which the boundary elements of the high-order mesh are able to span and slide between different geometric entities of the target model by using a virtual geometry kernel. To accomplish this, we have deduced a novel methodology to introduce the boundary condition of a high-order mesh curving problem based on the projection of the boundary nodes onto the geometric model.

Currently, the proposed algorithm to solve the constrained minimization problem is a penalty method. Nevertheless, we could solve the constrained minimization problem using the augmented Lagrangian method, as we have done in other works. Using the augmented Lagrangian formulation, the value of penalty parameter is bounded and therefore, the condition number of the system matrix does not increase.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 715546. This work has also received funding from the Generalitat de Catalunya under grant number 2017 SGR 1731. The work of Xevi Roca has been partially supported by the Spanish Ministerio de Economía y Competitividad under the personal grant agreement RYC-2015-01633.

References

1. B. Szabó and I. Babuška. *Finite Element Analysis*. John Wiley & Sons New York, 1991.
2. C Schwab. *p- and hp-finite element methods: Theory and applications in solid and fluid mechanics*. Clarendon Press Oxford, 1998.

3. MO Deville, PF Fischer, and EH Mund. *High-order methods for incompressible fluid flow*, volume 9. Cambridge University Press, 2002.
4. JS Hesthaven and T Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics. Springer, 2007.
5. G Karniadakis and S Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.
6. P. E. Vos, S. Sherwin, and R. Kirby. From h to p efficiently: implementing finite and spectral/hp element methods to achieve optimal performance for low- and high-order discretisations. *J. Comput. Phys.*, 229(13):5161–5181, 2010.
7. C. Cantwell, S. Sherwin, R. Kirby, and P. Kelly. From h to p efficiently: strategy selection for operator evaluation on hexahedral and tetrahedral elements. *Comput. Fluids*, 43(1):23–28, 2011.
8. C. Cantwell, S. Sherwin, R. Kirby, and P. Kelly. From h to p efficiently: selecting the optimal spectral/hp discretisation in three dimensions. *Math. Model. Nat. Phenom.*, 6(3):84–96, 2011.
9. R. Löhner. Error and work estimates for high-order elements. *Int. J. Numer. Meth. Fluids*, 67(12):2184–2188, 2011.
10. M. Yano et al. *An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes*. PhD thesis, Massachusetts Institute of Technology, 2012.
11. R. Kirby, S. Sherwin, and B. Cockburn. To CG or to HDG: a comparative study. *J. Sci. Comput.*, 51(1):183–212, 2012.
12. A. Huerta, X. Roca, A. Angeloski, and J. Peraire. Are high-order and hybridizable discontinuous Galerkin methods competitive? *Oberwolfach Rep.*, 9(1):485 – 487, 2012.
13. R. Löhner. Improved error and work estimates for high-order elements. *Int. J. Numer. Meth. Fluids*, 72:1207–1218, 2013.
14. Z.J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H.T. Huynh, et al. High-order cfd methods: current status and perspective. *Int. J. Numer. Meth. Fluids*, 72(8):811–845, 2013.
15. A. Huerta, A. Angeloski, X. Roca, and J. Peraire. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int. J. Numer. Meth. Eng.*, 96:529–560, 2013.
16. S. Dey, M. Shephard, and J.E. Flaherty. Geometry representation issues associated with p-version finite element computations. *Comput. Meth. Appl. M.*, 150(1–4):39–55, 1997.
17. S. Dey, R. O’Bara, and M. S. Shephard. Curvilinear mesh generation in 3D. *Comput. Aided Design*, 33:199–209, 2001.
18. X. Luo, M. S. Shephard, J.-F. Remacle, R. O’Bara, M. Beall, B. Szabó, and R. Actis. P-version mesh generation issues. In *Proc. 11th Int. Meshing Roundtable*, pages 343–354. Springer Berlin Heidelberg, 2002.
19. X. Luo, M. S. Shephard, R. O’Bara, R. Nastasia, and M. Beall. Automatic p-version mesh generation for curved domains. *Eng. Comput.*, 20(3):273–285, 2004.
20. M. S. Shephard, J. E. Flaherty, K. Jansen, X. Li, X. Luo, N. Chevaugéon, J.-F. Remacle, M. Beall, and R. O’Bara. Adaptive mesh generation for curved domains. *Appl. Numer. Math.*, 52(2-3):251–271, 2005.
21. P.-O. Persson and J. Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. In *Proc. 47th AIAA*, 2009.
22. D. Moxey, M.D. Green, S.J. Sherwin, and J. Peiró. An isoparametric approach to high-order curvilinear boundary-layer meshing. *Computer Methods in Applied Mechanics and Engineering*, 283(0):636 – 650, 2015.
23. A. Gargallo-Peiró, X. Roca, J. Peraire, and J. Sarrate. Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 103(5):342–363, 2015.
24. D. Moxey, D. Ekelschot, Ü. Keskin, S.J. Sherwin, and J. Peiró. High-order curvilinear meshing using a thermo-elastic analogy. *Computer-Aided Design*, 72:130–139, 2016.
25. M. Fortunato and P.E. Persson. High-order unstructured curved mesh generation using the winslow equations. *Journal of Computational Physics*, 307:1–14, 2016.

26. S. Sherwin and J. Peiró. Mesh generation in curvilinear domains using high-order elements. *Int. J. Numer. Meth. Eng.*, 53(1):207–223, 2002.
27. Z. Xie, R. Sevilla, O. Hassan, and K. Morgan. The generation of arbitrary order curved meshes for 3D finite element analysis. *Comput. Mech.*, 51:361–374, 2012.
28. Roman Poya, Ruben Sevilla, and Antonio J Gil. A unified approach for a posteriori high-order curved mesh generation using solid mechanics. *Computational Mechanics*, 58(3):457–490, 2016.
29. Ruben Sevilla, Luke Rees, and Oubay Hassan. The generation of triangular meshes for nurbs-enhanced fem. *International Journal for Numerical Methods in Engineering*, 108(8):941–968, 2016.
30. T. Toulorge, C. Geuzaine, J.-F. Remacle, and Jonathan Lambrechts. Robust untangling of curvilinear meshes. *J. Comput. Phys.*, 254:8 – 26, 2013.
31. Steve L Karman, J T Erwin, Ryan S Glasby, and Douglas Stefanski. High-order mesh curving using wcn mesh optimization. In *46th AIAA Fluid Dynamics Conference*, page 3178, 2016.
32. Mike Stees and Suzanne M Shontz. A high-order log barrier-based mesh generation and warping method. *Procedia Engineering*, 203:180–192, 2017.
33. L Liu, Y Zhang, T J R Hughes, M A Scott, and T W Sederberg. Volumetric t-spline construction using boolean operations. *Engineering with Computers*, pages 1–15, 2013.
34. A. Gargallo-Peiró, X. Roca, J. Peraire, and J. Sarrate. A distortion measure to validate and generate curved high-order meshes on cad surfaces with independence of parameterization. *International Journal for Numerical Methods in Engineering*, 2015.
35. T. Toulorge, J. Lambrechts, and J.F. Remacle. Optimizing the geometrical accuracy of curvilinear meshes. *Journal of Computational Physics*, 2016.
36. E. Ruiz-Gironés, X. Roca, and J. Sarrate. High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation. *Computer-Aided Design*, 72:52–64, 2016.
37. A. Kelly, L. Kaczmarczyk, and C.J. Pearce. Mesh Improvement Methodology for 3D Volumes with Non-Planar Surfaces. In *Proc. 21st Int. Meshing Roundtable*, 2011.
38. E. Ruiz-Gironés, J. Sarrate, and X. Roca. Generation of curved high-order meshes with optimal quality and geometric accuracy. *Procedia Engineering*, 163:315–327, 2016.
39. Eloi Ruiz-Gironés, Abel Gargallo-Peiró, Josep Sarrate, and Xevi Roca. An augmented lagrangian formulation to impose boundary conditions for distortion based mesh moving and curving. *Procedia Engineering*, 203:362–374, 2017.
40. Pointwise Inc. Mesh Generation Software for CFD — Pointwise, Inc. <http://www.pointwise.com>, 2018.
41. Timothy J Tautges. Cgm: A geometry interface for mesh generation, analysis and other applications. *Engineering with Computers*, 17(3):299–314, 2001.
42. Robert Haimes and Mark Drela. On the construction of aircraft conceptual geometry for high-fidelity analysis and design. In *50th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition*, page 683, 2012.
43. 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*, 31(3):423–437, 2015.
44. P. M. Knupp. Algebraic mesh quality metrics. *SIAM J. Numer. Anal.*, 23(1):193–218, 2001.
45. J. Nocedal and S. Wright. *Numerical optimization*. Springer Verlag, 1999.
46. Python Software Foundation. Python. <http://www.python.org>, 2018.
47. Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.
48. petsc4py. PETSc for Python, 2018.
49. Geode. Project Geode: Geometry for Simulation. <http://www.pointwise.com/geode/>, 2018.
50. Open CASCADE. Open CASCADE Technology, 3D modeling and numerical simulation. www.opencascade.org, 2012.
51. swig. Simplified Wrapper and Interface Generator, 2018.