

## Parametric finite elements with bijective mappings

Patrick Zulian<sup>1</sup>  · Teseo Schneider<sup>1</sup> ·  
Kai Hormann<sup>1</sup> · Rolf Krause<sup>1</sup>

Received: 27 June 2016 / Accepted: 26 June 2017  
© Springer Science+Business Media B.V. 2017

**Abstract** The discretization of the computational domain plays a central role in the finite element method. In the standard discretization the domain is triangulated with a mesh and its boundary is approximated by a polygon. The boundary approximation induces a geometry-related error which influences the accuracy of the solution. To control this geometry-related error, iso-parametric finite elements and iso-geometric analysis allow for high order approximation of smooth boundary features. We present an alternative approach which combines parametric finite elements with smooth bijective mappings leaving the choice of approximation spaces free. Our approach allows to represent arbitrarily complex geometries on coarse meshes with curved edges, regardless of the domain boundary complexity. The main idea is to use a bijective mapping for automatically warping the volume of a simple parameterization domain to the complex computational domain, thus creating a curved mesh of the latter. Numerical examples provide evidence that our method has lower approximation error for domains with complex shapes than the standard finite element method, because we are able to solve the problem directly on the exact domain without having to approximate it.

---

✉ Patrick Zulian  
patrick.zulian@usi.ch

Teseo Schneider  
teseo.schneider@usi.ch

Kai Hormann  
kai.hormann@usi.ch

Rolf Krause  
rolf.krause@usi.ch

<sup>1</sup> Università della Svizzera italiana, Via Giuseppe Buffi 13, 6900 Lugano, Switzerland

**Keywords** Finite element method · Discretization · Barycentric coordinates · Bijective mapping · Super parametric · Parametrization

**Mathematics Subject Classification** 35-XX · 51-XX · 65N30 · 74-XX

## 1 Introduction

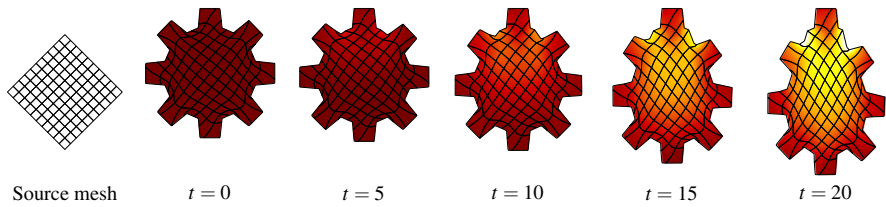
Simulating physical phenomena often requires to deal with complex geometric objects, generated, for instance, by computer aided design (CAD) software or captured from real life objects or organisms (e.g., 3D scans, MRI, etc.). When focusing on the finite element method (FEM), such highly complex geometries need to be represented in a sufficiently accurate way. This is the case because the accuracy of the geometric representation influences the approximation error of the discrete solution of a partial differential equation. In this paper we present a novel discretization combining FEM with bijective mappings. This combination allows solving the model problem on the exact geometry with the freedom of choosing the discretization independently.

The influence of the accuracy of the geometric representation on the approximation error has been studied for curved boundaries for iso-parametric discretizations [11, 38, 39] and for contact problems [24]. More recent research focuses on numerical studies for elliptic and Maxwell problems [43], and for different approximation spaces [4, 5]. In contrast, our approach always reproduces the exact geometry, hence has zero geometric error. Moreover, it can be applied to the numerical simulation of different physical phenomena.

During a simulation the approximation space might not be descriptive enough to represent the solution. This problem is usually solved by means of adaptive refinement strategies, such as  $h$ -refinement [6, 8] and  $p$ -refinement [33]. When using such strategies, a higher resolution at the boundary should be accompanied by a better approximation of the original surface [13]. However, due to the traditional one-way connection between geometric information and simulation environment, adaptive refinement is rarely accompanied by decrease of the geometric error, that is a better approximation of the underlying geometry. In other words, the mesh is generated within a modelling software and used in simulation environment without considering the original surface, preventing a better surface approximation.

Iso-geometric analysis (IGA) [20] allows to overcome this limitation by working directly with the geometric description provided by CAD software, such as non-uniform rational B-splines (NURBS). However, IGA is subject to several challenges such as the treatment of non-watertight surfaces, local refinement and topological flexibility [29]. Moreover, IGA approximations, similarly to many mesh-free methods, lead to complications in the imposition of essential boundary conditions, which can be either imposed in a weak sense [3], or least-square satisfied in the strong sense [20].

Additionally, when dealing with three-dimensional shapes, CAD models usually describe only the boundary. Creating a NURBS volume parameterization is a complex task, for which many different approaches exist. For instance, some of them require particular shapes [1], need special geometric information [31], or do not reproduce the surface exactly [28].



**Fig. 1** Transient non-linear elasticity simulation of a gear for a warped quadrilateral mesh with compressible-neo-Hookean material. The elastic gear is subject to vertical body forces (gravity) and has a fixed tooth on the top boundary. The *colour* represents the von Mises stress for the solution at the different time-steps  $t$  (color figure online)

An alternative to IGA is the NURBS-enhanced finite element method (NEFEM) [40] that allows exploiting CAD geometries for the exact description of the boundary. However, this method requires creating a parameterization mesh, and a special handling of the boundary, which according to [40] is still an open problem.

Another challenge regards geometric multigrid methods, which are based on a hierarchy of nested spaces for achieving optimal convergence [10, 18]. Such requirement can be satisfied by generating the hierarchy by refining a coarse mesh representation of the shape. However, such refinement usually does not improve the shape accuracy. An alternative approach [12] is to employ a parameterization such as transfinite interpolation [34, 35] and to build nested hierarchies in the parameterization domain. However, transfinite interpolation requires a surface parameterization, a specific parameterization domain, and it is not guaranteed to be bijective for general polytopes.

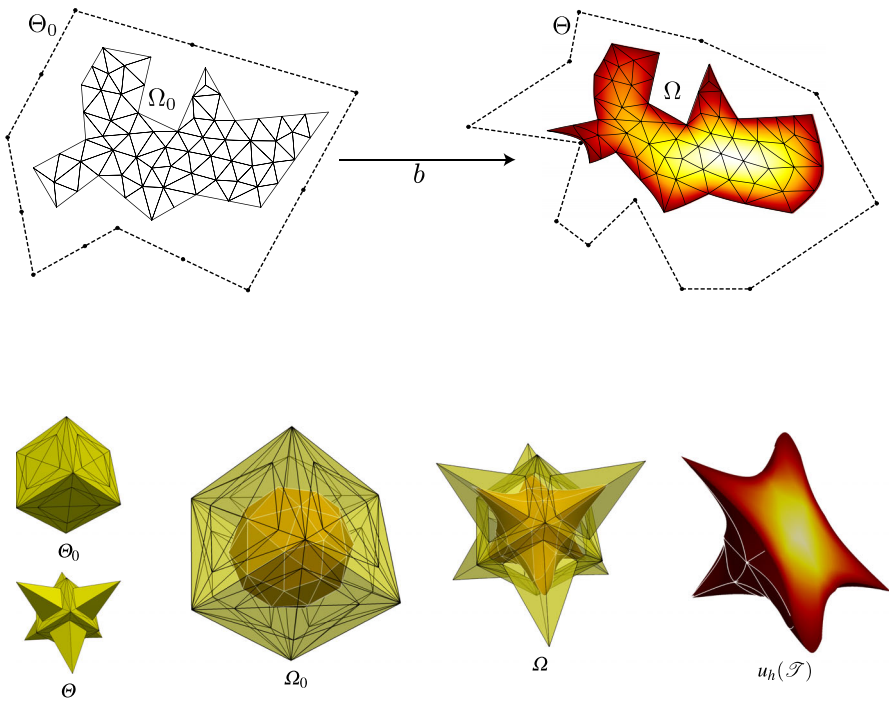
Here, we present a novel discretization which enables exploiting exact geometric descriptions (e.g., splines or surface meshes) together with strategies employed in standard finite element simulations (Sect. 2). This discretization has the advantage of decoupling the geometry and the approximation space, thus allowing for sub/iso/super-parametric elements. Although the presentation is based on the Poisson problem, our discretization can be naturally employed to solve more complex problems, such as transient non-linear elasticity shown in Fig. 1.

The problem of dealing with exact geometries has been deeply studied for CAD geometries by the IGA community. Unfortunately, a similar study for surface meshes is missing. For this reason, we focus on the *exact representation* provided by surface meshes, and present the construction of a bijective volume parameterization from arbitrarily shaped domains to arbitrarily shaped meshes (Sect. 3). Finally, we empirically illustrate that our new discretization can be used to remove geometric error while being comparable to the classical finite element method in terms of conditioning of the stiffness matrix and convergence of the solution (Sect. 4).

## 2 Parametric finite elements with bijective mappings

In this section we introduce the notation and derive the weak formulation for the Poisson problem including the change of metric needed by our method. Let us consider the standard Poisson problem

$$-\Delta u = f, \quad u|_{\partial\Omega} = g, \quad (1)$$



**Fig. 2** Overview of parametric finite elements with bijective mappings, with *colour-coded* solution of the Poisson problem (1) on a warped domain  $\Omega$ , with zero boundary conditions and constant right-hand side (color figure online)

where  $\Omega \in \mathbb{R}^d$  is the *computational domain*,  $\partial\Omega$  is the boundary of  $\Omega$ , and  $g$  describes the boundary values. In contrast to the classical construction where  $\Omega$  is defined by means of a mesh, here

$$\Omega = b(\Omega_0) \subseteq \Theta$$

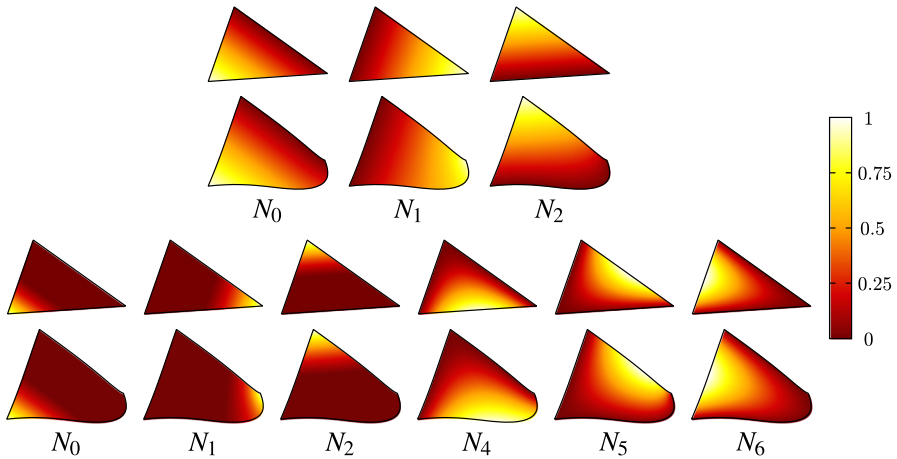
is given as the image of a sufficiently smooth *bijective mapping*

$$b: \Theta_0 \rightarrow \Theta,$$

where  $\Omega_0 \subseteq \Theta_0$  is the *source domain*,  $\Theta_0 \subset \mathbb{R}^d$  is the *parameterization domain*, and  $\Theta \subset \mathbb{R}^d$  is the *parameterization image*. Note that  $\Omega$  and  $\Theta$  are different mathematical objects:  $\Omega$  is a domain, whereas  $\Theta$  is a polytope. Figure 2 shows an overview of our construction and the solution of the Poisson problem (1).

Let  $u \in V = H_0^1(\Omega)$  be the Sobolev space of weakly differentiable functions vanishing on the boundary,  $f \in L^2(\Omega)$ , and  $g \in H^{1/2}(\partial\Omega)$ . Using integration by parts, we rewrite (1) in its weak form, which is: find  $u \in V$  such that

$$\int_{\Omega} \nabla u \cdot \nabla v = \int_{\Omega} f v \quad \forall v \in V.$$



**Fig. 3** The standard linear and quadratic shape functions  $N_i$  on the element of the *source mesh* and the corresponding warped element

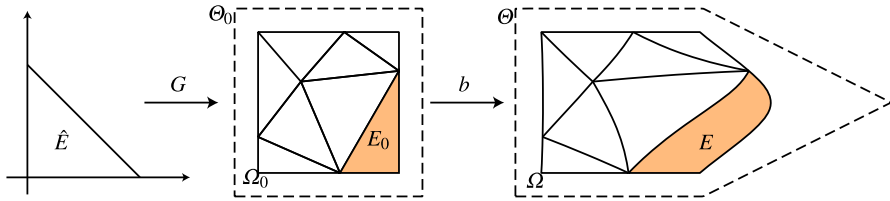
Using  $b$ , we express the previous integral with respect to the *source domain*  $\Omega_0$ . Considering that  $u(\mathbf{x}) = u(b(\mathbf{x}_0))$  and  $v(\mathbf{x}) = v(b(\mathbf{x}_0))$  where  $\mathbf{x} \in \Omega$  and  $\mathbf{x}_0 = b^{-1}(\mathbf{x}) \in \Omega_0$ , and applying change of variables in the integrals, we rewrite the weak form: find  $u \in V$  such that

$$\int_{\Omega_0} J_b^{-T} \nabla u \cdot J_b^{-T} \nabla v \det(J_b) = \int_{\Omega_0} f v \det(J_b) \quad \forall v \in V, \tag{2}$$

where  $J_b$  is the Jacobian matrix of the mapping  $b$ .

In order to solve this problem, we represent the computational domain  $\Omega$  by means of a warped mesh  $\mathcal{T} = b(\mathcal{T}_0)$ , where  $\mathcal{T}_0 = \{E_0 \subseteq \Omega_0 \mid \bigcup \bar{E}_0 = \bar{\Omega}_0\}$  is a conforming mesh (i.e., the intersection of two different elements is either empty, a common node, edge, or side), which describes the source domain  $\Omega_0$ . By  $\bar{\Omega}$  we denote the closure of  $\Omega$ . Note that, as described in (2), the bijective mapping warps the entire volume, creating warped elements  $E = b(E_0)$ . Let the finite element space associated to  $\mathcal{T}$  be  $V_h = V_h(\mathcal{T})$ , where  $h$  stands for the discretization parameter, and let  $\{N_1, \dots, N_m\}$  be a basis of  $V_h$ , where  $m$  is the number of basis functions. Figure 3 depicts an example of such basis functions for a warped element. We approximate the function  $u$  by means of  $u_h \in V_h$ , by expressing  $u_h$  in terms of its basis, obtaining  $u_h = \sum_{i=1}^m u_i N_i$ , where  $u_i$  are real coefficients. By choosing the test space as  $V_h$ , we discretize (2) as

$$\sum_{i=0}^m u_i \int_{\Omega} J_b^{-T} \nabla N_i \cdot J_b^{-T} \nabla N_j \det(J_b) = \sum_{i=0}^m f_i \int_{\Omega} N_i N_j \det(J_b) \quad \forall j = 1, \dots, m,$$



**Fig. 4** Overview of the geometric transformations from the reference element  $\hat{E}$  to the source element  $E_0 \in \mathcal{T}_0$  and to the warped element  $E \in \mathcal{T}$

which can be represented in the classical matrix form

$$\mathbf{L}\mathbf{u} = \mathbf{M}\mathbf{f}, \tag{3}$$

with  $\mathbf{u} = [u_1, \dots, u_m]^T$  and  $\mathbf{f} = [f_1, \dots, f_m]^T$

To assemble the stiffness matrix  $\mathbf{L}$  and the mass matrix  $\mathbf{M}$  we perform numerical quadrature. Because of the non-linearity of  $J_b$  we need to choose a quadrature scheme of possibly higher order (Sect. 3.3) even when the basis functions of the approximation space  $V_h$  are low order polynomials.

As usual, we introduce the reference element  $\hat{E}$  and the transformation  $G: \hat{E} \rightarrow E_0$  from the reference element to the corresponding element  $E_0$  in the source domain. We perform the quadrature in  $\hat{E}$ , using quadrature points  $\hat{\mathbf{x}}_k \in \hat{E}$ ,  $\mathbf{x}_k = G(\hat{\mathbf{x}}_k)$  with the respective *quadrature weights*  $\alpha_k \in \mathbb{R}$ ,  $k = 1, \dots, K$ . Figure 4 shows all the geometric transformations from the reference element  $\hat{E}$  to the warped element  $E$ . We denote by  $\hat{N}_i$  the basis functions on the reference element and by  $J_G$  the Jacobian of  $G$ . This allows assembling the local matrices for the element  $E$

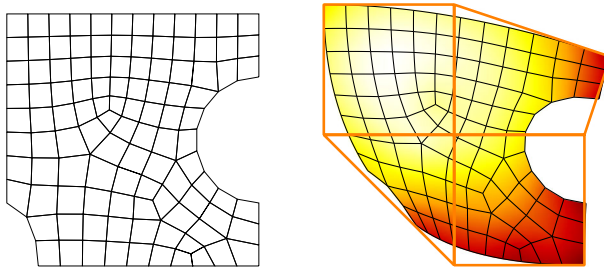
$$L_{i,j}^E = \sum_{k=1}^K \beta_k J^{-T}(\mathbf{x}_k) \nabla \hat{N}_i(\hat{\mathbf{x}}_k) \cdot J^{-T}(\mathbf{x}_k) \nabla \hat{N}_j(\hat{\mathbf{x}}_k),$$

$$M_{i,j}^E = \sum_{k=1}^K \beta_k \hat{N}_i(\hat{\mathbf{x}}_k) \hat{N}_j(\hat{\mathbf{x}}_k),$$
(4)

where  $J(\mathbf{x}_k) = J_b(\mathbf{x}_k)J_G(\hat{\mathbf{x}}_k)$  and  $\beta_k = \alpha_k \det(J(\mathbf{x}_k))|\hat{E}|$ , with  $|\hat{E}|$  the volume of  $\hat{E}$ . These local contributions are then gathered to compute the matrices  $\mathbf{L}$  and  $\mathbf{M}$ .

Note that the weak formulation and the assembly procedures are very similar to classical finite elements. In fact, the only difference is the usage of the geometric terms depending on the bijective mapping  $b$ , such as  $J_b$  which contributes to  $J = J_b J_G$ . As in standard FEM, the choice of the basis of  $V_h$  is independent from the geometric description, leading to super/sub/iso-parametric approximations. In our method the geometric description is given by the mapping  $b$ , which is usually non-linear, so that our discretization falls into the category of super-parametric elements.

If we assume that  $b(\mathcal{T}_0)$  describes the exact geometry, then the geometric error is zero. However, the error in the solution is also connected to the choice of the approximation space and the shape of the elements. This error is influenced by the



**Fig. 5** Example of parametric finite elements using a B-spline as the parameterization  $b$ . The colour describes the solution of the Poisson problem (1) (color figure online)

Jacobian  $J_b$  of the bijective mapping. We estimate it by means of the condition number

$$\kappa = \sup_{x_0 \in \Omega_0} \|J_b(x_0)\| \|J_b^{-1}(b(x_0))\| \quad (5)$$

as in standard parametric finite elements estimates [7,9].

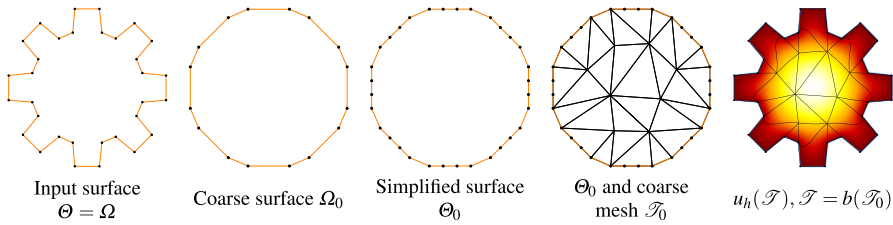
### 3 Shape and volume parameterization

The quality of a numerical solution of a partial differential equation is influenced by the accuracy of the geometric description and by the choice of the approximation space. In other words, a parameterization which describes the geometry exactly does not introduce any error related to the shape. The choice of this parameterization depends on the input geometry and includes every smooth bijective mapping, such as bijective spline mappings [15] (see Fig. 5), composite mean value mappings [37], or harmonic mappings [36].

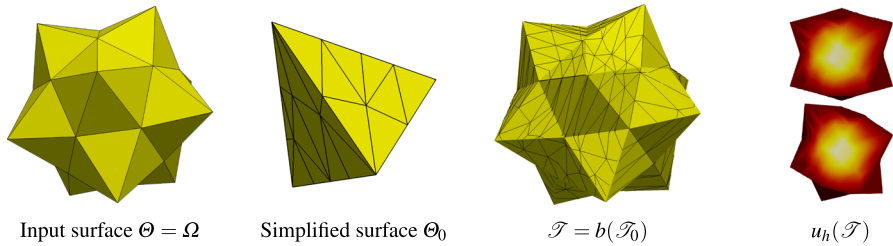
Since for CAD geometries the problem has been widely studied by the IGA community, we focus our study on volume parameterization between arbitrary surface meshes. The first challenge is the construction of a simpler surface  $\Theta_0$ , a coarse *source domain*  $\Omega_0$ , and a *parameterization image*  $\Theta$ , such that  $\Omega = b(\Omega_0)$  (Sect. 3.1). The other challenges are the construction of the volume parameterization  $b$  (Sect. 3.2), an accurate quadrature procedure (Sect. 3.3), and the efficient evaluation of the forms within a simulation work-flow (Sect. 3.4).

#### 3.1 Constructing the parameterization domain

In order to solve the model problem with the exact input geometry, the shape of  $\Theta$  must coincide with  $\Omega$ , which describes the exact shape. As carried out in detail in Sect. 2, our approach still requires a *parameterization domain*  $\Theta_0$  and a *source domain*  $\Omega_0$ . Hence, we first need to construct  $\Theta_0$  with the same mesh connectivity as  $\Theta$  while ensuring that  $\Theta_0$  describes a simpler shape. Note that in order to reproduce  $\Omega$  by



**Fig. 6** Given an input surface  $\Theta$  we simplify it to obtain  $\Theta_0$ , which coincides with  $\Omega_0$ . We mesh  $\Omega_0$  obtaining  $\mathcal{T}_0$  and solve the problem with respect to  $\mathcal{T}$ , which has the same boundary as  $\Theta$



**Fig. 7** Three-dimensional example of the work-flow of our approach, from the input surface to the solution of the problem in to the warped mesh  $\mathcal{T}$

means of  $b$  the shapes of  $\Omega_0$  and  $\Theta_0$  must also coincide, even if two meshes have a different number of faces and vertices.

The approximation space for the finite element solution for the model problem can now be chosen independently from the shape, since  $\Omega_0$  and  $\Theta_0$  are arbitrary (e.g., the octagon in Fig. 6 or the tetrahedron in Fig. 7). This allows meshing  $\Omega_0$  with arbitrary mesh size to obtain  $\mathcal{T}_0$ . Hence, by applying  $b$  to  $\mathcal{T}_0$ , we control the resolution of  $\mathcal{T} = b(\mathcal{T}_0)$  independently from the shape of  $\Theta$  without influencing the shape accuracy.

As illustrated in Fig. 6, in the 2D case,  $\Omega_0$  is constructed by removing vertices from  $\Theta$ . In order to obtain  $\Theta_0$  we reintroduce the removed vertices on the edges of  $\Omega_0$ , without modifying the shape described by  $\Omega_0$ . Finally, we mesh  $\Omega_0$  to obtain  $\mathcal{T}_0$  and solve the problem in  $\mathcal{T} = b(\mathcal{T}_0)$ .

The 3D case requires to coarsen  $\Theta$  in order to obtain  $\Omega_0$  while constructing a surface parameterization to build  $\Theta_0$  [14,25,27]. In our implementation we use the multi-resolution adaptive parameterization of surfaces (MAPS) algorithm [27], which produces a geometrically non-conforming parameterization (i.e.,  $\Theta_0$  is not nested inside  $\Omega_0$ ). To overcome this limitation, we extend the MAPS algorithm by snapping the vertices of  $\Theta_0$  to the edges of  $\Omega_0$ , and by applying few element splits to  $\Theta_0$  and  $\Theta$  when that is not feasible. We remark that the only operation performed on  $\Theta$  is splitting, which does not change its shape.

Summing up, we start with a high resolution mesh representing the exact geometry  $\Theta$ . Then, from  $\Theta$  we compute a coarse surface  $\Omega_0$  which we mesh to obtain  $\mathcal{T}_0$ . Finally, we use the parameterization obtained with MAPS to construct a surface  $\Theta_0$  with the same connectivity as  $\Theta$  and the same shape as  $\Omega_0$ . An example of a result of this procedure is shown in Fig. 7.



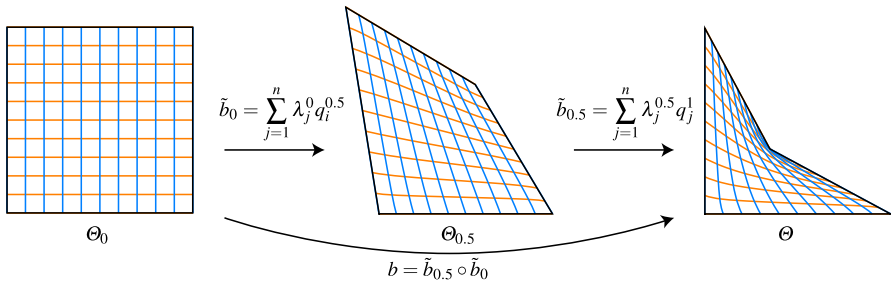


Fig. 8 Overview a composite barycentric mapping for  $\tau = [0, 0.5, 1]$

### 3.2 Composite mean value mappings

To the best of our knowledge only the *composite mean value mappings* [37] allow creating smooth-bijective mappings between polytopes, such as polygons or polygonal meshes. Convenient properties of such mappings are that they can be evaluated pointwise, are provided in closed form, and are  $C^\infty$  in the interior of the domain. These mappings are based on the *mean value mapping*

$$\tilde{b}(x) = \sum_{j=1}^n \lambda_j(x) \mathbf{q}_j,$$

where  $\mathbf{q}_j$  are the vertices of  $\Theta$  and the functions  $\lambda_j : \Theta_0 \rightarrow \mathbb{R}, j = 1, \dots, n$  are a set of *mean value coordinates* [16, 22] with respect to  $\Theta_0$ . That is,

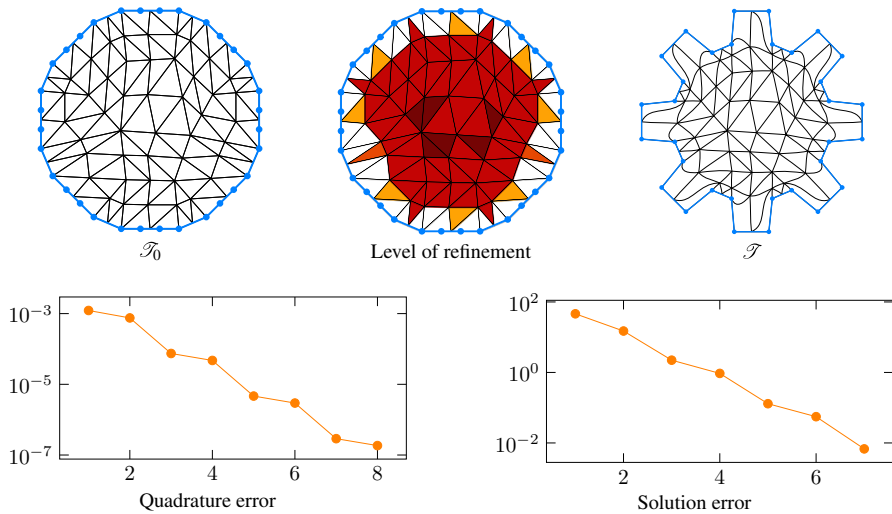
$$\lambda_j = \frac{w_j}{\sum_{k=1}^n w_k} \quad \text{with} \quad w_j = \frac{\tan(\alpha_{j-1}/2) + \tan(\alpha_j/2)}{r_j},$$

where  $\alpha_j$  is the angle between the edges  $[x, \mathbf{q}_{j+1}^0]$  and  $[x, \mathbf{q}_j^0]$  and  $r_j = \|x - \mathbf{q}_j^0\|$ , with  $\mathbf{q}_j^0$  the vertices of  $\Theta_0$ .

Unfortunately, the mapping  $\tilde{b}$  is not guaranteed to be bijective for all pair of polytopes [21]. To overcome this limitation we follow [37] and “split” the mapping from source to target polytope into a finite number of steps, where each step perturbs the vertices only slightly. To this end, suppose that  $\zeta_i : [0, 1] \rightarrow \mathbb{R}^2, i = 1, \dots, n$  are a set of continuous *vertex paths* between each vertex  $\mathbf{q}_i^0 = \zeta_i(0)$  and its corresponding vertex  $\mathbf{q}_i = \zeta_i(1)$ .

Let  $\tau_s = (t_0, t_1, \dots, t_s)$  with  $t_k = k/s$  for  $k = 0, \dots, s$  be a *uniform partitioning* of  $[0, 1]$  and  $\tilde{b}_k$  be the barycentric mapping from  $\Theta_{t_k}$  to  $\Theta_{t_{k+1}}$ , based on the barycentric coordinates  $\lambda_i^{t_k} : \Theta_{t_k} \rightarrow \mathbb{R}$ . Then we define the *composite barycentric mapping* from  $\Theta_0$  to  $\Theta$  as

$$b = \tilde{b}_{s-1} \circ \tilde{b}_{s-2} \circ \dots \circ \tilde{b}_0,$$



**Fig. 9** Quadrature error and error in the solution  $u_h \in V(\mathcal{T})$  for different levels of quadrature refinements. The level of refinement is higher (white) where the distortion of  $b$  is large (in proximity of the boundary), while no refinement is necessary in the interior (dark red color) (color figure online)

which is bijective for sufficiently large  $s$  according to [37]. Figure 8 shows an example of how a composite barycentric mapping is constructed for  $s = 2$ .

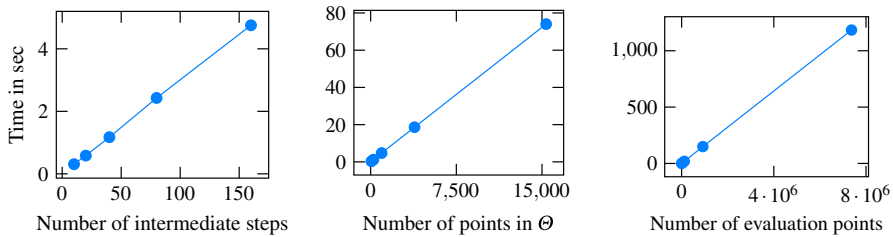
### 3.3 Quadrature

As previously mentioned, the non-linearity of the bijective mapping  $b$  may introduce high error when low order numerical quadrature is employed. We implement an adaptive Simpson scheme [32] for accurately computing the stiffness matrix and right-hand side vector. For each element  $E$ , we compute the element mass matrix  $M^E$  by means of a Gaussian quadrature formula. We subdivide  $E$ , and we use the same quadrature formula on each sub-element to re-compute  $M^E$  more accurately. We compute the difference of these two results and continue to subdivide until a given numerical tolerance is reached.

Figure 9 shows the convergence behaviour of the adaptive quadrature strategy with respect to the number of maximum refinements for reaching a quadrature tolerance of  $10^{-8}$ . The same figure shows how this error propagates to the solution for a low resolution mesh. It is interesting to note that the scheme automatically adapts to  $b$ , creating more levels where the mapping is less linear.

### 3.4 Pre-computation of the composite mean value mapping

The evaluation of composite mean value mapping described in Sect. 3.2 is computationally intensive. For this reason we need to avoid computing the mapping and its Jacobian multiple times. Similar to the classical assembly procedure of the sys-



**Fig. 10** Running times for computing the composite mean value mapping and its Jacobian. The computational time depends on three parameters: the number of intermediate steps  $s$ , the vertices  $n$  of  $\Theta$ , and the evaluation points. For each of the three experiments we vary only one of the parameters, whose base values are  $s = 10$ ,  $n = 62$ , and 1800 evaluation points

tem matrices, we start by deciding the order of quadrature. The order of quadrature depends on the problem we want to solve, the choice of the approximation space, and, especially for our approach, the bijective mapping  $b$ .

Instead of directly assembling the system matrices in (3), we divide the assembly procedure into two stages. The first stage consists of generating and storing all the quadrature data associated with the geometry necessary for the assembly, such as the global quadrature points  $b(G(\hat{x}))$  and the Jacobian matrices  $J_b(\hat{x})$ .

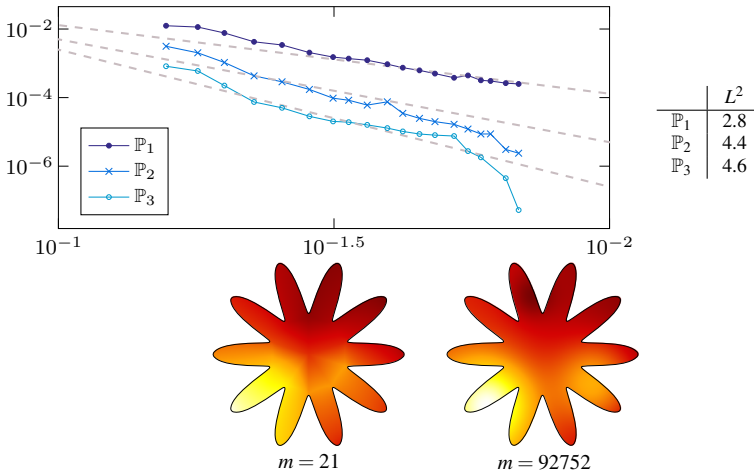
The second stage consists of the standard assembly procedure of the element matrices (4), now using the precomputed quadrature quantities. This strategy allows assembling the matrices like for standard finite elements without the need of evaluating  $b$  and  $J_b$  for each new operator.

For the standard finite element assembly procedure storing the quadrature data is usually not necessary, making our two stage approach less memory-efficient. However, the caching allows both a parallel evaluation of  $b$  and the possibility of reusing the quadrature data for different operators (e.g., Laplacian and mass matrix) and multiple time-steps (e.g., in case of transient simulations in non-linear elasticity). For instance, in Fig. 1 the quantities related to  $b$  are computed only at the first time-step and reused in the following ones.

Despite the pre-computation, the evaluation of  $b$  remains expensive. Fortunately, mean value coordinates are straightforward to parallelize on shared memory processors. In fact, every point-wise evaluation of  $b$  and  $J_b$  can be computed in a completely independent way. Figure 10 shows the parallel-running times using OpenCL [23] with respect to different input sizes, computed on a laptop computer with Intel Core i7 2.3GHz processor and 16GB RAM, we refer to [41] for a comprehensive explanation of how to compute the gradient of mean value coordinates.

## 4 Numerical experiments

We restrict our study to super-parametric discretizations (i.e., discretizations where the geometric mapping is more descriptive than the basis functions) based on composite mean value mappings (Sect. 3.2) with linear, quadratic, and cubic Lagrangian elements ( $\mathbb{P}_1$ ,  $\mathbb{P}_2$ , and  $\mathbb{P}_3$ ). For our experiments the analytical solution is unknown, hence we estimate it by computing a reference solution  $u \in V(\mathcal{I}_f)$  on a very fine mesh  $\mathcal{I}_f$ . To



**Fig. 11** Top left visualization of  $e(u_h)$  against the mesh size  $h$ , where the straight dashed lines show the quadratic, cubic, and quartic trends. Top right  $L^2$ -convergence rates for the different discretizations. Bottom solution of the Poisson problem, where  $u$  is the Franke function, for different number of nodes  $m$ . We use  $s = 10$  uniform steps in  $b$ . The experiment has the mesh set-up shown in Fig. 13

evaluate the quality of our discretization and the standard discretization, we compute different solutions  $u_h \in V_h(\mathcal{T})$  for several mesh sizes  $h$ .

### 4.1 Convergence

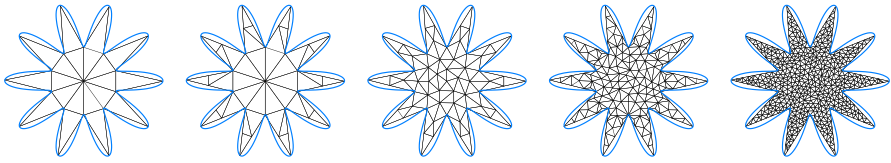
The solution is expected to converge quadratically in  $L^2(\Omega)$  to the exact one with respect to the mesh size  $h$  for classical FEM with linear elements for  $H^2$ -regular problems. Hence, we study the convergence related to our approach by measuring the approximation error as

$$e(u_h) = \|P(u_h) - u\|_{L^2(\mathcal{T}_f)},$$

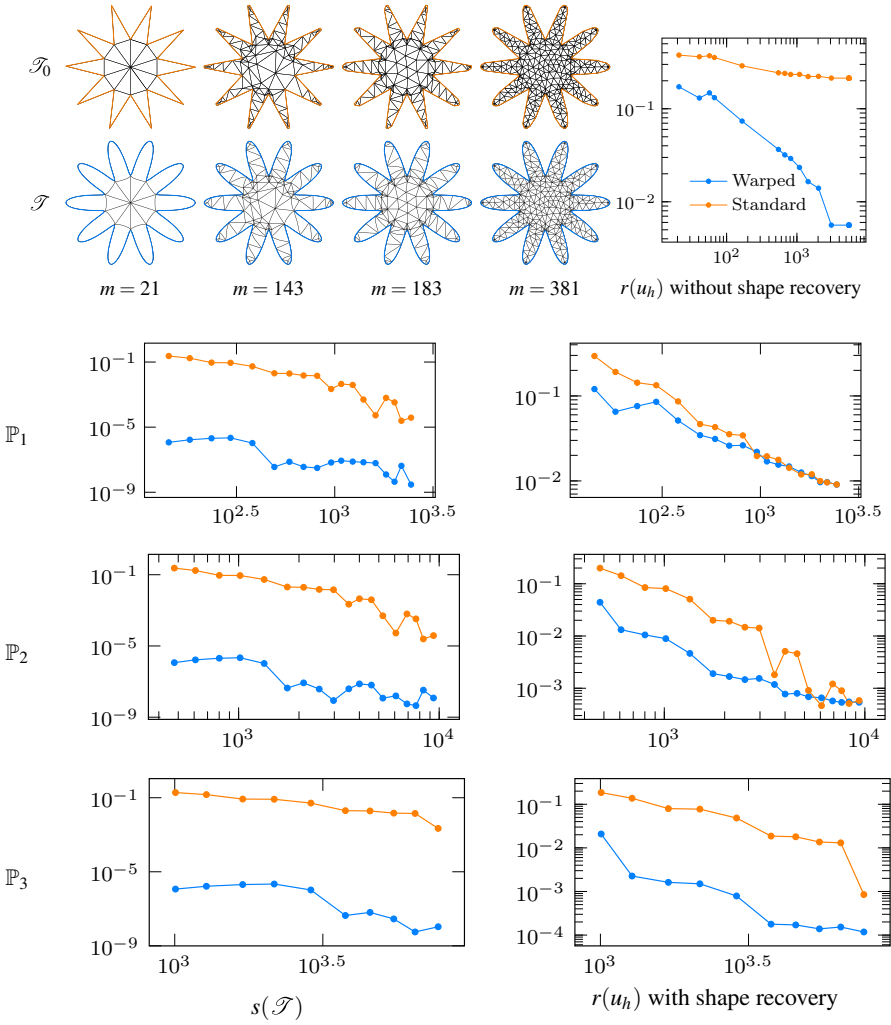
where  $P : V_h(\mathcal{T}) \rightarrow V(\mathcal{T}_f)$  is the  $L^2$ -projection operator [26, 42] (the assembly of  $P$  is performed in the parameterization domain). Similar to standard FEM, our method shows a quadratic, cubic, and quartic convergence behaviour for the Poisson problem, as illustrated in the plot in Fig. 11. Despite the fact that the computation is always performed on the exact geometry, the approximation error is not zero because of the piecewise polynomial approximation of the solution, which is visible for a mesh with small  $m$  and decreases for larger  $m$ .

### 4.2 Comparison

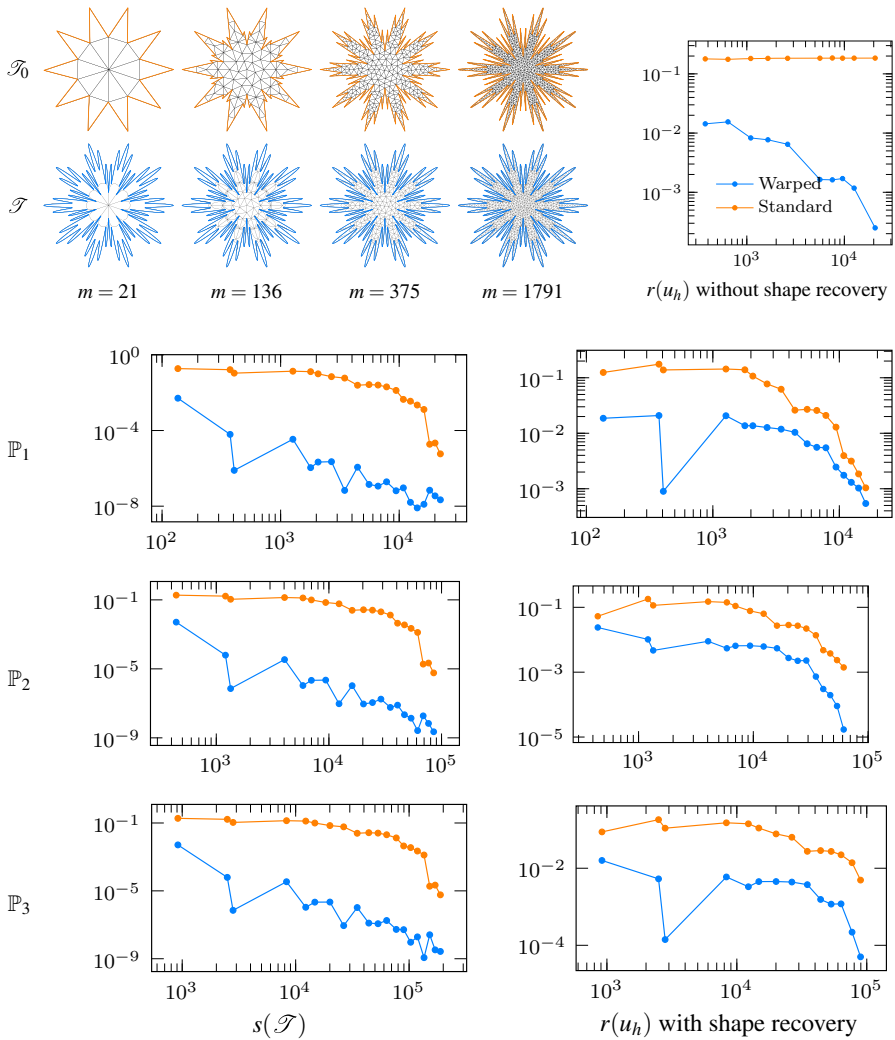
We compare our method with the standard finite element discretization for a simple 2D problem (Fig. 13), an extreme 2D problem (Fig. 14), and for a realistic 3D shape



**Fig. 12** Mesh refinement without shape recovery. Even at fine resolution (*last image*) we do not recover the original shape (*blue polygon*) (color figure online)



**Fig. 13** Source meshes  $\mathcal{T}_0$  with boundary  $\mathcal{O}_0$  (*first row*), warped meshes  $\mathcal{T}$  used by our method (*second row*) with  $s = 10$  uniform steps, and convergence plots against different numbers of degrees of freedom  $m$  (*last row*)

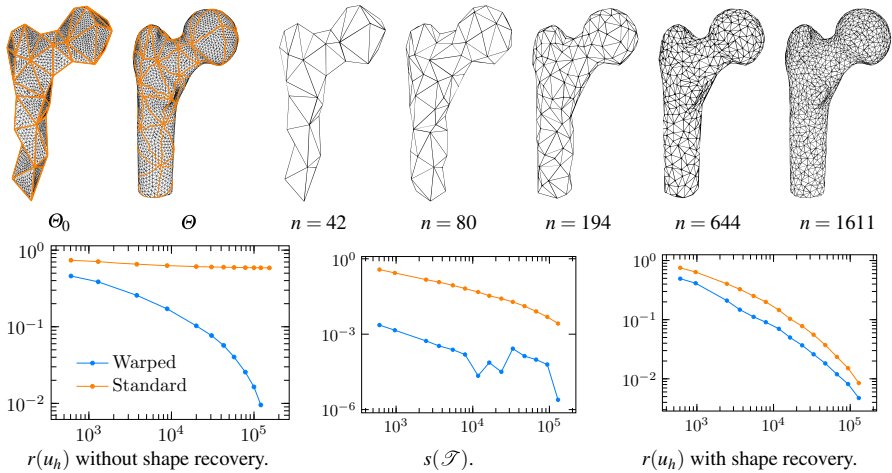


**Fig. 14** Source meshes  $\mathcal{T}_0$  with boundary  $\partial_0$  (first row), warped meshes  $\mathcal{T}$  used by our method (second row) with  $s = 50$  uniform steps, and convergence plots against different numbers of degrees of freedom  $m$  (last row)

(Fig. 15). Since for the standard finite element discretization, the boundary of  $\mathcal{T}$  differs from  $\Omega$ , we measure

$$r(u_h) = \left| \frac{\|u_h\|_{L^2(\mathcal{T})}}{\|u\|_{L^2(\mathcal{T}_f)}} - 1 \right|$$

to estimate the approximation error [30], where  $\|u\|_{L^2(\mathcal{T}_f)}$  is computed on a dense mesh with  $\mathbb{P}_1$  elements.



**Fig. 15** Convergence plots against different numbers of degrees of freedom  $m$  for a 3D experiment. We use  $s = 10$  uniform steps in the composite mapping integrated with a standard Gaussian quadrature formula

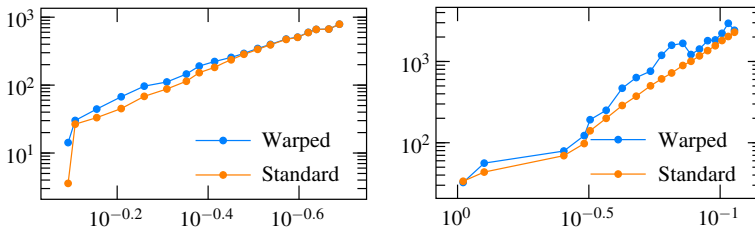
In classical finite element simulations the original shape is usually not recovered when performing mesh refinement as shown in Fig. 12. For this reason,  $r(u_h)$  does not converge to zero for the standard solution, while our approach converges (top plots in Figs. 13, 14, 15).

In order to better understand this behaviour, we measure the actual geometric deviation with

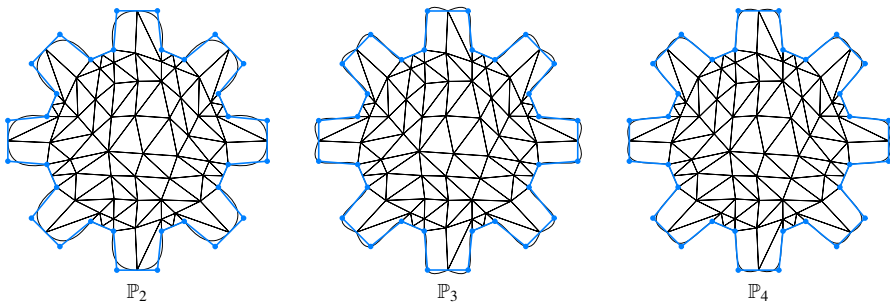
$$s(\mathcal{T}) = \left| \frac{\|1\|_{L^2(\mathcal{T})}}{\|1\|_{L^2(\Omega)}} - 1 \right|$$

which corresponds to the volume of the mesh (note that  $\|1\|_{L^2(\mathcal{T})}$  is equivalent to the square root of the sum of the entries of the mass-matrix). We compute the volume by means of numerical quadrature, which might introduce errors (Sect. 3.3), since our discretization consists of warped elements. For the standard discretization, when refining the mesh without recovering the shape, the volume trivially stays constant. Hence, in order to have a fair comparison, we increase the shape accuracy while refining the mesh to ensure that the shape of the domain also converges to the exact one. The behaviour of  $s(\mathcal{T})$  shows that our discretization has almost zero geometrical error independently of  $h$ , while the standard discretization has higher geometrical error (middle plots in Figs. 13, 14, 15).

In order to investigate how the approximation error is influenced by the geometrical error, we measure  $r(u_h)$  for our method and classical finite elements with shape recovery. Our discretization always has a smaller approximation error compared to the standard discretization (right plots in Figs. 13, 14, 15). This is due to the fact that our approach allows solving the problem in the exact geometry, even at low resolutions.



**Fig. 16** Condition number of the discrete Laplace operator  $\kappa(L)$  against the mesh size  $h$  for the examples in Fig. 13 (left) and Fig. 14 (right)



**Fig. 17** Polynomial boundary descriptions (*mesh*) compared to the exact boundary (*blue polygon*) (color figure online)

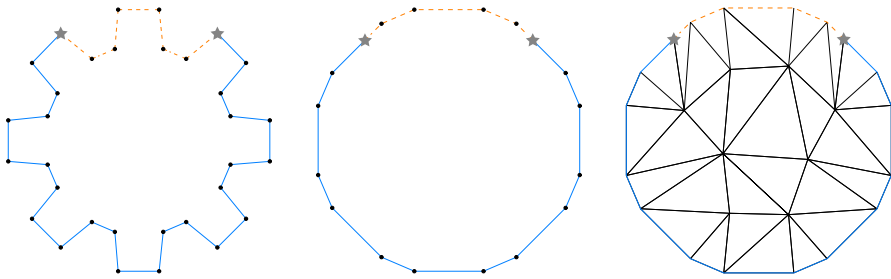
### 4.3 Conditioning

For solution methods such as iterative solvers, the condition number  $\kappa$  of the stiffness matrix plays an important role for the convergence rate [2]. In order to understand how our discretization affects the condition number, we compute  $\kappa$  for the discrete Laplace operator  $L$  with respect to different mesh sizes  $h$  for both our discretization and the standard one. Because of the influence of the bijective mapping  $b$ , as shown in (5), our discretization has a slightly larger condition number. Figure 16 shows that  $\kappa(L)$  behaves similarly for both discretizations which suggests that iterative solvers perform nearly as well for our discretization as for the standard one.

### 5 Conclusions

The idea of combining the finite element method with bijective mappings allows representing complex geometries on coarse meshes and enables specifying interpolation conditions as in the classical finite element method. For instance, our method can be used with Lagrange elements, splines, NURBS, or mixed FEM independently from the complexity of the input geometry. We introduce this novel discretization focusing on the particular case of composite mean value mappings which automatically creates a volume parameterization given only the boundary description.





**Fig. 18** Handling the interface (grey stars) between the Neumann boundary (blue solid lines) and the Dirichlet boundary (orange dashed lines) from  $\Theta$  to  $\mathcal{T}_0$  (color figure online)

An alternative is the use of iso-parametric finite elements, where the geometric mapping is of the same order as the basis functions. Therefore, the domain mesh is composed by piecewise polynomial elements, allowing a higher order description of the boundary shape. This approach describes smooth shapes well, but produces a sub-optimal approximation of non-smooth surfaces, which is the main focus of our discretization. Figure 17 shows how high-order polynomials fail in describing the boundary of a simple gear. Moreover, such approximations do not have any simple construction strategies that guarantee bijectivity (i.e., avoids negative volumes).

Through numerical experiments we show that our discretization has a lower approximation error compared to standard approaches, due to the higher geometric accuracy, without significant changes on the conditioning of the discrete operators. The method becomes computationally more expensive when employing the composite mean value mapping, however much of the related data can be precomputed and reused for different operators, as explained in Sect. 3.4. Moreover from the assembly point of view, our method only requires to change to quadrature procedure (4) by including the terms containing  $b$ .

Although we focus our study on the case of discrete geometry and composite mean value mappings, our construction might be suitable for any other choice of bijective mapping  $b$ , and it would be interesting to further investigate this flexibility. For instance, within the composite mapping, we can employ other types of smooth barycentric coordinates for which we can compute the Jacobian, such as maximum entropy coordinates [17, 19].

The integration of our approach with efficient and modern solution techniques, such as multigrid methods, is possible. In fact, the flexibility provided by arbitrarily choosing the mesh for describing  $\Omega_0$ , allows to naturally generate nested geometric multigrid hierarchies with exact geometry. Moreover, the construction of the interpolation and restriction operators is trivially performed using standard mesh refinement of the source mesh  $\mathcal{T}_0$ , since the mapping  $b$  is the same for all levels.

Our discretization with composite mean value mapping enables treating boundary conditions with arbitrary precision even for the non-homogeneous case. For instance, let us consider the example problem in Fig. 18, where Dirichlet boundary conditions are specified on  $\partial\Omega_D \subseteq \partial\Omega$  (orange dashed lines) and Neumann conditions on  $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$  (blue solid lines). Let the interface (grey stars) between  $\partial\Omega_D$  and  $\partial\Omega_N$  be

$\Gamma$  and its corresponding interface in  $\Theta_0$  be  $\Gamma_0$  (i.e.,  $\Gamma = b(\Gamma_0)$ ). When generating the mesh  $\mathcal{T}_0$  we preserve  $\Gamma_0$  which is then mapped to its image  $\Gamma$  in  $\mathcal{T}$ . Since  $\Gamma$  is preserved, the boundary conditions which are specified on  $\partial\Omega_D$  and  $\partial\Omega_N$  can be equivalently handled on  $\Omega_0$ .

In our presentation we always define the mapping  $b$  as a global parameterization from  $\Theta_0$  to  $\Theta$ , though, in order to have a faster computation of the quadrature data, strategies for localizing the mapping can be applied. For instance, the mapping can be used to exclusively compute the quadrature data associated with the elements near the boundary, while for the elements in the interior the mapping can be applied to the nodal positions only, generating a piecewise polynomial approximation, similarly to [40].

**Acknowledgements** This work was supported by the SNF under Project Number 200020\_156178, SCCER-SoE, and SCCER-FURIES. The method presented in this paper is implemented within the MOONLITH software library. We thank the anonymous reviewers for their useful comments which helped improving both content and presentation of this paper.

## References

1. Aigner, M., Heinrich, C., Jüttler, B., Pilgerstorfer, E., Simeon, B., Vuong, A.V.: Swept Volume Parameterization for Isogeometric Analysis. Springer, Berlin (2009)
2. Bathe, K.J., Wilson, E.L.: Numerical methods in finite element analysis. *AMC* **10**, 12 (1976)
3. Bazilevs, Y., Michler, C., Calo, V., Hughes, T.: Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Comput. Methods Appl. Mech. Eng.* **199**(13–16), 780–790 (2010)
4. Bertrand, F., Muñizmaier, S., Starke, G.: First-order system least squares on curved boundaries: higher-order Raviart–Thomas elements. *SIAM J. Numer. Anal.* **52**(6), 3165–3180 (2014)
5. Bertrand, F., Muñizmaier, S., Starke, G.: First-order system least squares on curved boundaries: lowest-order Raviart–Thomas elements. *SIAM J. Numer. Anal.* **52**(2), 880–894 (2014)
6. Bey, J.: Tetrahedral grid refinement. *Computing* **55**, 355–378 (1995)
7. Braess, D.: Finite Elements, 3rd edn. Cambridge University Press, Cambridge (2007)
8. Bramble, J.H., Pasciak, J.E., Steinbach, O.: On the stability of the  $L^2$ -projections in  $H^1$ . *Math. Comput.* **71**(237), 147–156 (2002)
9. Brenner, S., Scott, R.: The Mathematical Theory of Finite Element Methods, Texts in Applied Mathematics, vol. 15. Springer, New York (2008)
10. Briggs, W.L., Henson, V.E., McCormick, S.F.: A Multigrid Tutorial, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia (2000)
11. Ciarlet, P.G., Raviart, P.A.: Interpolation theory over curved elements, with applications to finite element methods. *Comput. Methods Appl. Mech. Eng.* **1**(2), 217–249 (1972)
12. Dickopf, T., Krause, R.: Monotone Multigrid Methods Based on Parametric Finite Elements. Submitted to Lecture Notes in Computational Science and Engineering. Technical Report, ICS, USI (2011)
13. Dörfler, W., Rumpf, M.: An adaptive strategy for elliptic problems including a posteriori controlled boundary approximation. *Math. Comput. Am. Math. Soc.* **67**(224), 1361–1382 (1998)
14. Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., Stuetzle, W.: Multiresolution analysis of arbitrary meshes. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95, pp. 173–182. ACM (1995)
15. Erikson, A.P., Åström, K.: On the bijectivity of thin-plate splines. In: Åström, K., Persson, L.E., Silvestrov, S.D. (eds.) Analysis for Science, Engineering and Beyond: The Tribute Workshop in Honour of Gunnar Sparr held in Lund, May 8–9, 2008, pp. 93–141. Springer, Berlin, Heidelberg (2012)
16. Floater, M.S.: Mean value coordinates. *Comput. Aided Geom. Des.* **20**(1), 19–27 (2003)
17. Greco, F., Sukumar, N.: Derivatives of maximum-entropy basis functions on the boundary: theory and computations. *Int. J. Numer. Methods Eng.* **94**(12), 1123–1149 (2013)

18. Hackbusch, W.: *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, vol. 4. Springer, Berlin (1985)
19. Hormann, K., Sukumar, N.: Maximum entropy coordinates for arbitrary polytopes. *Comput. Graph. Forum* **27**(5), 1513–1520 (2008)
20. Hughes, T.J., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.* **194**(39), 4135–4195 (2005)
21. Jacobson, A.: *Bijective Mappings with Generalized Barycentric Coordinates: A Counterexample*. Technical Report, Department of Computer Science, ETH Zurich (2012)
22. Ju, T., Schaefer, S., Warren, J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph.* **24**(3), 561–566 (2005)
23. Khronos OpenCL Working Group: *The OpenCL Specification*, version 1.0.29. <http://khronos.org/registry/cl/specs/opencl-1.0.29.pdf> (2008)
24. Kikuchi, N., Oden, J.T.: *Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods*, vol. 8. SIAM, Philadelphia (1988)
25. Krause, R., Sander, O.: Automatic construction of boundary parametrizations for geometric multigrid solvers. *Comput. Vis. Sci.* **9**(1), 11–22 (2006)
26. Krause, R., Zulian, P.: A parallel approach to the variational transfer of discrete fields between arbitrarily distributed finite element meshes. *SIAM J. Sci. Comput.* **38**(3), C307–C333 (2016)
27. Lee, A.W., Sweldens, W., Schröder, P., Cowsar, L., Dobkin, D.: Maps: multiresolution adaptive parameterization of surfaces. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 95–104. ACM (1998)
28. Li, B., Li, X., Wang, K., Qin, H.: Surface mesh to volumetric spline conversion with generalized polycubes. *IEEE Trans. Vis. Comput. Graph.* **19**(9), 1539–1551 (2013)
29. Lian, H., Bordas, S.P.A., Sevilla, R., Simpson, R.N.: *Recent Developments in CAD/Analysis Integration*. ArXiv e-prints (2012)
30. Luo, X., Shephard, M.S., Remacle, J.F.: The influence of geometric approximation on the accuracy of high order methods. *Rensselaer SCOREC Report*, vol. 1 (2001)
31. Martin, T., Cohen, E.: Volumetric parameterization of complex objects by respecting multiple materials. *Comput. Graph.* **34**(3), 187–197 (2010)
32. McKeeman, W.M.: Algorithm 145: adaptive numerical integration by Simpson’s rule. *Commun. ACM* **5**(12), 604–608 (1962)
33. Melenk, J., Wohlmuth, B.: On residual-based a posteriori error estimation in hp-fem. *Adv. Comput. Math.* **15**(1), 311–331 (2001)
34. Randrianarivony, M.: *Tetrahedral Transfinite Interpolation with B-patch Faces: Construction and Regularity*, vol. 803. INS Preprint (2008)
35. Randrianarivony, M.: On transfinite interpolations with respect to convex domains. *Comput. Aided Geom. Des.* **28**(2), 135–149 (2011)
36. Schneider, T., Hormann, K.: Smooth bijective maps between arbitrary planar polygons. *Comput. Aided Geom. Des.* **35–36**(c), 243–354 (2015)
37. Schneider, T., Hormann, K., Floater, M.S.: Bijective composite mean value mappings. *Comput. Graph. Forum* **32**(5), 137–146 (2013)
38. Scott, L.R.: *Finite element techniques for curved boundaries*. Ph.D. Thesis, Massachusetts Institute of Technology (1973)
39. Scott, R.: Interpolated boundary conditions in the finite element method. *SIAM J. Numer. Anal.* **12**(3), 404–427 (1975)
40. Sevilla, R., Fernández-Méndez, S., Huerta, A.: Nurbs-enhanced finite element method (nefem). *Arch. Comput. Methods Eng.* **18**(4), 441–484 (2011)
41. Thiery, J.M., Tierny, J., Boubekur, T.: Jacobians and Hessians of mean value coordinates for closed triangular meshes. *Vis. Comput.* **30**(9), 981–995 (2014)
42. Wohlmuth, B.I.: A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM J. Numer. Anal.* **38**, 989–1012 (1998)
43. Xue, D., Demkowicz, L., et al.: 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), 283–300 (2005)