Journal of Computational Physics 443 (2021) 110500

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Measuring and improving the geometric accuracy of piece-wise polynomial boundary meshes

Eloi Ruiz-Gironés [a], Josep Sarrate [b], Xevi Roca [a,*]

[a] Computer Applications in Science and Engineering, Barcelona Supercomputing Center - BSC, E-08034 Barcelona, Spain
[b] Laboratori de Càlcul Numèric (LaCàN), Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, Spain

ARTICLE INFO

Article history:
Available online 10 June 2021

Keywords:
Curved high-order mesh
Geometric accuracy
Disparity measure
Mesh optimization
Order of convergence
Non-interpolative meshes

ABSTRACT

We present a new disparity functional to measure and improve the geometric accuracy of a curved high-order mesh that approximates a target geometry model. We have devised the disparity to account for compound models, be independent of the entity parameterization, and allow trimmed entities. The disparity depends on the physical mesh and the auxiliary parametric meshes. Since it is two times differentiable on all these variables, we can minimize it with a second-order method. Its minimization with the parametric meshes as design variables measures the geometric accuracy of a given mesh. Furthermore, the minimization with both the physical and parametric meshes as design variables improves the geometric accuracy of an initial mesh. We have numerical evidence that the obtained meshes converge to the target geometry (unitary normal) algebraically, in terms of the element size, with order $2p$ ($2p-1$, respectively), where p is the polynomial degree of the mesh. Although we obtain meshes with non-interpolative boundary nodes, we propose a post-process to enforce, if required by the application, meshes with interpolative boundary nodes and featuring the same order of geometric accuracy. In conclusion, we can obtain super-convergent orders, at least for sufficiently smooth parametric curve (surface) entities, for meshes of polynomial degrees up to 4 (3, respectively). In perspective, this super-convergence might enable using a lower polynomial degree to approximate the geometry than to approximate the solution without hampering the required geometric accuracy for high-order analysis.

1. Introduction

Curved meshes represented by piece-wise polynomial parameterizations where the boundary points interpolate the boundary of the approximated CAD geometry are perceived accurate enough to perform finite element analysis with high-order methods. This perception is so since when successively refining the meshes and projecting the boundary nodes onto the geometry boundary, the utilization of interpolative boundary points leads to a $(p+1)$-order approximation to geometry, where p is the used polynomial degree for the mesh representation.

There are many methods [1–18] to generate curved high-order meshes with interpolative boundary nodes on top of the geometric entities of complex CAD models. Nevertheless, when performing a simulation using these meshes, the accuracy and order of convergence of the approximated solution might be insufficient in a twofold manner. First, several works [1,

* Corresponding author.
E-mail addresses: eloi.ruizgirones@bsc.es (E. Ruiz-Gironés), jose.sarrate@upc.edu (J. Sarrate), xevi.roca@bsc.es (X. Roca).

https://doi.org/10.1016/j.jcp.2021.110500

19,20] highlight the significant influence of geometric error in the accuracy of the numerical solution. Since curved meshes cannot reproduce, in general, the target geometries, the discretization of the target domain induces a geometric error. Second, the expected accuracy and order of convergence of approximation to solution are both reduced for reference frame formulations with parametric curved elements. This reduction becomes higher as the polynomial degree of the geometric approximation increases, see [21,22]. Therefore, the question arises whether it is possible to improve the accuracy and order of convergence of standard curved meshes with interpolative boundary nodes. In perspective, if the accuracy improvement is significant, it poses new possibilities. We might want to answer if it is possible to use a lower polynomial degree to approximate the geometry than to approximate the solution while preserving the required geometric accuracy.

The answers to these questions are essential for engineers and scientists that combine unstructured high-order methods [23–27] with curved meshes to perform computational analysis. This combination is well suited to tackle problems where the solution is highly sensitive to discretization errors, and thus, high numerical accuracy and geometrical flexibility are needed. This is the case of the prediction of transition from laminar to turbulent flow, where low numerical dissipation and dispersion, and high geometric accuracy are needed to predict where the flow transitions appear [28,29]. On the one hand, high-order methods converge as $\mathcal{O}(h^{k+1})$, where h is the characteristic mesh size, and k is the polynomial degree of the approximation to solution, and thus, they have been proved to be faster than low-order methods in several applications [30–38], especially in those problems where an implicit solver is required [39]. On the other hand, curved high-order meshes have the potential to approximate complex domains with the required accuracy to enable the advantages of high-order methods. Moreover, the approximation of the geometry with curved meshes features other advantages such as reducing the spurious numerical artifacts that may arise from a piece-wise linear approximation of the curved domain boundaries [1,8,19,20,40–42].

In the literature, there are several ways of computing a *distance* between a mesh and a manifold. For instance, the Hausdorff distance, the Fréchet distance [43], the area-based and Taylor-based distances introduced in [44] and [45], respectively, or the disparity measure introduced in [46]. These distances define a measure of the accuracy of a mesh that does not depend on the parameterization of the geometric entity. However, they may be difficult to compute, are not differentiable, require an interpolative mesh, or only take into account a single geometric entity. Thus, they cannot be used to check the geometric accuracy of a mesh against a geometric model composed of several entities.

In [46,47], we proposed a disparity functional directly in computational coordinates (domain coordinates for the physical and parametric meshes) and without an explicit mapping that pairs source and target points in physical coordinates (codomain coordinates for the physical mesh and target geometry). This expression in computational coordinates is related to a commutative mapping composed of three mappings. The mappings correspond to the physical mesh (from computational to physical coordinates), the parametric mesh (from computational to parametric coordinates), and the geometry parameterization (from parametric to physical coordinates). The functional implicitly pairs a source point with a target point by using the same computational coordinates. Since we directly posed a non-intrinsic functional in computational coordinates, it can only attain a super-convergence order of $2p - 1/2$. This incomplete super-convergence order, related to the $-1/2$ term, has significant consequences for low polynomial degrees, the ones often desired by practitioners of unstructured high-order methods to approximate curved geometry. In particular, it impedes the perspective to approximate with non-standard fourth-order (sixth-order) accuracy a curved target geometry with a piece-wise quadratic (cubic) curved mesh. Furthermore, for linear meshes, the convergence order impedes achieving an approximation to target geometry with standard second-order accuracy. We need to address these issues since the objective of this work is to generate curved high-order meshes that optimally approximate a given geometric model. To do so, we propose two main contributions.

The first main contribution of this work is to propose a novel derivation to obtain a new intrinsic disparity measure expressed in physical coordinates. Furthermore, we also prove that the disparity is independent of the mesh geometry parameterization, and we show numerical evidence that it super-converges with an order of $2p$. Specifically, we first pose a disparity functional in terms of a new mapping that explicitly pairs source points with target points in physical coordinates. This setting corresponds to a commutative diagram with four mappings, containing the additional pairing mapping in physical coordinates not considered in our previous papers. Then, from that intrinsic functional, we derive the expression of the new disparity in computational coordinates. This new derivation allows us to prove that the disparity measure is independent of the geometry parameterization.

The second main contribution is to pose an optimization problem to compute and improve the disparity value. The selection of the design variables determines the goal of the optimization. To compute the disparity value, we select the parametric nodes as design variables. To improve the disparity value of a given mesh, we select the physical and parametric nodes as design variables. The result of the latter optimization is a non-interpolative mesh. We use the same second-order optimization implementation of our previous papers to obtain the results, but now using the value, first derivative, and second derivative for the new disparity functional. The obtained results provide numerical evidence that the new disparity super-converges with an order of $2p$ for the checked degrees. It also permits the possibility of increasing the geometric accuracy of linear meshes while achieving second-order accuracy. Finally, it enables the compelling perspective to use low polynomial degrees to approximate curved geometries since it features a non-standard super-convergent accuracy of order $2p$.

To obtain interpolative meshes, we also propose a post-process of the optimal non-interpolative mesh that relocates the nodes onto the geometric model. This post-process only involves evaluating the position of the mesh nodes through a mapping. Therefore, it is less computationally expensive than performing the actual orthogonal projection for each node of

the mesh. Although the post-processed mesh is less accurate than the optimal one, we show numerical evidence that it also super-converges to the target geometry with a convergence order of $2p$. Thus, in this work, we show that it is possible to approximate a geometric model with high accuracy using either interpolative or non-interpolative meshes.

It is important to point out that the contributions of this work are to check and to improve the geometric accuracy of a curved high-order mesh. Thus, mesh quality is out of the scope of this article. Nevertheless, in [47] we have used a disparity measure in a mesh quality framework in order to generate curved high-order meshes that are optimal both in the sense of element quality and geometric accuracy.

The layout of this paper is the following. In Section 2, we review the related work. In Section 3, we introduce several definitions. In Section 4, we show the problem statement and the methodology. In Section 5, we define the disparity measure. In Section 6, we formulate the optimization problem to improve the geometric accuracy of a curved high-order mesh. In Section 7, we present several examples to illustrate the features of the proposed disparity measure. Finally, in Section 9, we present the conclusions and the future work.

2. Related work

Let Σ^1 and Σ^2 be two m-dimensional manifolds in $\mathbb{R}^n$, parameterized by $\boldsymbol{\varphi}^1$ and $\boldsymbol{\varphi}^2$, respectively, in such a way that

$$\boldsymbol{\varphi}^i : \begin{array}{ccc} \mathcal{U}_i \subset \mathbb{R}^m & \longrightarrow & \Sigma^i \subset \mathbb{R}^n \\ \mathbf{u} & \longmapsto & \mathbf{x} = \boldsymbol{\varphi}^i(\mathbf{u}) \end{array}, \qquad i = 1, 2,$$

where $\mathcal{U}^1$ and $\mathcal{U}^2$ are the parametric spaces of Σ^1 and Σ^2, respectively. In the literature, there are several approaches to compute a distance measure between them. The Hausdorff distance between Σ^1 and Σ^2 is defined as

$$d_H(\Sigma^1, \Sigma^2) = \max\left\{ \sup_{\mathbf{x}\in\Sigma^1} \inf_{\mathbf{y}\in\Sigma^2} d(\mathbf{x}, \mathbf{y}), \sup_{\mathbf{y}\in\Sigma^2} \inf_{\mathbf{x}\in\Sigma^1} d(\mathbf{x}, \mathbf{y}) \right\}, \tag{1}$$

where $d(\mathbf{x}, \mathbf{y})$ is a distance between points. In [48] the authors propose a method to compute the Hausdorff distance between two triangular meshes ($m = 2$, $n = 3$). The main idea is to sample one of the meshes and then compute the closest point of each sample to the other mesh.

Note that the Hausdorff distance is commonly used to compute the distance between two sets. Thus, it does not take into account that Σ^1 and Σ^2 are parameterized manifolds. For this reason, when dealing with manifolds, it is more common to use the Fréchet distance defined as

$$d_F(\Sigma^1, \Sigma^2) = \inf_{\boldsymbol{\pi}} \sup_{\mathbf{u}\in\mathcal{U}^1} d\Big(\boldsymbol{\varphi}^1(\mathbf{u}), \boldsymbol{\varphi}^2(\boldsymbol{\pi}(\mathbf{u}))\Big), \tag{2}$$

where $\boldsymbol{\pi}$ is the set of all the possible orientation-preserving diffeomorphisms between $\mathcal{U}^1$ and $\mathcal{U}^2$. It can be proven that d_F defines a distance between two manifolds that does not depend on the selected parameterizations for them [43]. However, the calculation of d_F is not straightforward, and several methods to approximate it have been proposed. For instance, in [43,49] the authors compute the Fréchet distance between two polygonal curves ($m = 1$, $n = 2$), and in [50] they compute an approximation of the Fréchet distance between two triangulated surfaces ($m = 2$, $n = 3$).

In [44], the authors define an area-based distance between two polygonal curves in the plane as the area between the curves. To approximate the area distance between two arbitrary curves ($m = 1$, $n = 2$), they approximate the curves using poly-lines and compute the area distance between the poly-lines. It is important to highlight that the authors use the area-based distance in a minimization process to obtain a high-order mesh that approximates with high accuracy the boundaries of the geometry and, at the same time, obtain a valid mesh for simulation purposes. In the minimization process, the derivatives of the area-based distance in terms of the nodal coordinates are computed using finite differences, which may reduce the quadratic convergence order of Newton's method.

Later, reference [45] introduces a Taylor-based distance for planar meshes and curves ($m = 1$, $n = 2$). Specifically, the Taylor distance consists in checking the difference between the tangent and curvature vectors at the interpolation points of the curve. For the three-dimensional case ($m = 2$, $n = 3$), the Taylor distance is the difference between the normal vectors of the mesh and the surface at the interpolation points. Note that the Taylor-based distance is continuously differentiable, both for the two- and three-dimensional case, and it is specific for interpolative meshes.

3. Preliminaries

Before introducing the problem statement and detailing the methodology, we need to detail how we represent the geometric model and the curved high-order mesh.

3.1. Geometric model

A compound model, Ω, is represented as a collection of geometric entities (i.e., vertices, edges, faces, ...) in such a way that

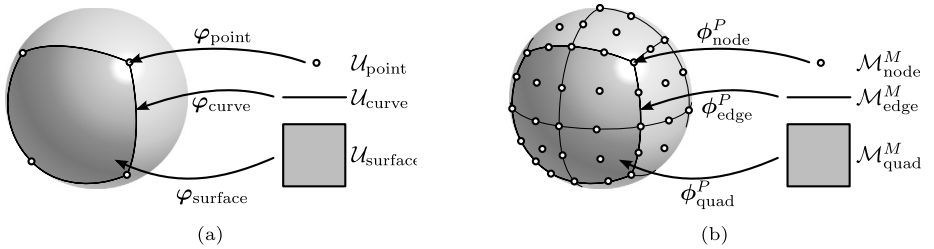


Fig. 1. (a) A geometric model of a sphere composed of six surfaces, twelve curves and eight points; and (b) a curved high-order mesh of polynomial degree two that approximates the geometric model of a sphere.

$$\Omega = \bigcup_{k=1}^{N} \Omega_k.$$

Each geometric entity, Ω_k, is located on a manifold, Σ_k, parameterized as

$$\begin{aligned} \boldsymbol{\varphi}_k : \mathcal{U}_k \subset \mathbb{R}^{d_k} &\longrightarrow \Sigma_k \subset \mathbb{R}^n \\ \mathbf{u} = (u_1, \dots, u_{d_k}) &\longmapsto \mathbf{x} = \boldsymbol{\varphi}_k(\mathbf{u}) \end{aligned},$$

where n and d_k are the dimension of the embedding space and the manifold, respectively, $\boldsymbol{\varphi}_k$ is the parameterization of Σ_k, and $\mathcal{U}_k$ its parametric domain. Fig. 1a shows a geometric model of a sphere composed of six surfaces, twelve curves and eight points. Each entity has its own parametric domain and its own parameterization.

Our implementation needs access to a small subset of the query functions available in a standard CAD system. Specifically, the methods need access to the geometric entities that compose the model and their corresponding parameterizations. Furthermore, the minimization procedure, that we propose later, uses Newton's method and thus, we also need to evaluate the first and second derivatives of the corresponding parameterizations.

3.2. Curved high-order mesh

A curved high-order mesh of polynomial degree p that approximates a geometric model, Ω, is represented as

$$\mathcal{M}^P = \bigcup_{k=1}^{N} \mathcal{M}^P_k,$$

where the curved high-order mesh $\mathcal{M}^P_k$ approximates the geometric entity Ω_k.

We define the mesh in terms of the isoparametric mapping, $\boldsymbol{\phi}$, in such a way that

$$\begin{aligned} \boldsymbol{\phi} : \mathcal{M}^M &\longrightarrow \mathcal{M}^P \subset \mathbb{R}^n \\ \boldsymbol{\xi} = (\xi_1, \dots, \xi_n) &\longmapsto \mathbf{x} = \boldsymbol{\phi}(\boldsymbol{\xi}) = \sum_{v \in \mathcal{M}^P} \mathbf{x}_v N_v(\boldsymbol{\xi}) \end{aligned},$$

where $\mathcal{M}^M$ is the set of reference elements of mesh $\mathcal{M}^P$, $\{N_v\}_{v \in e}$ is a basis of shape functions of continuous element-wise polynomials of degree p, v are the nodes of the mesh, and $\mathbf{x}_v$ their coordinates. Each sub-mesh $\mathcal{M}^P_k$ is defined by restricting the isoparametric mapping to its corresponding reference mesh, $\mathcal{M}^M_k$, see Fig. 1b. Since the isoparametric mapping is continuous, there are no gaps between the different sub-meshes that compose the curved high-order mesh.

4. Problem statement and methodology

4.1. Problem statement, input and output

We consider the problem of evaluating and optimizing the geometric accuracy of the approximation to a target geometry with a piece-wise polynomial mesh. Our solution optimizes an intrinsic disparity functional expressed in terms of physical and parametric meshes. The rest of this section describes the inputs and the outputs of the proposed method.

The input of our algorithm is a geometric model, Ω, defined in terms of a CAD model. In our code, we use the OpenCASCADE [51] library to read a CAD file and interact with the associated geometric model. Moreover, we also need a curved high-order mesh, $\mathcal{M}^P$, of polynomial degree p that approximates the geometric model [52]. Although it is usual that the physical mesh interpolates the geometric model, this is not necessary for our algorithm. To compute the geometric accuracy, we also need a curved high-order mesh of polynomial degree q in the parametric domain of each geometric entity, $\mathcal{M}^U_k$, for $k = 1 \dots, N$. To simplify the computations, we assume that each parametric mesh has the same element connectivity as

the associated restriction of the physical mesh. Nevertheless, we allow different polynomial degrees for the representation of the physical and parametric meshes.

From the input data, we obtain two outputs. The first one is a measure of the geometric accuracy of the physical mesh. The second one is the curved high-order mesh that best approximates the geometric domain. Note that in our approach, we do not take into account topological changes of the physical mesh. That is, we only allow moving the mesh nodes.

4.2. Methodology

The proposed approach is composed of the following steps.

1. *Compute the geometric accuracy of a curved high-order mesh*. We propose a non-linear optimization problem in terms of the parametric meshes and the geometric model. The solution to this problem is a measure of the geometric accuracy of the physical mesh. The unknowns of the optimization problem are the node coordinates of the parametric meshes. To seek the solution, we use Newton's method equipped with a backtracking line-search.
2. *Optimize the geometric accuracy of a curved high-order mesh*. To improve the geometric accuracy of the physical mesh, we propose a non-linear minimization problem. The unknowns of this problem are the node coordinates of the physical mesh and the node coordinates of the parametric meshes. To optimize this problem, we also use Newton's method combined with a backtracking line-search.
3. *Provide different measures of the geometric accuracy*. To check different aspects of the geometric approximation, we provide different measures of the geometric accuracy of the physical meshes. Specifically, we show the accuracy measure proposed in this work, the maximum of the point-wise distances, and a measure of the unit normals error, the latter also proposed in this work. The first two provide information about the *distance* between the curved high-order mesh and the geometric model. The last one provides information about the error of the derivatives of the approximation.

5. Definition of the disparity measure

5.1. The disparity measure and geometric error for one geometric entity

Let Ω^1 and Ω^2 be two geometric entities defined over the manifolds Σ^1 and Σ^2, respectively. We define the disparity measure of Ω^1 with respect to Ω^2 as

$$d\left(\Omega^1, \Omega^2\right) := \inf_{\boldsymbol{\pi} \in \Pi} \left(\frac{\int_{\Omega^1} \|\mathbf{x} - \boldsymbol{\pi}(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega^2\right|^2} \right)^{1/2}, \tag{3}$$

where $\left|\Omega^2\right| = \int_{\Omega^2} 1 \, \mathrm{d}\mathbf{x}$ is the measure of Ω^2, and

$$\Pi = \left\{ \boldsymbol{\pi} \in \mathcal{H}^1(\Omega^1, \Omega^2) \text{ such that } \boldsymbol{\pi} \text{ is a diffeomorphism between } \Omega^1 \text{ and } \Omega^2 \right\}.$$

We define the *geometric error* between a source approximation Ω^1 and a target domain Ω^2 as

$$\left(\frac{\int_{\Omega^1} \|\mathbf{x} - \boldsymbol{\Pi}(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega^2\right|^2} \right)^{1/2},$$

where $\boldsymbol{\Pi}$ is the projection that maps a source point to the closest point in the target geometry. It is a measure of the geometric error since it averages the point-to-point distance between the source and target domains.

The disparity measure is an upper bound of the geometric error. To show this, given a source point $\mathbf{x}$ we know that

$$\|\mathbf{x} - \boldsymbol{\Pi}(\mathbf{x})\| \leq \|\mathbf{x} - \boldsymbol{\pi}(\mathbf{x})\|, \qquad \forall \boldsymbol{\pi} \in \Pi,$$

since $\boldsymbol{\Pi}(\mathbf{x})$ is the closest target point. This inequality holds for all the source points and diffeomorphisms in Π. Hence, we have the inequality

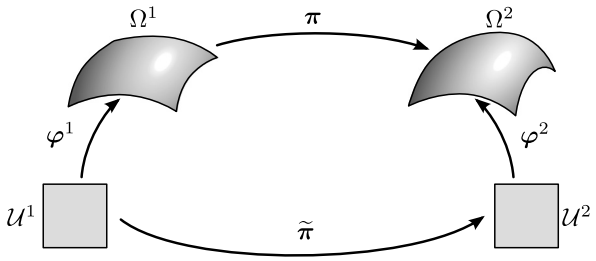


Fig. 2. Diagram of mappings to compute the disparity measure of a single geometric entity.

$$\left(\frac{\int_{\Omega^1} \|\mathbf{x} - \boldsymbol{\Pi}(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega^2\right|^2} \right)^{1/2} \leq \inf_{\boldsymbol{\pi} \in \Pi} \left(\frac{\int_{\Omega^1} \|\mathbf{x} - \boldsymbol{\pi}(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega^2\right|^2} \right)^{1/2} = d\left(\Omega^1, \Omega^2\right).$$

Consequently, the measure of the geometric error, left term, is smaller than the disparity measure, right term.

If the disparity converges to zero, it means that we have a sequence of source domains and pairing mappings $\boldsymbol{\pi}$ where the average length of the lines connecting source and target points converges to zero. Furthermore, taking into account the previous upper bound, if the disparity converges to zero with a specific order, then the geometric error converges to zero at least with that order.

5.2. Computation of the disparity measure and properties

To compute the disparity, we rewrite the integral of Equation (3) in terms of the corresponding parameterizations of Ω^1 and Ω^2, $\boldsymbol{\varphi}^1$ and $\boldsymbol{\varphi}^2$, respectively,

$$d\left(\Omega^1, \Omega^2\right) = \inf_{\boldsymbol{\pi} \in \Pi} \left(\frac{\int_{\mathcal{U}^1} \left\| \boldsymbol{\varphi}^1(\mathbf{u}) - \boldsymbol{\varphi}^2 \circ \widetilde{\boldsymbol{\pi}}(\mathbf{u}) \right\|^2 |J| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2} \right)^{1/2}, \tag{4}$$

where

$$|J| = \sqrt{\det\left(\left(\mathrm{D}\boldsymbol{\varphi}^1\right)^T \mathrm{D}\boldsymbol{\varphi}^1 \right)}$$

and $\widetilde{\boldsymbol{\pi}} = \left(\boldsymbol{\varphi}^2\right)^{-1} \circ \boldsymbol{\pi} \circ \boldsymbol{\varphi}^1$ is the expression of $\boldsymbol{\pi}$ in the local coordinates induced by the parameterizations of the geometric entities. Since $\boldsymbol{\pi}$ is a diffeomorphism, then $\widetilde{\boldsymbol{\pi}}$ defines a diffeomorphism between the parametric spaces $\mathcal{U}^1$ and $\mathcal{U}^2$. Fig. 2 shows the maps involved to compute the disparity measure of a single geometric entity.

The disparity measure is invariant by rotations, translation, and symmetries of the geometric entities. Moreover, if the geometric entities are scaled by a factor, the disparity measure is also scaled by the same factor. It is only necessary to take into account that the Euclidean norm of vectors satisfies these properties.

Note that Equation (3) seems to be an intrinsic definition of the disparity measure since it is not posed in terms of the entity parameterizations. However, to compute the value of the disparity, it is necessary to take parameterizations for both domains and compute the integral using Equation (4). Accordingly, we need to check that the disparity measure is well defined, in the sense that its value does not depend on the selected parameterizations of the geometric entities.

Proposition 1. *Let Ω^1 and Ω^2 be two geometric entities. We consider two different parameterizations for Ω^1, $\boldsymbol{\varphi}^1$ and $\boldsymbol{\alpha}^1$, and for Ω^2, $\boldsymbol{\varphi}^2$ and $\boldsymbol{\alpha}^2$. Let*

$$d_{\boldsymbol{\varphi}} = \inf_{\boldsymbol{\pi} \in \Pi} \left(\frac{\int_{\mathcal{U}_{\boldsymbol{\varphi}}} \left\| \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 \circ \widetilde{\boldsymbol{\pi}} \right\|^2 |J_{\boldsymbol{\varphi}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2} \right)^{1/2},$$

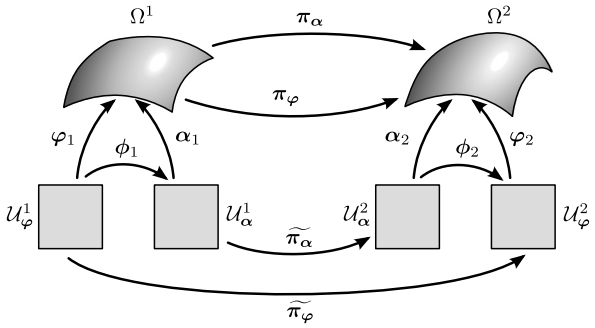


Fig. 3. Two geometric entities with two different parameterizations, and the two optimal diffeomorphisms with the corresponding expression in local coordinates.

$$d_{\boldsymbol{\alpha}} = \inf_{\boldsymbol{\pi}\in\Pi} \left(\frac{\int_{\mathcal{U}_{\boldsymbol{\alpha}}} \left\| \boldsymbol{\alpha}^1 - \boldsymbol{\alpha}\circ\widetilde{\boldsymbol{\pi}} \right\|^2 |J_{\boldsymbol{\alpha}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2} \right)^{1/2}.$$

Then, $d_{\boldsymbol{\varphi}} = d_{\boldsymbol{\alpha}}$.

Proof. To prove Proposition 1, we can show that $d_{\boldsymbol{\alpha}} \leq d_{\boldsymbol{\varphi}}$ and $d_{\boldsymbol{\alpha}} \geq d_{\boldsymbol{\varphi}}$. By symmetry, we only need to detail the proof that $d_{\boldsymbol{\alpha}} \geq d_{\boldsymbol{\varphi}}$. To this end, let us use the diagram of mappings shown in Fig. 3. Let $\boldsymbol{\pi}_{\boldsymbol{\alpha}}$ and $\boldsymbol{\pi}_{\boldsymbol{\varphi}}$ be the diffeomorphisms that achieve the infimum value in $d_{\boldsymbol{\alpha}}$ and $d_{\boldsymbol{\varphi}}$, respectively. Moreover, let $\boldsymbol{\phi}^1 = \left(\boldsymbol{\alpha}^1\right)^{-1} \circ \boldsymbol{\varphi}^1$ and $\boldsymbol{\phi}^2 = (\boldsymbol{\varphi}^2)^{-1} \circ \boldsymbol{\alpha}^2$ be the corresponding change of parameterization of the geometric entities Ω^1 and Ω^2, respectively. Then,

$$(d_{\boldsymbol{\alpha}})^2 = \frac{\int_{\mathcal{U}_{\boldsymbol{\alpha}}} \left\| \boldsymbol{\alpha}^1 - \boldsymbol{\alpha}^2 \circ \widetilde{\boldsymbol{\pi}_{\boldsymbol{\alpha}}} \right\|^2 |J_{\boldsymbol{\alpha}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2} = \frac{\int_{\mathcal{U}_{\boldsymbol{\varphi}}} \left\| \boldsymbol{\varphi}^1 - \boldsymbol{\alpha}^2 \circ \widetilde{\boldsymbol{\pi}_{\boldsymbol{\alpha}}} \circ \boldsymbol{\phi}^1 \right\|^2 |J_{\boldsymbol{\alpha}}||J_{\boldsymbol{\phi}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2},$$

where we have applied the change of coordinates induced by $\boldsymbol{\phi}^1$. That is, instead of integrating in the parametric domain $\mathcal{U}_{\boldsymbol{\alpha}}$, we integrate in the parametric domain $\mathcal{U}_{\boldsymbol{\varphi}}$. By using that $|J_{\boldsymbol{\alpha}}||J_{\boldsymbol{\phi}}| = |J_{\boldsymbol{\varphi}}|$, directly derived from the properties of the determinant and the chain rule, and $\boldsymbol{\alpha}^2 = \boldsymbol{\varphi}^2 \circ \boldsymbol{\phi}^2$, we have that

$$(d_{\boldsymbol{\alpha}})^2 = \frac{\int_{\mathcal{U}_{\boldsymbol{\varphi}}} \left\| \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 \circ \boldsymbol{\phi}^2 \circ \widetilde{\boldsymbol{\pi}_{\boldsymbol{\alpha}}} \circ \boldsymbol{\phi}^1 \right\|^2 |J_{\boldsymbol{\varphi}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2}.$$

Since $\boldsymbol{\phi}^2 \circ \widetilde{\boldsymbol{\pi}_{\boldsymbol{\alpha}}} \circ \boldsymbol{\phi}^1$ is a diffeomorphism that belongs to Π, we can conclude that

$$(d_{\boldsymbol{\alpha}})^2 = \frac{\int_{\mathcal{U}_{\boldsymbol{\varphi}}} \left\| \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 \circ \boldsymbol{\phi}^2 \circ \widetilde{\boldsymbol{\pi}_{\boldsymbol{\alpha}}} \circ \boldsymbol{\phi}^1 \right\|^2 |J_{\boldsymbol{\varphi}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2} \geq \frac{\int_{\mathcal{U}_{\boldsymbol{\varphi}}} \left\| \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 \circ \widetilde{\boldsymbol{\pi}_{\boldsymbol{\varphi}}} \right\|^2 |J_{\boldsymbol{\varphi}}| \mathrm{d}\mathbf{u}}{\left|\Omega^2\right|^2} = (d_{\boldsymbol{\varphi}})^2. \quad \square$$

As a direct consequence of Proposition 1, if a mapping $\boldsymbol{\pi}$ is a minimizer of Functional (3) for a given parameterization of the geometric entity, then $\boldsymbol{\pi}$ is a minimizer for all the possible parameterizations of the geometric models. This consequence is so since $d_{\boldsymbol{\alpha}} = d_{\boldsymbol{\varphi}}$ and thus, $\boldsymbol{\pi}_{\boldsymbol{\varphi}}$ is the mapping that achieves the disparity value for parameterization $\boldsymbol{\varphi}$, but also for parameterization $\boldsymbol{\alpha}$.

5.3. The disparity measure for geometric models

Let Ω^1 and Ω^2 be two geometric models composed of several geometric entities (i.e. vertices, edges and faces) in such a way that

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

We define the disparity measure of Ω^1 with respect to Ω^2 using the disparity measures of each geometric entity. Specifically,

$$d\left(\Omega^1, \Omega^2\right) := \inf_{\boldsymbol{\pi}_k \in \Pi_k} \left(\sum_{k=1}^{N} \frac{\int_{\Omega_k^1} \|\mathbf{x} - \boldsymbol{\pi}_k(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega_k^2\right|^2} \right)^{1/2}, \tag{5}$$

where

$$\Pi_k = \left\{ \boldsymbol{\pi}_k \in \mathcal{H}^1(\Omega_k^1, \Omega_k^2) \text{ such that } \boldsymbol{\pi} \text{ is a diffeomorphism between } \Omega_k^1 \text{ and } \Omega_k^2 \right\}.$$

The disparity measure for geometric models has the same properties as the disparity measure for single geometric entities. Specifically, it is invariant by rotations, translations, and symmetries. If the geometric models are scaled by a factor, the disparity measure is scaled by the same factor. Finally, using Proposition 1 and extending it to geometric models composed of several geometric entities, it is straightforward to prove that the disparity measure for geometric models is well defined in the sense that it does not depend on the chosen parameterizations of the geometric entities. Thus, we can express the disparity measure using the parameterizations of the geometric entities as

$$d\left(\Omega^1, \Omega^2\right) = \inf_{\boldsymbol{\pi}_k \in \Pi_k} \left(\sum_{k=1}^{N} \frac{\int_{\mathcal{U}_k^1} \left\| \boldsymbol{\varphi}_k^1(\mathbf{u}) - \boldsymbol{\varphi}_k^2 \circ \widetilde{\boldsymbol{\pi}}_k(\mathbf{u}) \right\|^2 |J_k| \mathrm{d}\mathbf{u}}{\left|\Omega_k^2\right|^2} \right)^{1/2}. \tag{6}$$

6. Formulation of the optimization process

6.1. Continuous optimization framework for geometric models

Given a geometric model, Ω, we want to characterize a new geometric model, $\Omega^{\boldsymbol{\phi}^*}$, where

$$\Omega = \bigcup_{k=1}^{N} \Omega_k, \qquad \Omega^{\boldsymbol{\phi}^*} = \bigcup_{k=1}^{N} \Omega_k^{\boldsymbol{\phi}_k^*},$$

and $\Omega_k^{\boldsymbol{\phi}^*}$ is defined in terms of a parameterization $\boldsymbol{\phi}_k^* \in \mathcal{H}^1(\mathcal{U}_k, \Omega_k^{\boldsymbol{\phi}_k^*})$, where $\mathcal{U}_k$ is the corresponding parametric space. The new geometric model has to optimally approximate Ω in terms of the proposed disparity measure, see (5). Thus, the optimal mappings $\boldsymbol{\phi}_k^*$, are the ones that satisfy

$$\begin{aligned} \left\{\boldsymbol{\phi}_1^*, \ldots, \boldsymbol{\phi}_N^*, \boldsymbol{\pi}_1^*, \ldots, \boldsymbol{\pi}_N^*\right\} &= \underset{\substack{\boldsymbol{\phi}_k \in \mathcal{H}^1(\mathcal{U}_k, \Omega_k^{\boldsymbol{\phi}_k}) \\ \boldsymbol{\pi}_k \in \Pi_k}}{\operatorname{argmin}} \; E(\phi_1, \ldots, \phi_N, \pi_1, \ldots, \pi_N) \\ &= \underset{\substack{\boldsymbol{\phi}_k \in \mathcal{H}^1(\mathcal{U}_k, \Omega_k^{\boldsymbol{\phi}_k}) \\ \boldsymbol{\pi}_k \in \Pi_k}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{\int_{\Omega_k^{\boldsymbol{\phi}_k}} \|\mathbf{x} - \boldsymbol{\pi}_k(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega_k^2\right|^2}. \end{aligned} \tag{7}$$

Note that the functional in Equation (7) has two sets of unknowns, $\boldsymbol{\phi}_k$ and $\boldsymbol{\pi}_k$, for $k = 1, \ldots, N$. By fixing $\boldsymbol{\phi}_k$ and minimizing the functional with respect of $\boldsymbol{\pi}_k$, we compute the disparity measure of $\Omega^{\boldsymbol{\phi}}$ and Ω. Instead, by minimizing the functional (7) with respect of $\boldsymbol{\phi}_k$ and $\boldsymbol{\pi}_k$ at the same time, we obtain the optimal geometric model, $\Omega^{\boldsymbol{\phi}^*}$, that approximates Ω in the sense of the proposed disparity measure.

Functional (7) does not have, in general, a unique solution. The main reason is that we obtain a parameterization, $\boldsymbol{\phi}_k^*$, for each $\Omega_k^{\boldsymbol{\phi}_k^*}$. Thus, different parameterizations of the same geometric model $\Omega^{\boldsymbol{\phi}^*}$ are also valid minimizers of Functional (7).

6.2. Continuous optimization framework for high-order meshes

The main objective of this work is to generate curved high-order meshes that optimally approximate a given geometric model. Thus, we use the functional in Equation (7) applied to a curved high-order mesh, $\mathcal{M}^P$, and a geometric model, Ω. The parametric domain of the physical mesh is a valid straight-sided mesh, $\mathcal{M}^R$, that has the same connectivity and number of elements as $\mathcal{M}^P$. The reference mesh, $\mathcal{M}^R$, approximates the geometric domain Ω in the sense that it contains sub-meshes $\mathcal{M}^R_k$ that approximate the geometric entities Ω^k. Moreover, we have a mapping $\boldsymbol{\phi}$ that parameterizes the physical mesh from the reference mesh. The optimal mesh according to the disparity measure is defined as

$$\begin{aligned}\left\{\boldsymbol{\phi}^*, \boldsymbol{\pi}_1^*, \dots, \boldsymbol{\pi}_N^*\right\} &= \underset{\substack{\boldsymbol{\phi}\in\mathcal{V}^p \\ \boldsymbol{\pi}_k\in\Pi_k}}{\operatorname{argmin}} E(\boldsymbol{\phi}, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_N) \\ &= \underset{\substack{\boldsymbol{\phi}\in\mathcal{V}^p \\ \boldsymbol{\pi}_k\in\Pi_k}}{\operatorname{argmin}} \sum_{i=1}^{N} \frac{\int_{\mathcal{M}_k^P} \|\mathbf{x} - \boldsymbol{\pi}_k(\mathbf{x})\|^2 \, \mathrm{d}\mathbf{x}}{\left|\Omega_k^2\right|^2}\end{aligned} \tag{8}$$

where

$$\mathcal{V}^P = \left\{\boldsymbol{\phi} \in \left[\mathcal{C}^0(\mathcal{M}^R)\right]^n \text{ such that } \boldsymbol{\phi}_{|e} \in \left[\mathcal{P}^p(e)\right]^n, \forall e \in \mathcal{M}^R\right\},$$

where $\mathcal{P}^p(e)$ is the space of polynomial functions in element e of degree at most p. This corresponds to the standard function space for continuous Galerkin methods, determined by element-wise polynomials of degree at most p and continuous at the element interfaces. Note that the functional in Equation (8) has two unknowns: the parametric mapping that defines the high-order mesh, and the set of diffeomorphisms $\boldsymbol{\pi}_k$, for $k = 1, \dots, N$.

Then, we write Functional (8) in terms of the parametric mapping, $\boldsymbol{\phi}$, and the parameterizations of the geometric entities, $\boldsymbol{\varphi}_k$:

$$E(\boldsymbol{\phi}, \boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_N) = \sum_{k=1}^{N} \frac{\int_{\mathcal{M}_k^R} \left\|\boldsymbol{\phi}_k - \boldsymbol{\varphi}_k \circ \widetilde{\boldsymbol{\pi}}_k\right\|^2 |J_k| \mathrm{d}\boldsymbol{\xi}}{\left|\Omega_k^2\right|^2}, \tag{9}$$

where $\boldsymbol{\phi}_k = \boldsymbol{\phi}_{|\mathcal{M}_k^R}$ is the restriction of the parametric mapping $\boldsymbol{\phi}$ over the sub-mesh $\mathcal{M}_k^R$, and $\widetilde{\boldsymbol{\pi}}_k = \left(\boldsymbol{\varphi}_k\right)^{-1} \circ \boldsymbol{\pi}_k \circ \boldsymbol{\phi}_k$ is the diffeomorphism $\boldsymbol{\pi}_k$ expressed in terms of the parametric coordinates of the physical mesh and the manifolds.

6.3. Discrete optimization framework for high-order meshes

In order to minimize Functional (9), in the discrete framework we approximate $\widetilde{\boldsymbol{\pi}}_k$ using a continuous piece-wise polynomial function, $\boldsymbol{\psi}_k \in \mathcal{V}_k^U$, where

$$\mathcal{V}_k^U = \left\{\boldsymbol{\psi} \in \left[\mathcal{C}^0(\mathcal{M}_k^R)\right]^{d_k} \text{ such that } \boldsymbol{\psi}_{|e} \in \left[\mathcal{P}^q(e)\right]^{d_k}, \forall e \in \mathcal{M}_k^R\right\}.$$

Note that we approximate $\widetilde{\boldsymbol{\pi}}_k$ by element-wise polynomials of degree at most q. In general, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are approximated using polynomials of different degree. Although we do not have yet a theoretical setting to select the value of q, it depends on the parameterization of the geometric entity, and the polynomial degree of the physical mesh, p. In our experience, if q is high enough the resulting meshes obtain optimal convergence order.

The discretized problem consists in computing

$$\underset{\substack{\boldsymbol{\phi}\in\mathcal{V}^P \\ \boldsymbol{\psi}_k\in\mathcal{V}_k^U}}{\operatorname{argmin}} E\left(\boldsymbol{\phi}, \boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_N\right) = \underset{\substack{\boldsymbol{\phi}\in\mathcal{V}^P \\ \boldsymbol{\psi}_k\in\mathcal{V}_k^U}}{\operatorname{argmin}} \sum_{k=1}^{N} \frac{\int_{\mathcal{M}_k^M} \left\|\boldsymbol{\phi}_k - \boldsymbol{\varphi}_k \circ \boldsymbol{\psi}_k\right\|^2 |J_k| \mathrm{d}\boldsymbol{\xi}}{\left|\Omega_k\right|^2}. \tag{10}$$

Since $\boldsymbol{\psi}_k$ is a mapping from $\mathcal{M}_k^R$ to $\mathcal{U}_k$, it defines a curved high order mesh on the parametric space of Σ_k. Intuitively, the physical mesh approximates the geometric entity, while the parametric mesh aligns the points of the physical mesh with the parameterization of the geometric entity. Finally, note that $\boldsymbol{\phi}$ and $\boldsymbol{\psi}_k$ depend only on the position of the physical and parametric nodes, respectively, and for this reason, so does Functional (10). Thus,

$$\inf_{\substack{\boldsymbol{\phi}\in\mathcal{V}^P \\ \boldsymbol{\psi}_k\in\mathcal{V}_k^U}} E\left(\boldsymbol{\phi},\boldsymbol{\psi}_1,\ldots,\boldsymbol{\psi}_N\right) = \inf_{\substack{\mathbf{x} \\ \mathbf{u}_1,\ldots,\mathbf{u}_N}} E\left(\mathbf{x},\mathbf{u}_1,\ldots,\mathbf{u}_N\right), \tag{11}$$

where $\mathbf{x}$ denotes the coordinates of the physical mesh nodes, and $\mathbf{u}_k$ the parametric coordinates of the nodes in the k-th parametric mesh.

To compute the disparity or to optimize a mesh, we consider different sets of design variables. Note that the functional in Equation (11) is written in terms of two sets of variables: the nodal positions of the physical meshes, and the nodal positions of the parametric meshes. If we optimize the functional with the parametric nodes as design variables, we obtain the disparity measure of the physical mesh and the target geometry. Alternatively, if we optimize the functional with the physical and parametric nodes as design variables, we obtain the physical mesh that approximates the target geometry with optimal disparity measure.

During the optimization process, we do not impose that the physical nodes have to be on the corresponding geometric entity. Thus, the resulting mesh approximates the target geometric model in a non-interpolative manner. Note that the interpolative case can be considered in the proposed framework by imposing that each node has to slide on the corresponding geometric entity it belongs to, similar to [53]. However, this adds further restrictions to the optimization problem and for this reason, the accuracy of the optimized interpolative mesh is lower than the accuracy of the non-interpolative mesh.

6.4. Optimization approach

The optimization of the non-linear objective function (11) is globalized using a backtracking line-search iterative algorithm, see [54]. That is,

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \alpha_j \mathbf{p}_j,$$

where $\mathbf{y}_j$ is the set of unknowns at the j-th iteration of the line-search algorithm, and α_j and $\mathbf{p}_j$ are the step-length and the advancing direction, respectively. In our framework, the advancing direction, $\mathbf{p}_j$, is determined using Newton's method. That is, $\mathbf{p}_j$ is computed by solving the linear system

$$\mathbf{H}E\left(\mathbf{y}_j\right)\mathbf{p}_j = -\nabla E\left(\mathbf{y}_j\right),$$

where ∇E and $\mathbf{H}E$ are the gradient and the Hessian of E, respectively. The step length is computed by imposing the Armijo rule, [54]

$$E\left(\mathbf{y}_j + \alpha_j \mathbf{p}_j\right) \leq c_1 \alpha_j \mathbf{p}_j \cdot \nabla E\left(\mathbf{y}_j\right), \tag{12}$$

where $c_1 = 10^{-4}$ according to [54]. To this end, we start with $\alpha_j = 1$, and divide it in half until the Armijo rule (12) is satisfied.

6.5. Post-process of the optimized mesh

After minimizing Equation (11), we have obtained a curved high-order mesh, $\mathcal{M}_P^*$, that approximates the target geometry in a non-interpolative manner. Nevertheless, it is possible to post-process the resulting mesh in order to obtain a new one, $\widehat{\mathcal{M}_P}$, that interpolates the geometric model. To this end, we define the post-processed mesh in terms of an iso-parametric mapping $\widehat{\boldsymbol{\phi}^*}$ as

$$\widehat{\boldsymbol{\phi}^*} = \sum_{i=1}^{N_n} \widehat{\mathbf{x}_i} N_i, \tag{13}$$

where N_n is the number of mesh nodes, $\{N_i\}_{1=1,\ldots,N}$ is a Lagrangian basis of polynomial shape functions, and $\widehat{\mathbf{x}_i}$ are the coordinates of the i-th node, computed as

$$\widehat{\mathbf{x}_i} = \boldsymbol{\pi}_k(\mathbf{x}_i) = \boldsymbol{\varphi}_k \circ \boldsymbol{\psi}_k \circ (\boldsymbol{\phi}_k)^{-1}(\mathbf{x}_i).$$

Since $\boldsymbol{\pi}_k$ is a diffeomorphism between the physical mesh and the target geometric model, the post-processed nodes are located on top of the geometric entities they belong to. Thus, we obtain an interpolative mesh. Note that we are able to avoid solving a non-linear problem to compute the orthogonal projection of each node of the mesh. Instead, it is only needed to evaluate the mapping $\boldsymbol{\pi}_k$ at the mesh nodes.

Although the post-processed mesh, $\widehat{\mathcal{M}_P}$, has a higher disparity value than the optimal one, in practice, the geometric accuracy of both meshes is of the same order.

7. Examples

The following examples show the features and capabilities of the presented formulation to check and optimize the geometric accuracy of a curved high-order mesh. We consider three quantities of interest: the proposed disparity measure, the maximum of the point-wise distances, d_p, and the error of the normals, $d_{\mathbf{n}}$, defined as

$$d_p := \left(E_p(\boldsymbol{\phi})\right)^{1/2} = \max_{k=1,\dots,N} \left[\sup_{\mathbf{x}\in\mathcal{M}_k^P} \left(\inf_{\mathbf{y}\in\Omega_k} \|\mathbf{x}-\mathbf{y}\| \right) \right] \tag{14}$$

and

$$d_{\mathbf{n}} := \left(E_{\mathbf{n}}(\boldsymbol{\phi},\boldsymbol{\psi})\right)^{1/2} = \left(\sum_{k=1}^{N} \frac{\int_{\mathcal{M}_k^P} \left\| \mathbf{n}_{\mathcal{M}_k^P}(\mathbf{x}) - \mathbf{n}_{\Omega_k}(\boldsymbol{\pi}_k(\mathbf{x})) \right\|^2 \mathrm{d}\mathbf{x}}{\left|\Omega_k^2\right|^2} \right)^{1/2}, \tag{15}$$

where $\boldsymbol{\pi}_k$ are the diffeomorphisms induced by $\boldsymbol{\psi}_k$, and $\mathbf{n}_{\mathcal{M}_k^P}$ and $\mathbf{n}_{\Omega_k}$ are the unit exterior normal of the physical mesh, $\mathcal{M}_k^P$, and the geometric entity, Ω_k, respectively. In all the cases, we denote by $\boldsymbol{\phi}$ and $\boldsymbol{\phi}^*$ the initial and optimized physical meshes, respectively, and by $\boldsymbol{\psi}_k$ and $\boldsymbol{\psi}_k^*$ the initial and optimized parametric meshes. Note that we do not optimize neither Equation (14) nor (15) in any case, we use them as alternative ways to check the accuracy of the meshes.

In order to approximate the maximum point-wise distance, we use a point sampling over the elements and take the maximum distance over the sampling. Note that we do not use the integration points of the optimization process to sample the element. Instead, we use a bigger number of sampling points than the number of integration points to obtain a fair result.

7.1. Implementation

The optimizer needs the parametric coordinates of the mesh nodes associated with the different curves and surfaces of the compound model. To meet this requirement, we have implemented the code in Python, Anaconda distribution [55], and we have linked it, through SWIG [56], with our hierarchical mesh generation code [57,58]. The code also features an implementation of Newton's method equipped with a backtracking line-search strategy. The linear systems arising from the optimization are solved with the SuperLU sparse direct solver provided by SciPy [59], and NumPy [60] libraries. The first and second disparity derivatives are also implemented in Python and are computed by assembling element contributions. Our SWIG [56] wrapper to the OpenCASCADE [51] library allows importing STEP geometries and provides access to geometry queries. The wrapper allows querying the parameterization values and the first and second derivatives. It also allows projecting points to the corresponding geometric entities. We use nodal high-order basis functions to represent the piece-wise polynomial physical and parametric meshes. For segments, we use Legengre-Gauss-Lobatto interpolation point distributions. The tensor product of these interpolation points determines the distribution for quadrilateral elements. To approximate the integrals, we use Gaussian quadrature to integrate exactly polynomials of degree up to three times the degree of the parametric meshes.

7.2. Interpolation points: logarithmic spiral

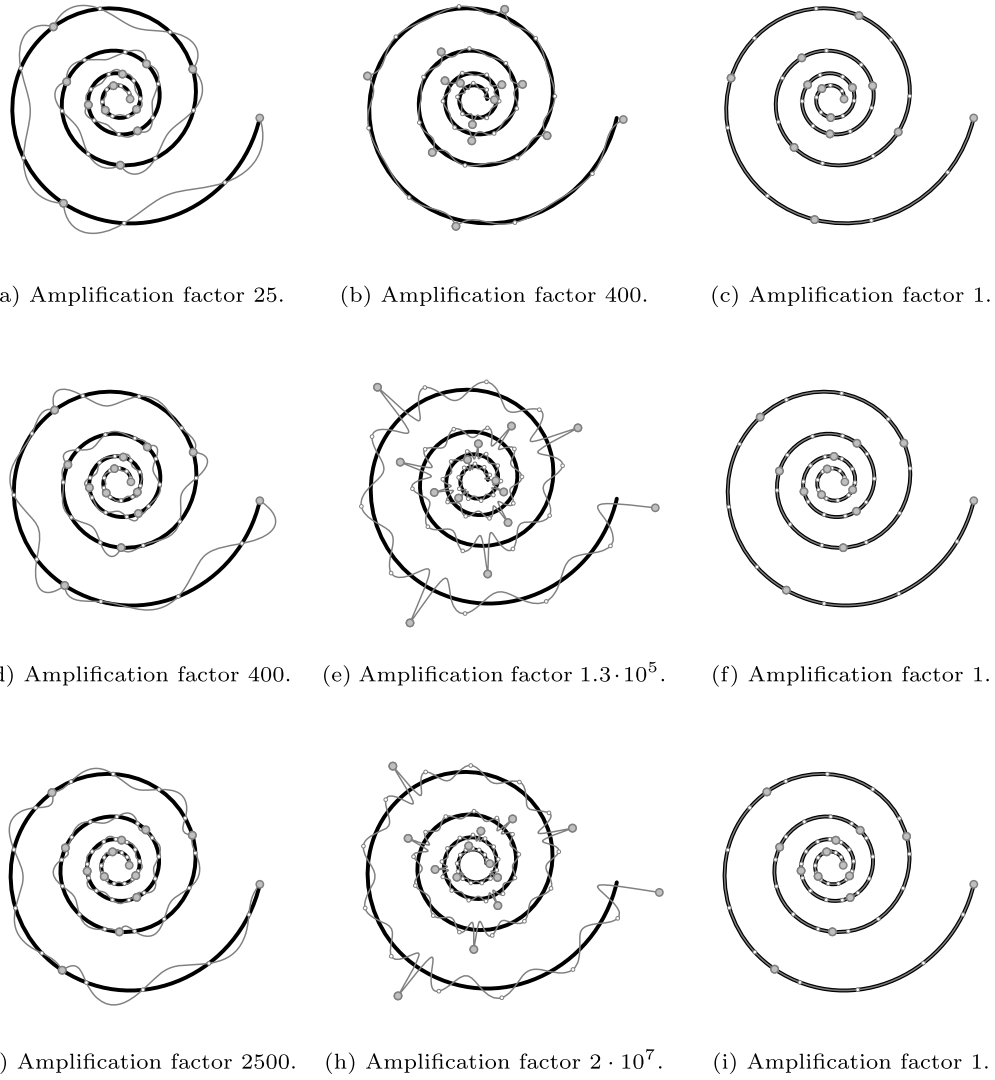
In this example, we highlight the number of interpolation points of the initial interpolative and the final approximate meshes. To this end, we generate three meshes for a logarithmic spiral composed of thirteen elements of polynomial degrees three, four and five, see Figs. 4a, 4d and 4g, respectively. To generate these meshes, we use the standard interpolative scheme in which a high-order nodal distribution determines $p+1$ interpolation points in each element.

To visually differentiate the mesh from the target geometry, we consider the amplified error curve. The amplified error curve is critical to inspect highly accurate meshes visually. It allows visualizing small geometric errors and the intersection points between the mesh and target geometry. We define the amplified error curve as:

$$e_\mu = \varphi \circ \psi^* + \mu e,$$

where $e = \phi - \varphi \circ \psi^*$, and μ is the amplification factor. That is, we visualize the mesh as an amplified error displacement of the target geometry. Note that for $\mu = 1$, the geometric error is not amplified, and we obtain the original mesh.

In Fig. 4, we highlight the interpolation points as the intersection of the amplified error curve (gray) with the target curve (black). Figs. 4b, 4e and 4h show the optimized meshes of polynomial degree three, four and five, respectively. Note that the optimized meshes increase the geometric accuracy by several orders of magnitude. Accordingly, the oscillations of the error curve for the optimized meshes have a smaller amplitude than for the initial meshes. Moreover, there are more



(a) Amplification factor 25. (b) Amplification factor 400. (c) Amplification factor 1.

(d) Amplification factor 400. (e) Amplification factor $1.3 \cdot 10^5$. (f) Amplification factor 1.

(g) Amplification factor 2500. (h) Amplification factor $2 \cdot 10^7$. (i) Amplification factor 1.

Fig. 4. Meshes generated for a logarithmic spiral and amplified geometric error. In columns, amplified error for the initial interpolative meshes (a), (d) and (g); amplified error for the optimized approximative meshes (b), (e) and (h); and final optimized meshes (c), (f) and (i). In rows, meshes of polynomial degree three (a), (b) and (c); polynomial degree four (d), (e) and (f), and polynomial degree five (g), (h) and (i).

oscillations that allow interpolating the target geometry in $p-1$ additional points, for a total of $2p$ interpolation points. The optimization algorithm automatically computes a curve reparameterization to interpolate these additional points and, hence, approximate the target geometry with higher accuracy while using the same polynomial degree.

7.3. Algebraic super-convergence order: logarithmic spiral

In this example, we show that we can generate a series of meshes that super-converge algebraically to the target geometry. To this end, we use the proposed numerical method to approximate a geometric model according to the disparity measure. According to standard approximation theory, we should expect that the disparity converges to zero proportionally to h^{p+1}, where h is a measure of the element size and p the polynomial degree of the mesh. We should also expect that the order of approximation to unitary normal vectors is one order less, that is h^p. Remarkably, with our non-standard approach, we obtain higher orders of approximation to geometry, and to unitary normal vectors.

To show the properties of the proposed method, in this example we consider several cases. For each case, we approximate a logarithmic spiral parameterized as

$$\boldsymbol{\varphi}(t) = \mathrm{e}^{t/10}\left(\sin(t), \cos(t)\right), \qquad t \in [0, 8]. \tag{16}$$

To this end, we generate a series of meshes with polynomial degree between one and four and then we perform a convergence order analysis of different quantities of interest.

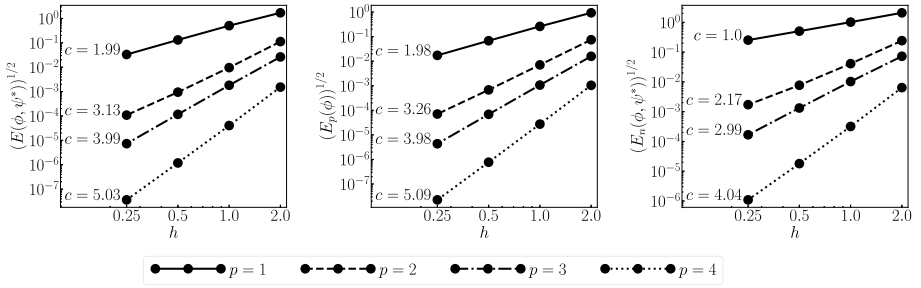


Fig. 5. Convergence order of the disparity measure, the maximum of the point-wise distances, d_p, and the error of the normals, $d_{\mathbf{n}}$, for the initial interpolative meshes.

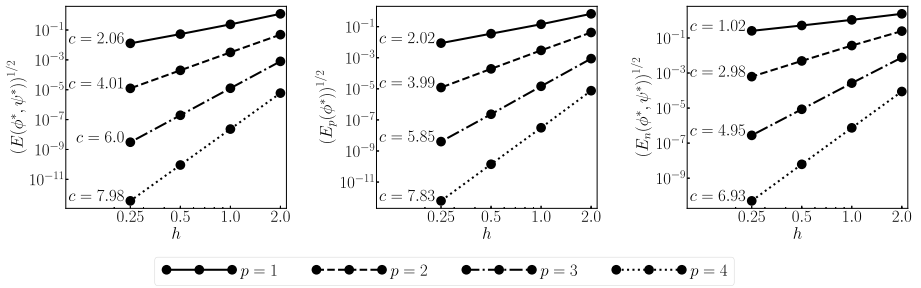


Fig. 6. Convergence order of the disparity measure, the maximum of the point-wise distances, d_p, and the error of the normals, $d_{\mathbf{n}}$, for the optimized approximative meshes.

Case (i): disparity measure of the initial interpolative meshes. In this example, we compute the proposed disparity measure of the initial interpolative meshes. Thus, we fix the physical mesh and optimize Functional (10) in terms of the parametric mesh. The objective of this case is to compute the disparity measure and the error of the normals of the initial interpolative meshes, see Fig. 5. Both the proposed disparity measure and the maximum of the point-wise distances converge with order $p+1$, while the unit normals of the mesh converge with order p. Note that the convergence order of the normals is one order less than the convergence of the disparity.

Case (ii): disparity measure of the optimal non-interpolative meshes. In this case, we optimize Functional (10) with respect to the physical and parametric mesh. Thus, we obtain the optimal physical mesh in the sense of the proposed disparity measure. Fig. 6 shows the convergence order of the disparity measure, the maximum of the point-wise distances, and the error of the normals for the optimized non-interpolative meshes. Both the disparity and the point-wise distances converge with order $2p$, while the normals converge with order $2p-1$. While we do not have a formal proof for the superconvergence of the disparity measure and the normals, we have observed this behavior in other practical examples. Note that although the convergence order of the initial interpolative meshes is $p+1$, we are obtaining a superconvergence order of $2p$. Thus, the geometric error is reduced by several orders of magnitude just by relocating the nodes, especially when using high polynomial degrees.

7.4. Sensitivity of the polynomial degree of the parametric meshes: circular arc

To compute the proposed disparity measure, we need to define the polynomial degree of the parametric meshes, denoted as q. Thus, in this example we study the dependence of the disparity measure computation on the q parameter using a circular arc parameterized in three different manners as

$$\begin{aligned}
\boldsymbol{\varphi}^1(t) &= (\sin(t\pi), \cos(t\pi)) && t \in [0,1],\\
\boldsymbol{\varphi}^2(t) &= \left(\sin\left(t^2/\pi\right), \cos\left(t^2/\pi\right)\right) && t \in [0,1],\\
\boldsymbol{\varphi}^3(t) &= \left(\sin\left(\pi\,\frac{\mathrm{e}^{2t/\pi}-1}{\mathrm{e}^2-1}\right), \cos\left(\pi\,\frac{\mathrm{e}^{2t/\pi}-1}{\mathrm{e}^2-1}\right)\right) && t \in [0,1].
\end{aligned}$$

The first parameterization corresponds to the usual definition of a circular arc using the angular parameter. The second and the third parameterizations correspond to a polynomial and an exponential re-parameterization of the circular arc, respectively.

We generate two sets of meshes to approximate the circular arc. The first set is composed of three meshes of polynomial degree three with four, eight, and sixteen elements, respectively. We approximate the disparity measure of these meshes

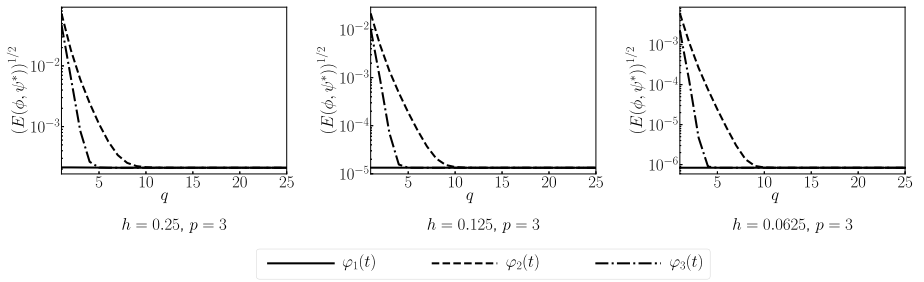


Fig. 7. Evolution of the approximated disparity measure of the curved high-order meshes with polynomial degree three for different values of q.

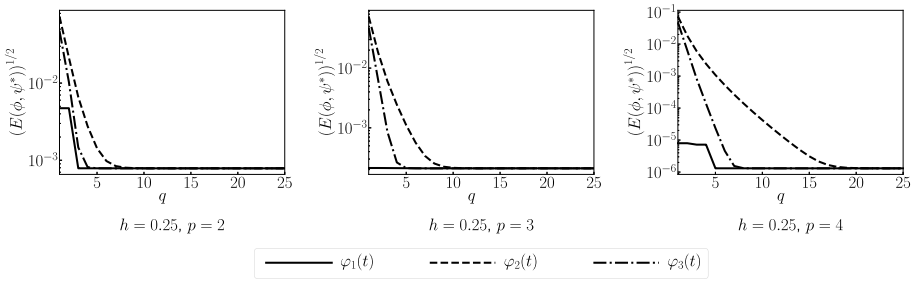


Fig. 8. Evolution of the approximated disparity measure of the curved high-order meshes with four elements for different values of q.

against the three parameterizations of the circular arc using different values of q, see Fig. 7. The behavior of the evolution is similar in the three meshes, and it is independent of the number of elements of the physical mesh. Note, however, that as the number of elements increases, the disparity measure decreases. Thus, the finer mesh better approximates the target geometry.

The second set of meshes is composed of three meshes of four elements and polynomial degree two, three, and four, respectively. We again approximate the disparity measure of the curved high-order meshes against the three parameterizations of the circular arc for different values of q, see Fig. 8. As in the first case, the value of the disparity measure converges to a single value. However, as the polynomial degree of the physical mesh increases, higher values of q are required to converge the value of the disparity measure.

In this example, we have shown that the value of the q parameter depends on the parameterization of the geometric entity. As expected, we need more degrees of freedom to compute the correct value of the disparity measure, when the parameterization is more *complex*. Moreover, the value of q depends on the polynomial degree of the physical mesh. The main reason is that as the *complexity* of the physical mesh increases, it is also necessary additional degrees of freedom in the parametric spaces. Nonetheless, q does not depend on the number of elements of the mesh. Since the parametric meshes have the same topology as the physical mesh, when the physical mesh is refined, so are the parametric ones. Thus, it is not necessary to increase q in order to better approximate the value of the disparity measure, even when the physical mesh is more *complex*.

7.5. Higher geometric accuracy for the same resolution: sphere

We optimize the disparity measure for different polynomial degrees to obtain mesh approximations to a compound model described by a CAD boundary representation. The implementation imports a STEP file describing a trimmed sphere of radius one. The model is composed of four surfaces, twelve curves, and eight points. The original sphere patches are cut along two iso-curves parallel to the equator to obtain machine-accurate trims and, thus, negligible gaps. These negligible gaps ensure that a loose gap tolerance does not limit the optimization of the disparity.

We generate several meshes of the same resolution for polynomial degrees one, two, four, and eight. As we increase the polynomial degree, we also double the element size, and therefore, the ration h/p remains constant. Fig. 9 shows the initial interpolative meshes (left column) and the optimized ones (right column). We have colored the meshes according to the point-wise distance, defined for each point of the physical mesh, $\mathbf{x}$, as

$$d_p(\mathbf{x}) = \inf_{\mathbf{y}\in\Omega} \|\mathbf{x}-\mathbf{y}\| \,.$$

The optimized high-order meshes are more accurate than the initial ones, and they reduce the point-wise distance by several orders of magnitude. Note that there are oscillations in the point-wise distance even for the mesh of highest polynomial degree. This is because the spherical patches cannot be exactly represented using polynomials. Nevertheless,

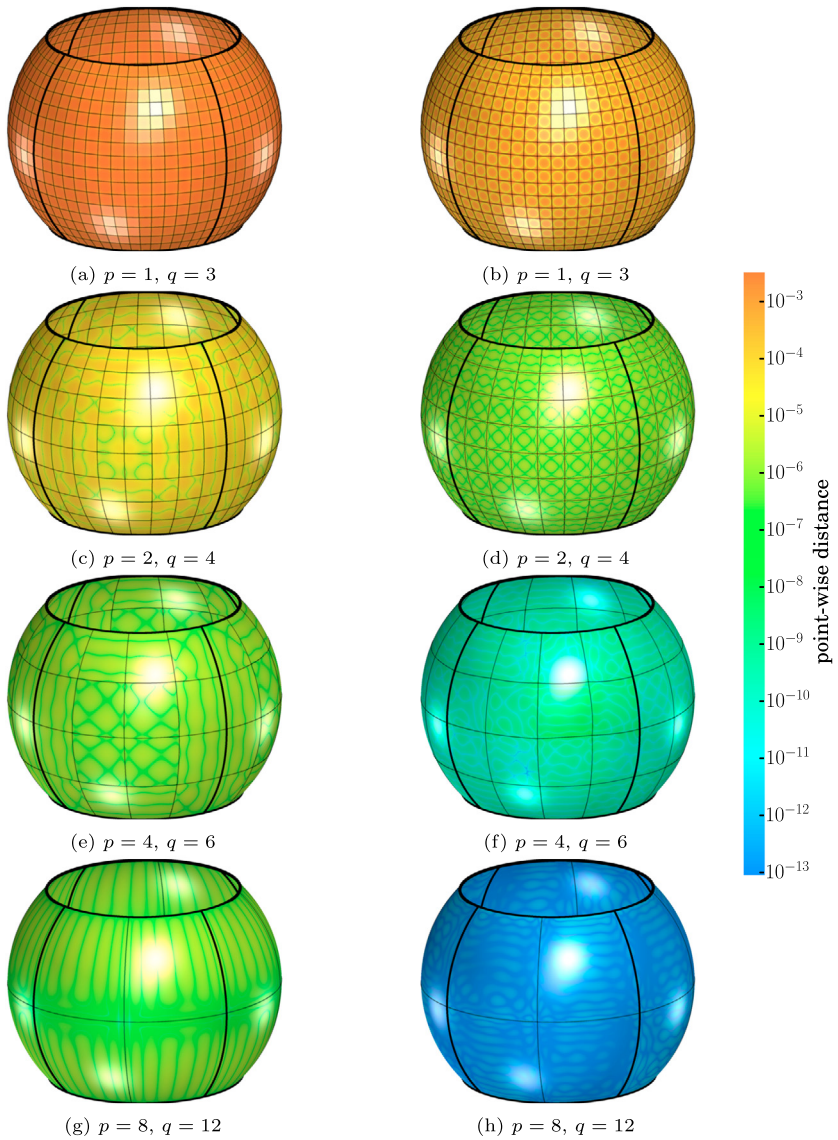


Fig. 9. Meshes generated for a spherical surface. In columns, initial interpolating meshes (a), (c), (e) and (g); and optimized approximating meshes (b), (d), (f) and (h). In rows, the polynomial degrees of the physical and parametric meshes. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

the amplitude of the oscillations is reduced as the polynomial degree is increased, and the oscillations for the mesh of polynomial degree eight are of the order of 10^{-12}.

Fig. 10 shows the evolution of the disparity measure, the maximum of the point-wise distances, and the error of the unit normals for the initial interpolating and the optimized curved high order meshes. Note that the three quantities of interest present a steady descent for the optimized meshes. However, the initial interpolative meshes of polynomial degree four and eight approximate the geometric model with similar accuracy. It is worth noticing that the optimized meshes not only approximate the geometry with high-accuracy but also approximate the normals of the geometry with high-accuracy, see Fig. 10. This is so even we do not explicitly target the maximum of the point-wise distances or the accuracy of the normals.

7.6. Algebraic super-convergence order of interpolative meshes: torus

The objective of this example is to compare the geometric error obtained by the optimized approximative mesh and the post-processed interpolative mesh. To show that, we generate a curved high-order mesh of polynomial degree three for a toroidal CAD boundary representation, see Fig. 11a. The model is composed of 16 vertices, 32 curves, and 16 surfaces. To not limit the optimization of the disparity measure, we devised the model to present negligible gaps. In Figs. 11b and

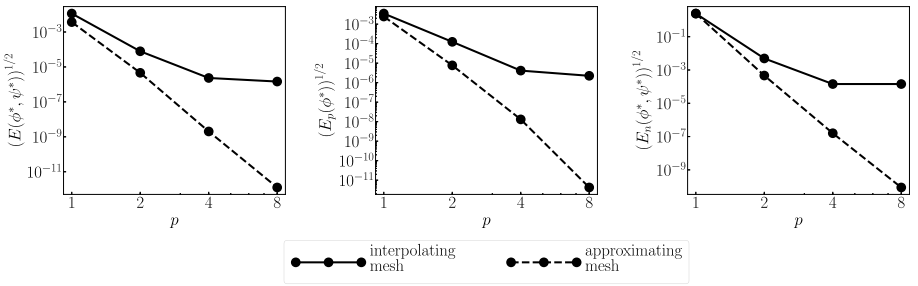


Fig. 10. Evolution of the disparity measure (left), the maximum of the point-wise distances, d_p, (center), and the error of the unit normals, $d_{\mathbf{n}}$, (right) for different polynomial degrees.

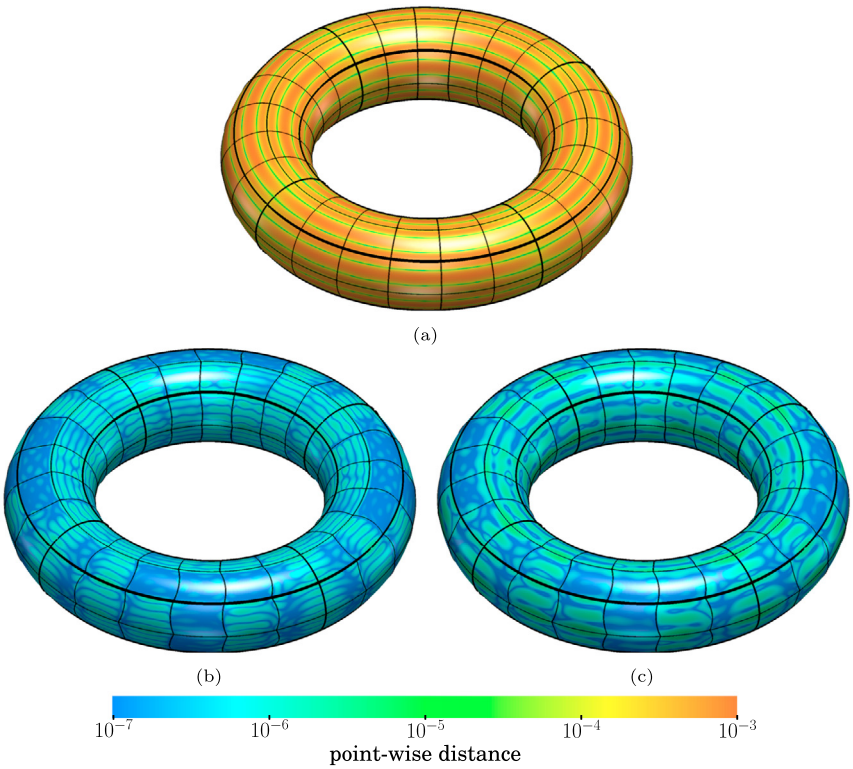


Fig. 11. Meshes generated for a toroidal surface: (a) initial interpolative mesh; (b) optimized approximative mesh; and (c) post-processed interpolative mesh.

11c, we show the optimized approximative mesh and the post-processed interpolative one, respectively. Note that both meshes present geometric errors of the same order, although the approximative mesh is the optimal one in the sense of the proposed disparity measure.

In the second part of this example, we show the convergence order of the interpolative mesh obtained by post-processing the optimized approximative mesh. To this end, we generate a series of meshes for the toroidal geometric model using polynomial degrees between one and three, see Fig. 12. The initial interpolative meshes converge to the target geometry with order $p+1$, begin p the polynomial degree of the mesh. Both the optimized and the post-processed meshes converge to the target geometry with order $2p$. Thus, in this example, we show that it is possible to obtain super-convergent high-order meshes even when interpolating the geometric model.

8. Discussion and future work

After detailing the methods and results to measure and improve the geometric accuracy of high-order meshes, we present a discussion on many aspects related to CAD models, optimization unknowns, non-interpolative and interpolative meshes, super-convergence, and iso-resolution. We finalize the section with the limitations and opportunities we consider interesting to address shortly.

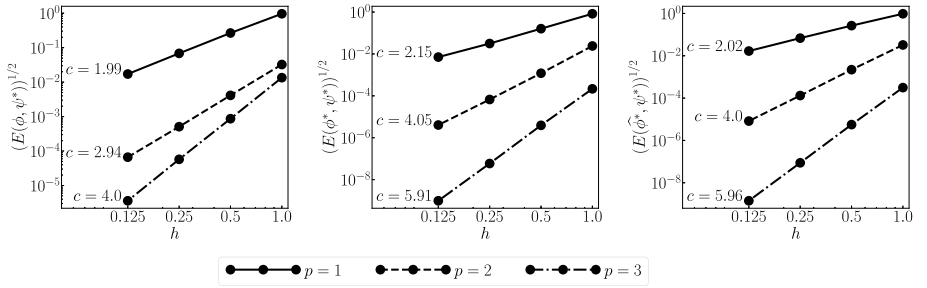


Fig. 12. Convergence order of the disparity measure for the initial interpolative meshes, the optimized approximative meshes and the post-processed interpolative meshes generated for a torus.

8.1. CAD models

The goal of the proposed disparity is to measure and improve the geometric accuracy of curved high-order meshes that approximate a target compound model represented by a CAD model. Accordingly, we have devised the disparity measure to deal with:

- *Dimensional hierarchies of geometric entities*. Standard CAD models are composed of several entities with different topological dimensions: vertices (0D), curves (1D), surfaces (2D), and volumes (3D). Thus, to apply our disparity measure in standard CAD models, we have considered a definition that accounts for target compound models. In the definition, we have also assumed that the dimension of each geometric entity is less than the dimension of the embedding space. We do it so since the geometric accuracy only needs to depend on the distance between the mesh boundary and the entities on top of the model boundary.
- *Multiple entity parameterizations*. We have proved that the definition of the disparity measure is independent of the selected parameterization of the geometric entities. This independence is an important feature since a geometric entity can allow different CAD parameterizations, often of low quality, which can potentially lead to different values of the disparity measure. In other words, we have seen that the disparity measure is an intrinsic property of the geometric model.
- *Trimmed surfaces*. The utilization of surfaces trimmed with boundary curves is a key modeling tool in modern CAD packages. Accordingly, we have devised the disparity to allow the utilization of trimmed surfaces once we have an initial straight-edged mesh matching the trimming entities. Specifically, the minimization of the disparity functional only involves the parameterization of the geometric entities and their derivatives, and no topological information of the corresponding trimming entities is required.

8.2. Optimization unknowns: physical mesh and parametric mesh

We have detailed how to optimize the disparity measure to improve the geometric accuracy of a curved high-order mesh. The interpretation of this optimization corresponds to an adaptive process with two different sets of unknowns: the physical mesh, and the parametric meshes. Each parametric mesh is an auxiliary curved high-order mesh in the parametric space of the corresponding geometric entity.

Each set of unknowns targets different goals. During the optimization, the physical mesh gets closer to the target manifolds since it tries to match the physical image of the corresponding parametric meshes. At the same time, the parametric meshes approximate and non-linearly track the orthogonal projection of the integration points of the physical mesh onto the target geometry. Hence, using parametric meshes we can avoid the costly computation of orthogonal projections (inverse problem) and their derivatives, since we only need to evaluate the CAD parameterizations (forward problem) and their derivatives.

Furthermore, the parametric meshes enable to feature the independence on entity parameterization. In practice, this independence is achieved if the polynomial degree of the parametric meshes is high enough, as shown in the test cases. Although we do not have an ideal setting to choose the polynomial degree of the parametric meshes, we have seen that it depends on the *complexity* of the parameterization of the geometric entity and the polynomial degree of the physical mesh.

8.3. Non-interpolative and interpolative boundary points

Meshes non-interpolating the domain boundary have two main advantages when compared to interpolative ones. First, we can obtain more accurate meshes since imposing an interpolative mesh adds additional constraints to the optimization problem, and thus, it leads to a less general solution. Second, since there are no additional constraints, we minimize an unconstrained optimization problem, which is computationally more manageable than a constrained one. Accordingly, during the optimization process, we do not impose that the physical mesh has to interpolate the geometric model.

Although we can optimize non-interpolative meshes to have higher geometric accuracy, we can obtain interpolative meshes with the same order of geometric accuracy. To this end, we have detailed a post-process to relocate the nodes of an optimized mesh to be on top of the target geometric model. In this post-process, we avoid computing the orthogonal projection of each node (expensive inverse problem). Instead, we compute the position of the new nodes by evaluating the diffeomorphism obtained during the optimization process (cheaper forward problem).

The main difference with the initial interpolative mesh and the post-processed one is the location of the interpolation points. The initial mesh interpolates the geometry at fixed points. On the contrary, the post-processed interpolative mesh interpolates the geometry at specific points computed using the optimization process. These interpolative points depend on the geometry and tend to increase the resolution in high-curvature areas and decrease the resolution in low-curvature areas.

8.4. Super-convergence

The examples provide experimental evidence that the proposed disparity measure converges to zero with super-convergent order $2p$, where p is the polynomial degree of the physical mesh. Furthermore, although in the minimization problem we do not explicitly optimize it, the maximum point-wise distance also super-converges to zero with order $2p$. As desirable, the outer unit normal of the physical mesh super-converges, only losing one order, to the outer unit normal of the target boundary with order $2p-1$. While we do not have a mathematical proof for the obtained convergence order for the disparity measure and the outer unit normals, we have observed this behavior in all the tested geometries.

Consequently, we provide numerical evidence that the error of approximating a manifold, geometric error, converges to zero with order $2p$. That is, the proposed super-convergent disparity measure is by construction a measure of the error of curved high-order mesh to approximate a target manifold. We have proved and shown that different re-parameterizations provide the same results. And thus, although we characterize the manifold using a parameterization, we are approximating the manifold and not the given parameterization.

Standard *a posteriori* approaches to curve straight-edged meshes with interpolative boundary nodes lead to an order of geometric accuracy of $p+1$. That is, when curving a straight-edged mesh, with an *a posteriori* approach, the partition in straight-edged elements is chosen in advance by a linear (multi-linear) mesh algorithm considering only low-order geometric information. Then, the whole mesh is elevated to feature high-order nodes, where the boundary nodes are projected to be on top of the target boundary. Finally, the interior nodes are relocated to accommodate the curvature of the displaced boundary nodes. This *a posteriori* approach is equivalent to approximate a function on a fixed reference straight-edged mesh, but now the function to approximate is the target geometry. This is how *a posteriori* curved meshing leads to standard high-order accuracy.

We want to highlight that the geometric accuracy order of $p+1$ featured by standard curved meshes with interpolative boundary points is not in disagreement with the super-convergent order of $2p$ of our approach. Note that in standard mesh curving, the linear meshing algorithm determines in advance the partition in straight-edged elements considering only low-order geometric information. Moreover, the distribution of the physical high-order nodes is performed by trying to replicate a reference distribution of high-order nodes in a reference frame. Thus, it only takes into account the *element quality*.

On the contrary, what enables super-convergence in our approach is to adapt the location of the physical mesh nodes, even though we keep the same partition into elements of the target geometry. That is, the use of interpolative or non-interpolative meshes is not significant to feature a super-convergent order to the target geometry. In both cases we obtain a curved adapted mesh with finer resolution on highly variant regions and coarser resolution on lesser variant regions. Therefore, we are able to significantly improve the geometric accuracy.

The super-convergence results for 2D curves approximated with interpolative points, and non-interpolative points agree with the still unproved conjectures on super-interpolation posed in the area of computer aided geometric design [61]. The interpolative case follows directly from the super-interpolation conjecture, which states that it is possible to interpolate a 2D curve in $2p$ points. In the non-interpolative case, the super-interpolative conjectures provide a lower bound for the order of super-convergence. This lower bound is so since non-interpolative meshes are more general than interpolative ones and hence, they have to feature at least the same order of accuracy. Although an equivalent conjecture for 3D surfaces approximated with interpolative points has not been discussed in [61], an analogous reasoning derived from super-interpolation might be developed for 3D surfaces.

8.5. Iso-resolution

In Example 7.5, we have seen that we can significantly improve geometric accuracy for a given resolution by doubling the polynomial degree, doubling the element size, and optimizing the disparity. As we can see in Fig. 9, the optimized meshes feature significantly smaller amplitude of the oscillations than the initial interpolative meshes. Furthermore, we observe a higher number of oscillation valleys, where the geometric error is closer to zero, in the optimized non-interpolative mesh than in the interpolative one.

Intuitively, these additional valleys indicate that if we post-process the optimized non-interpolative meshes we can interpolate the target geometry in more points, where geometric error is equal to zero, than with standard approaches. This

is possible since in standard approaches the location of interpolative nodes is fixed while with our approach the location depends on the target geometry.

8.6. Practical aspects

We have proposed a disparity formulation that accounts for compound models described with a CAD boundary representation. However, to handle non-academic CAD models, we need to overcome the limitations of the current implementation. With this implementation, we show numerical results for compound surfaces described by simple CAD boundary representations. Following, we detail the current implementation limitations and the future solutions according to CAD and implementation aspects:

Tolerance gaps. The definition of the disparity handles tolerance gaps in compound models. However, the maximal CAD tolerance limits the disparity of the optimal boundary mesh. The tolerance gaps appear around the curve interfaces between incident surfaces and the point interfaces between incident points. The curved mesh approximation features a hierarchy of elements approximating the vertices, the physical curves, and the physical surfaces. Thus, the elements adjacent to a vertex or a curve have a disparity ruled by the respective tolerance gaps. The influence of the gaps is smaller on those elements away from the surface boundary. Real CAD models have a relative gap tolerance around 10^{-5} that limits the mesh with minimal disparity.

Accurate trimming. If the compound model features tight gap tolerances, the method can obtain highly accurate and super-convergent surface meshes, see Examples 7.5 and 7.6. To obtain compound models with tight trimming tolerance in current CAD codes, one needs to skip tolerance-based operations. There are some trimming operations limited only by machine accuracy, such as cutting surfaces along iso-curves. Although this workaround limits the model complexity, it allows illustrating the accuracy potential of the proposed approach. For instance, in Example 7.5, we consider a sphere composed of four trimmed surfaces. The original sphere patches are cut along two iso-curves parallel to the equator, thus obtaining negligible gaps.

Degenerated edges. The method needs a hierarchical mesh that accounts for degenerated edges. The standard meshing workaround is to understand the degenerated edge as a vertex, and hence, have only one master point in physical coordinates and one slave node in parametric coordinates. Using the hierarchical mesh, we can set design variables to approximate the corresponding vertex points, in this case, for the degenerated edge. In contrast, the corresponding parametric coordinates remain fixed.

Surface smoothness. NURBS allows providing continuous surfaces with discontinuities in the derivatives referred to as kinks. To deal with kinks and still obtain super-convergence, we consider two possibilities to explore soon. First, one can split the surface along the kink and then apply the disparity optimization in the compound surface. Second, one can embed in the mesh a set of edges that discretize the discontinuity line. Then, one can optimize the disparity to expect a mesh that aligns with the discontinuity line.

Integrated implementation. We need to integrate the optimization method with a commercial mesh generator and geometry kernel to deal with more complex geometries. Note that the disparity optimization needs data not available in the output of a standard mesh generator. Specifically, we need a linear mesh associated with the model hierarchy, a parametric mesh for each model entity, a one-to-one identification between parametric and physical mesh entities accounting for model orientation, an initial physical mesh of degree p, and an initial parametric mesh of degree q. To meet these requirements, we found it convenient to develop an intrusive first implementation inside our academic hierarchical mesh generation code [57,58]. We use the OpenCASCADE [51] geometry kernel to perform all the required forward and inverse evaluations. Shortly we would like to integrate the optimization method with the Pointwise mesh generator [62] and the Geode geometry kernel [63]. This integration will be key to optimize the disparity with CAD-based virtual geometries.

Parameterization second derivatives. Since we use second-order optimization, we only can apply the implementation to compound models where all the entity parameterizations have second derivatives. We have favored a second-order optimization implementation since it features quadratic convergence. If we use a first-order optimization implementation, we only need the entity parameterizations to feature first derivatives. However, the convergence rates are slower.

Virtual geometry. The current method does not work with virtual geometry models. That is, we cannot virtually merge adjacent geometry entities. This limitation is so since we explicitly need a continuous piece-wise mesh composed of continuous piece-wise meshes. Each sub-mesh has to be associated with the parameterization of the original curves and surfaces of the compound model.

8.7. Future work

We have obtained numerical evidence of disparity super-convergence for curves and surfaces that seem to agree with existent results on geometry interpolation. Specifically, our results for curves agree with the conjecture on curve super-interpolation stated in [61]. The conjecture states that it is possible to interpolate in exactly 2p points a sufficiently smooth curve with a polynomial parameterization of degree p. The 2p interpolation points are previously undetermined and dependent on the target curve. Similarly, our numerical evidence of super-convergence for surfaces would agree with an analogous super-interpolation conjecture but extended to surfaces. Nevertheless, the super-interpolation conjectures only indicate that our numerical evidence can be expected but do not provide theoretical proof. Accordingly, in the near future, we will work on proving the super-convergence statement evidenced by our numerical results.

It might be interesting to combine the approach herein with recently proposed methods for topology optimization on curved meshes [64,65]. This combined approach would potentially improve the accuracy of the physical mesh for a fixed number of nodes. This improvement would be so since the topology changes provide additional flexibility to align the mesh with the target geometry. Thus, given a group of elements defining a cavity, we should explore those topological changes that improve the mesh accuracy of the remeshed cavity. The curved cavity operation provides an initial approximation to the disparity equilibrium configuration on the disc. To obtain the optimal configuration, we need to fix the design variables determining the cavity boundary. Then, the equilibrium position of the interior design variables is recomputed.

In this work, we focus on measuring and optimizing the disparity of curved surface mesh approximations to target compound geometry. We understand that super-accurate piece-wise polynomial meshes could provide an alternative boundary representation of a compound model. Furthermore, these super-accurate meshes might be relevant to bound valid curved volume meshes. To this end, we have already explored the combination of a disparity measure with a distortion-based mesh curving approach [47]. Nevertheless, we would like to extend the previous approach with the new fully super-convergent disparity measure. This extension might enable generating high-quality curved meshes using a lower polynomial degree than with standard interpolative meshes while keeping the required geometric accuracy.

For the examples considered, the number of non-linear iterations ranges from tens to hundreds of non-linear iterations. Each non-linear iteration corresponds to a call to the sparse-direct solver. Nevertheless, the memory and the CPU time has not been a problem since the example meshes are coarse, and hence, the number of degrees of freedom is also small. We have to highlight that in the convergence studies, the memory and CPU times have also not been an issue, although the degrees of freedom for surfaces (curves) scale quadratically (linearly) with the reciprocal of the element size. Shortly, we will consider the possibility of devising a specific-purpose solver to deal with finer and more complex meshes. The new implementation should reduce the required non-linear iterations, the required memory, and the CPU time. To reduce the number of non-linear iterations, and thus the CPU time, we are considering performing solution continuation by adding an external loop to sweep the polynomial degree. Furthermore, to further reduce the CPU time, we should favor the steepest descent direction, computationally cheaper, in the first iterations, and automatically favor Newton's direction, more expensive but with quadratic convergence order in the convergence region. To reduce the required memory, we should substitute the sparse-direct solver with a preconditioned iterative solver. We should consider a pre-conditioner reducing the total number of linear iterations and, thus, the CPU time.

9. Concluding remarks

Our numerical experiments show evidence that for 3D (2D) compound models bounded by surfaces (curves) we can approximate the geometry and the unit normal vectors with super-convergent orders of $2p$, and $2p-1$, respectively. We have checked this for sufficiently smooth target surfaces (curves), and polynomial degrees up to 3 (4, respectively). Note that the disparity measure converges to zero with order $2p$. Thus, since the disparity measure is an upper bound of the geometric error, we also have that the geometric error converges to zero with order $2p$.

We have demonstrated that we can meet these super-convergent orders of geometric accuracy both for meshes with non-interpolative and interpolative boundary points. To this end, the non-interpolative points of the proposed super-convergent mesh optimization can be cheaply relocated to be on top of the boundary manifolds, with the proposed post-process, and still preserve the same super-convergent order. Thus, although non-interpolative optimized meshes are in practice slightly more accurate, practitioners can still use interpolative meshes but now featuring super-convergent orders of geometric approximation.

In conclusion, it is possible to significantly increase the order of approximation to geometry for curved parametric meshes of a given polynomial degree. This result is of general interest in finite element analysis with curved elements, both for formulations in a reference and physical frame, where high accuracy is required to approximate geometry.

In perspective, with our approach, we might use a smaller polynomial degree to approximate a geometry than to approximate a solution without hampering the required geometric accuracy for high-order analysis. This is plausible since for the same polynomial degree ($k = p > 1$), the order of geometric accuracy is higher than the expected order of convergence $p+1$ of the solution. This alternative might be critical in reference frame formulations, where practitioners prefer smaller polynomial degrees to represent geometry, and thus, mitigate the reduction of order of convergence of approximation to solution originated by using parametric curved elements.

CRediT authorship contribution statement

Eloi Ruiz-Gironés: Conceptualization, Methodology, Software, Writing – review & editing. **Josep Sarrate:** Conceptualization, Funding acquisition, Methodology, Writing – review & editing. **Xevi Roca:** Conceptualization, Funding acquisition, Methodology, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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. This work has been supported by FEDER and the Spanish Government, Ministerio de Ciencia, Innovación y Universidades grant project contract PGC2018-097257-B-C33. 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] S. Dey, M. Shephard, J. Flaherty, Geometry representation issues associated with p-version finite element computations, Comput. Methods Appl. Math. 150 (1–4) (1997) 39–55.
[2] S. Dey, R. O'Bara, M.S. Shephard, Curvilinear mesh generation in 3D, Comput. Aided Des. 33 (2001) 199–209.
[3] X. Luo, M.S. Shephard, J.-F. Remacle, R. O'Bara, M. Beall, B. Szabó, R. Actis, P-version mesh generation issues, in: Proc. 11th Int. Meshing Roundtable, Springer, Berlin, Heidelberg, 2002, pp. 343–354.
[4] X. Luo, M.S. Shephard, R. O'Bara, R. Nastasia, M. Beall, Automatic p-version mesh generation for curved domains, Eng. Comput. 20 (3) (2004) 273–285.
[5] M.S. Shephard, J.E. Flaherty, K. Jansen, X. Li, X. Luo, N. Chevaugeon, J.-F. Remacle, M. Beall, R. O'Bara, Adaptive mesh generation for curved domains, Appl. Numer. Math. 52 (2–3) (2005) 251–271.
[6] P.-O. Persson, J. Peraire, Curved mesh generation and mesh refinement using Lagrangian solid mechanics, in: Proc. 47th AIAA, 2009.
[7] D. Moxey, M. Green, S. Sherwin, J. Peiró, An isoparametric approach to high-order curvilinear boundary-layer meshing, Comput. Methods Appl. Mech. Eng. 283 (2015) 636–650.
[8] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, Optimization of a regularized distortion measure to generate curved high-order unstructured tetrahedral meshes, Int. J. Numer. Methods Eng. 103 (5) (2015) 342–363, https://doi.org/10.1002/nme.4888.
[9] D. Moxey, D. Ekelschot, Ü. Keskin, S. Sherwin, J. Peiró, High-order curvilinear meshing using a thermo-elastic analogy, Comput. Aided Des. 72 (2016) 130–139.
[10] M. Fortunato, P. Persson, High-order unstructured curved mesh generation using the Winslow equations, J. Comput. Phys. 307 (2016) 1–14.
[11] J. Eichstädt, M. Green, M. Turner, J. Peiró, D. Moxey, Accelerating high-order mesh optimisation with an architecture-independent programming model, Comput. Phys. Commun. 229 (2018) 36–53.
[12] S. Sherwin, J. Peiró, Mesh generation in curvilinear domains using high-order elements, Int. J. Numer. Methods Eng. 53 (1) (2002) 207–223.
[13] Z. Xie, R. Sevilla, O. Hassan, K. Morgan, The generation of arbitrary order curved meshes for 3D finite element analysis, Comput. Mech. 51 (2012) 361–374.
[14] R. Poya, R. Sevilla, A.J. Gil, A unified approach for a posteriori high-order curved mesh generation using solid mechanics, Comput. Mech. 58 (3) (2016) 457–490.
[15] R. Sevilla, L. Rees, O. Hassan, The generation of triangular meshes for NURBS-enhanced FEM, Int. J. Numer. Methods Eng. 108 (8) (2016) 941–968.
[16] T. Toulorge, C. Geuzaine, J.-F. Remacle, J. Lambrechts, Robust untangling of curvilinear meshes, J. Comput. Phys. 254 (2013) 8–26.
[17] M. Stees, S.M. Shontz, A high-order log barrier-based mesh generation and warping method, Proc. Eng. 203 (2017) 180–192.
[18] S. Karman, Curving for viscous meshes, in: Proceedings of the 27th International Meshing Roundtable, Springer, 2018, pp. 303–325.
[19] X. Luo, M.S. Shephard, J.-F. Remacle, The influence of geometric approximation on the accuracy of higher order methods, in: 8th Int. Conf. Numerical Grid Generation in Computational Field Simulations, 2002.
[20] D. Xue, L. Demkowicz, Control of geometry induced error in *hp* finite element (FE) simulations. I. Evaluation of FE error for curvilinear geometries, Int. J. Numer. Anal. Model. 2 (3) (2005) 283–300.
[21] P. Ciarlet, P.-A. Raviart, Interpolation theory over curved elements, with applications to finite element methods, Comput. Methods Appl. Mech. Eng. 1 (2) (1972) 217–249, https://doi.org/10.1016/0045-7825(72)90006-0.
[22] L. Botti, Influence of reference-to-physical frame mappings on approximation properties of discontinuous piecewise polynomial spaces, J. Sci. Comput. 52 (3) (2012) 675–703.
[23] B. Szabó, I. Babuška, Finite Element Analysis, John Wiley & Sons, New York, 1991.
[24] C. Schwab, p- and hp-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics, Clarendon Press, Oxford, 1998.
[25] M. Deville, P. Fischer, E. Mund, High-Order Methods for Incompressible Fluid Flow, vol. 9, Cambridge University Press, 2002.
[26] J. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, Texts in Applied Mathematics, Springer, 2007.
[27] G. Karniadakis, S. Sherwin, Spectral/hp Element Methods for Computational Fluid Dynamics, Oxford University Press, 2013.
[28] P. Fernandez, C. Nguyen, X. Roca, J. Peraire, Implicit large-eddy simulation of compressible flows using the interior embedded discontinuous Galerkin method, in: 54th AIAA Aerospace Sciences Meeting, 2016.
[29] P. Fernandez, N. Nguyen, J. Peraire, The hybridized discontinuous Galerkin method for implicit large-eddy simulation of transitional turbulent flows, J. Comput. Phys. 336 (2017) 308–329, https://doi.org/10.1016/j.jcp.2017.02.015.
[30] P.E. Vos, S. Sherwin, 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) (2010) 5161–5181.

[31] C. Cantwell, S. Sherwin, R. Kirby, P. Kelly, From h to p efficiently: strategy selection for operator evaluation on hexahedral and tetrahedral elements, Comput. Fluids 43 (1) (2011) 23–28.
[32] C. Cantwell, S. Sherwin, R. Kirby, P. Kelly, From h to p efficiently: selecting the optimal spectral/hp discretisation in three dimensions, Math. Model. Nat. Phenom. 6 (3) (2011) 84–96.
[33] R. Löhner, Error and work estimates for high-order elements, Int. J. Numer. Methods Fluids 67 (12) (2011) 2184–2188.
[34] M. Yano, An optimization framework for adaptive higher-order discretizations of partial differential equations on anisotropic simplex meshes, Ph.D. thesis, Massachusetts Institute of Technology, 2012.
[35] R. Kirby, S. Sherwin, B. Cockburn, To CG or to HDG: a comparative study, J. Sci. Comput. 51 (1) (2012) 183–212.
[36] A. Huerta, X. Roca, A. Angeloski, J. Peraire, Are high-order and hybridizable discontinuous Galerkin methods competitive?, Oberwolfach Rep. 9 (1) (2012) 485–487.
[37] R. Löhner, Improved error and work estimates for high-order elements, Int. J. Numer. Methods Fluids 72 (2013) 1207–1218.
[38] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, et al., High-order CFD methods: current status and perspective, Int. J. Numer. Methods Fluids 72 (8) (2013) 811–845.
[39] A. Huerta, A. Angeloski, X. Roca, J. Peraire, Efficiency of high-order elements for continuous and discontinuous Galerkin methods, Int. J. Numer. Methods Eng. 96 (2013) 529–560, https://doi.org/10.1002/nme.4547.
[40] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (2) (1997) 251–285.
[41] T.J. Barth, Simplified numerical methods for gas dynamics systems on triangulated domains, Ph.D. thesis, Stanford University, 1998.
[42] R. Sevilla, S. Fernández-Méndez, A. Huerta, NURBS–Enhanced Finite Element Method (NEFEM): a seamless bridge between CAD and FEM, Arch. Comput. Methods Eng. 18 (4) (2011) 441–484.
[43] H. Alt, M. Godau, Computing the Fréchet distance between two polygonal curves, Int. J. Comput. Geom. Appl. 5 (1) (1995) 75–91.
[44] J. Remacle, J. Lambrechts, C. Geuzaine, T. Toulorge, Optimizing the geometrical accuracy of 2D curvilinear meshes, Proc. Eng. 82 (2014) 228–239.
[45] T. Toulorge, J. Lambrechts, J. Remacle, Optimizing the geometrical accuracy of curvilinear meshes, J. Comput. Phys. 310 (2016) 361–380.
[46] E. Ruiz-Gironés, J. Sarrate, X. Roca, Defining an $\mathcal{L}_2$-disparity measure to check and improve the geometric accuracy of non-interpolating curved high-order meshes, Proc. Eng. 124 (2015) 122–134.
[47] E. Ruiz-Gironés, J. Sarrate, X. Roca, Generation of curved high-order meshes with optimal quality and geometric accuracy, Proc. Eng. 163 (2016) 315–327.
[48] P. Cignoni, C. Rocchini, R. Scopigno, Metro: measuring error on simplified surfaces, in: Computer Graphics Forum, vol. 17, Wiley Online Library, 1998, pp. 167–174.
[49] E.W. Chambers, E.C. De Verdiere, J. Erickson, S. Lazard, F. Lazarus, S. Thite, Homotopic Fréchet distance between curves or, walking your dog in the woods in polynomial time, Comput. Geom. 43 (3) (2010) 295–311.
[50] H. Alt, M. Buchin, Can we compute the similarity between surfaces?, Discrete Comput. Geom. 43 (1) (2010) 78–99.
[51] O. CASCADE, Open CASCADE technology, 3D modeling and numerical simulation, www.opencascade.org, 2012.
[52] A. Gargallo-Peiró, X. Roca, J. Peraire, J. Sarrate, A distortion measure to validate and generate curved high-order meshes on cad surfaces with independence of parameterization, Int. J. Numer. Methods Eng. 106 (13) (2016) 1100–1130, https://doi.org/10.1002/nme.5162.
[53] E. Ruiz-Gironés, X. Roca, J. Sarrate, High-order mesh curving by distortion minimization with boundary nodes free to slide on a 3D CAD representation, Comput. Aided Des. 72 (2016) 52–64.
[54] J. Nocedal, S. Wright, Numerical Optimization, Springer Verlag, 1999.
[55] Anaconda software distribution, https://www.anaconda.com/, 2020.
[56] swig, Simplified wrapper and interface generator, https://swig.org, 2018.
[57] X. Roca, E. Ruiz-Gironés, J. Sarrate, EZ4U: mesh generation environment, www.lacan.upc.edu/ez4u.htm, 2010.
[58] X. Roca, Paving the path towards automatic hexahedral mesh generation, Ph.D. thesis, Universitat Politècnica de Catalunya, 2009.
[59] P. Virtanen, R. Gommers, T.E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, Nat. Methods 17 (3) (2020) 261–272.
[60] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with NumPy, Nature 585 (7825) (2020) 357–362, https://doi.org/10.1038/s41586-020-2649-2.
[61] K. Höllig, J. Koch, Geometric Hermite interpolation, Comput. Aided Geom. Des. 12 (6) (1995) 567–580, https://doi.org/10.1016/0167-8396(94)00034-P.
[62] Pointwise Inc., Mesh generation software for CFD | Pointwise, Inc., http://www.pointwise.com, 2018.
[63] Geode, Project geode: geometry for simulation, http://www.pointwise.com/geode/, 2018.
[64] R. Feuillet, A. Loseille, F. Alauzet, Optimization of p2 meshes and applications, Comput. Aided Des. 124 (2020) 102846.
[65] M.J. Zahr, A. Shi, P.-O. Persson, Implicit shock tracking using an optimization-based high-order discontinuous Galerkin method, J. Comput. Phys. 410 (2020) 109385.