

Elliptic grid generation techniques in the framework of isogeometric analysis applications



J. Hinz, M. Möller, C. Vuik

Department of Applied Mathematics (DIAM), Delft University of Technology, The Netherlands

ARTICLE INFO

Article history:

Available online 27 March 2018

ABSTRACT

The generation of an analysis-suitable computational grid from a description of no more than its boundaries is a common problem in numerical analysis. Most classical meshing techniques for finite-volume, finite-difference or finite-element applications such as the *Advancing Front Method* (Schöberl, 1997), *Delaunay Triangulation* (Triangle, 1996) and elliptic or hyperbolic meshing schemes (Thompson et al., 1998) operate with linear or multi-linear but straight-sided elements for the generation of structured and unstructured meshes, respectively, whereas the generation of high-quality curved meshes is still considered a major challenge. A recent development is the introduction of *Isogeometric Analysis* (IgA) (Hughes et al., 2005), which can be considered as a natural high-order generalisation of the finite-element method. A description of the geometry $\tilde{\Omega}$ is accomplished via a mapping operator $\mathbf{x} : \tilde{\Omega} \rightarrow \Omega$ that maps the unit hypercube in \mathbb{R}^n onto an approximation Ω of $\tilde{\Omega}$ utilizing a linear combination of higher-order spline functions. The numerical simulation is then carried out in the computational domain $\tilde{\Omega}$ via a ‘pull back’ using the mapping operator \mathbf{x} . The advantage is that the flexibility of higher-order spline-functions usually allows for an accurate description of $\tilde{\Omega}$ with much fewer elements which can significantly reduce the computational effort required for this step compared to traditional low-order methods. Furthermore, an analytical description of the geometry can be turned back into a traditional (structured or unstructured) grid by performing a large number of function evaluations in \mathbf{x} . This can, for instance, be utilized for local refinement without the need for remeshing.

A potential drawback of curved instead of linear elements is that the meshing techniques required for the creation of folding-free mappings tend to be more sophisticated and that it is a less trivial task to verify that the resulting mapping is indeed bijective.

For the purpose of creating folding-free mappings utilizing spline-functions, we will present an algorithm adopting the principles of *Elliptic Grid Generation* (EGG) whose basic principles have been adapted to meet the needs of IgA. In \mathbb{R}^2 , EGG has particularly appealing properties since bijectivity of the resulting mapping is guaranteed as long as the numerical accuracy to the computational approach of the mapping is sufficient. We will present an algorithm that is capable of generating folding-free mappings from a large number of geometry contours, including complicated geometries from industrial applications with extreme aspect ratios. This is accomplished by combining EGG with automatized reparameterization techniques and a sophisticated numerical approach for solving resulting governing (nonlinear) equations. The algorithm is equipped with the means to verify the bijectivity of the resulting mapping and with automatized defect-correction methods in case a violation of bijectivity is detected.

Furthermore, we will present possible strategies for the generation of folding-free mappings for certain types of volumetric geometries by combining EGG with transfinite-

E-mail address: j.p.hinz@tudelft.nl (J. Hinz).

interpolation as well as a number of other applications such as time-dependent settings. All applications are provided with example geometries.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

A mathematical description of the target geometry $\bar{\Omega}$ forms an integral part of any application within numerical analysis. Unfortunately, in most applications only a description of $\partial\hat{\Omega}$ is available which is why the generation of a structured or unstructured computational grid that approximates $\hat{\Omega}$ tends to contribute substantially to the overall computational costs. This is further exacerbated by the usage of linear (i.e. straight-sided) elements which often necessitate a high element density along the boundary to achieve an accurate approximation Ω of $\hat{\Omega}$.

Elliptic grid generation (EGG) is a popular PDE-based method to generate structured meshes for complex geometries from a description of no more than their boundaries. Traditionally, the CAD-based description of $\partial\Omega$ is turned into a sufficiently dense point-cloud that dictates the boundary element distribution and the governing equations are solved utilizing a finite-difference approach. Due to the structured nature of this approach, the total amount of unknowns is of the order of the product of the amount of points used in each coordinate-direction. Whenever an accurate description of $\hat{\Omega}$ is desired, this can lead to a disproportionately large number of unknowns, making the approach potentially unfeasible. This is further exacerbated by the fact that the nonlinear governing equations require an iterative approach. The feasibility concerns can therefore be addressed by the development of more efficient iterative approaches, the means to generate better initial guesses and the reduction of boundary elements required for an accurate description of $\partial\hat{\Omega}$ by using, for instance, curved elements.

Isogeometric analysis (IgA) (Hughes et al., 2005) is a recent development in the field of numerical analysis that attempts to bridge the gap between CAD and FEA. A description of the geometry is accomplished by a NURBS-based mapping operator as opposed to a triangulation of the CAD-based input geometry $\hat{\Omega}$. This attempt to facilitate the interaction between CAD and FEA comes with the potential of addressing many of the aforementioned feasibility concerns. We will present an algorithm that is based on the canonical embedding of the principles of EGG into an IgA-framework. After a short motivation in Section 2, a detailed description of how the algorithm attempts to mitigate all the aforementioned feasibility concerns and how a high-quality mapping function is achieved is given in Sections 4 to 8. In Section 11, we will propose numerous possible applications within the realm of IgA, as well as classical meshing applications and demonstrate the applicability of our approach for a number of example geometries.

It is assumed that the reader is familiar with B-spline basis functions and hierarchical spline bases (Hughes et al., 2005; Vuong, 2012; Falini et al., 2015).

2. Motivation

The generation of analysis-suitable parameterizations for complex geometries in IgA-applications from no more than a description of their boundaries remains a challenging task, despite increased efforts to address this issue with various IgA-compatible parameterization methods in recent years. To the best of our knowledge, there exists no golden standard and different types of geometries often require specially-tailored parameterization methods. In the planar case, a parameterization of a geometry Ω is usually built from the linear span of a bivariate tensor-product B-spline space with known boundary control points. The objective is then to appropriately choose the inner control points such that the resulting mapping is (i) bijective and (ii) of good quality with respect to some quality functional. Consequently, most parameterization methods are based on the minimization of a quality functional. Let $\hat{\Omega} = [0, 1]^2$ and let $\mathbf{x} = (x(\xi, \eta), y(\xi, \eta))^T$ be a mapping function with the property that $\mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega$. Furthermore, let

$$[J] = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \tag{1}$$

and

$$[g] = [J]^T [J] = \begin{bmatrix} \mathbf{x}_\xi \cdot \mathbf{x}_\xi & \mathbf{x}_\xi \cdot \mathbf{x}_\eta \\ \mathbf{x}_\xi \cdot \mathbf{x}_\eta & \mathbf{x}_\eta \cdot \mathbf{x}_\eta \end{bmatrix} \tag{2}$$

be the Jacobian and the metric tensor corresponding to the mapping, respectively.

In the following we list the most common quality functionals (Winslow, 1981; Xu et al., 2013; Gravesen et al., 2012).

Area orthogonality: $\int_{\hat{\Omega}} g_{11}^2 g_{22}^2 \mathbf{d}\xi$ or $\int_{\hat{\Omega}} g_{11}^2 + g_{22}^2 \mathbf{d}\xi$

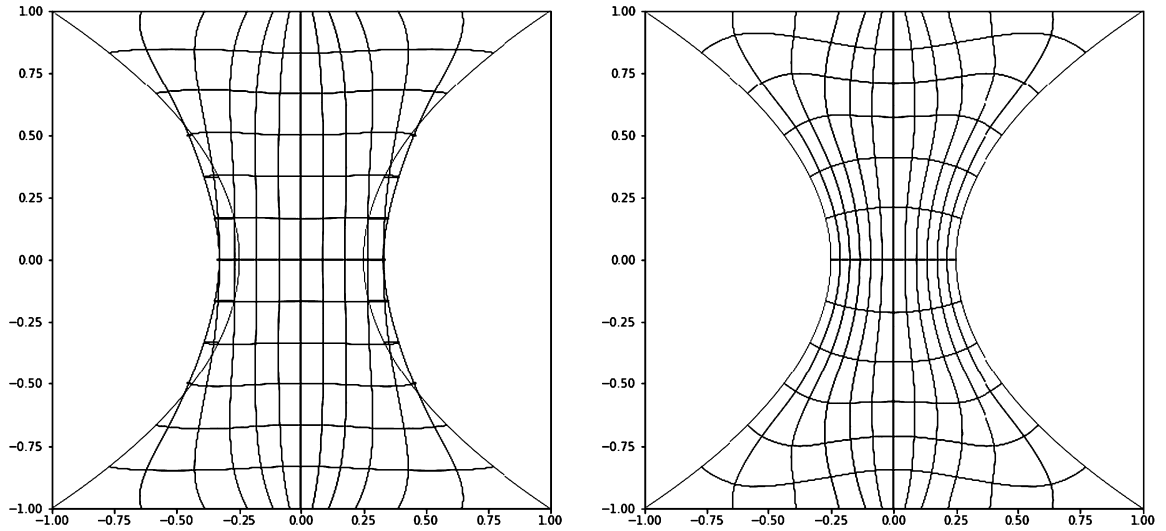


Fig. 1. A geometry parameterization based on the Liao-function in the unconstrained case (left) and the constrained (right). Unlike in the constrained case, the mapping fails to be bijective in the unconstrained case.

$$\begin{aligned}
 \text{Liao: } & \int_{\hat{\Omega}} g_{11}^2 + g_{22}^2 + 2g_{12}^2 \mathbf{d}\xi \\
 \text{Winslow: } & \int_{\hat{\Omega}} \frac{g_{11}^2 + g_{22}^2}{\det[J]} \mathbf{d}\xi \\
 \text{Uniformity: } & \int_{\hat{\Omega}} \|\mathbf{x}_{\xi\xi}\|^2 + \|\mathbf{x}_{\eta\eta}\|^2 + \|\mathbf{x}_{\xi\eta}\|^2 \mathbf{d}\xi \\
 \text{Harmonic energy: } & \int_{\hat{\Omega}} L(x, y, x)^2 + L(x, y, y)^2 \mathbf{d}\xi, \tag{3}
 \end{aligned}$$

where

$$L(x, y, z) = g_{22}z_{\xi\xi} - 2g_{12}z_{\xi\eta} + g_{11}z_{\eta\eta}. \tag{4}$$

Most algorithms minimize either a single or a linear combination of the above quality functionals over the unknown inner control points, such as (Xu et al., 2013) and (Falini et al., 2015).

Unfortunately, the unconstrained minimization of functionals from (3) does not guarantee bijectivity of the resulting mapping \mathbf{x} (see Fig. 1). An exception to this is the Winslow-functional which, as a downside, needs to be initialized with an already bijective mapping. To overcome this shortcoming, constrained minimization methods were proposed in Gravesen et al. (2012), Xu et al. (2011) and Xu et al. (2010). They replace the condition $\det[J] > 0$ by simpler linear or non-linear constraints which constitute sufficient conditions for bijectivity. A major challenge in constrained methods, however, is the finding of a feasible (i.e. bijective) initial guess.

Our main interest lies in the fast on-the-fly generation of analysis-suitable parameterizations within real-world modelling, simulation and optimisation workflows for engineering applications, and hence, our focus is not only on generating analysis-suitable parameterisations of high quality but also on aspects of computational efficiency. This applies both to the efficiency of the process of generating the parameterizations itself and to their suitability for enabling efficient simulations on heterogeneous high-performance computing (HPC) platforms. In practice, these range from a single workstation, which is possibly equipped with one or more hardware accelerator cards like GPUs, FPGAs and Intel's Xeon Phi, to large-scale clusters with hundreds or even thousands of compute nodes.

Inspired by the FEAST project (Becker, 2007), our approach is to map each single patch to a different compute device (CPU, GPU, etc.) and associate the multi-patch topology with the communication paths between hardware resources. Since communication is typically much slower than the actual computations, we try to reduce the amount of data to be exchanged between patches by adopting a discontinuous Galerkin-type multi-patch coupling approach. Our parameterization strategy is moreover designed to avoid topology changes, so that an optimized mapping between the collection of patches and available hardware resources that takes into account the physical proximity of devices and the speed of their interconnects can be pre-calculated at the beginning of the simulation. It is foreseen to implement a dynamic load-balancing strategy at a later

time that splits patches with a too high workload into smaller ones by increasing the multiplicity of individual knots until the patch ‘breaks’ into multiple sub-patches with only C^0 or C^{-1} connectivity. These sub-patches are then redistributed to the available compute resources.

To fully exploit the large computing power of modern multi- and many-core hardware architectures at the patch level, the workload per patch needs to be sufficiently high. Modern GPUs provide up to 16GB of memory and offer more than 5.000 processing cores, which motivates our striving for few patches with ideally hundreds of thousands of degrees of freedom rather than many patches with only a few hundred. Finally, modern processors largely benefit from regular memory access patterns and the use of vectorized operations, which is why we favour tensor-product B-splines, which lead to hardware-friendly band-structured system matrices, over hierarchical B-splines, which bring in all the computational challenges that are known from standard unstructured finite elements like unstructured matrices and overhead-costs due to indirect addressing.

Next to computational efficiency, our parameterization algorithm is moreover designed with reliability and robustness in mind. That is, it is supposed to produce mappings that are both bijective and of sufficiently high quality from an analysis-aware perspective for a wide range of planar geometries without the need for human interaction and/or quality post-checks. As mentioned above, we aim at employing fewer rather than more patches per geometry which is why the segmentation of geometries of genus ≥ 2 is carried out by hand to minimize the amount of patches needed and to avoid topology changes in the time-dependent case. From this perspective the optimal topology for our geometries is usually self-evident. Contrary to methods relying on the quality functionals from (3), our method approximately solves a PDE problem based on the principles of elliptic grid generation (EGG).

In the following we present a brief introduction to EGG.

3. Elliptic grid generation

Elliptic grid generation (EGG) addresses the frequently occurring problem of generating an analysis-suitable description of the geometry from a description of only its boundaries. The technique is commonly applied in finite-difference and finite-volume settings for the generation of structured grids.

Assuming the geometry $\hat{\Omega}$ is topologically equivalent to the unit quadrilateral $\hat{\Omega} = [0, 1] \times [0, 1]$, there exists a bijective mapping $\bar{\mathbf{x}} : \hat{\Omega} \rightarrow \Omega$ as well as an inverse-mapping $\bar{\mathbf{x}}^{-1} : \Omega \rightarrow \hat{\Omega}$ that follows from the assumption that $\partial\hat{\Omega}$ be mapped onto $\partial\Omega$ under $\bar{\mathbf{x}}$ and the converse under $\bar{\mathbf{x}}^{-1}$. Usually, the mathematical operator $\bar{\mathbf{x}}$ is not unique, which is why, besides computing a valid (i.e. bijective) mapping $\bar{\mathbf{x}}$, EGG attempts to compute a mapping that satisfies favorable mathematical properties from a numerical perspective such as evenly-spaced and orthogonal grid lines.

To this end EGG imposes the Laplace equation on the components of the inverse-mapping $\bar{\mathbf{x}}^{-1}$ under the assumption that $\bar{\mathbf{x}}^{-1}|_{\partial\Omega}$ is mapped onto $\partial\hat{\Omega}$. Assuming that the free topological variables are given by the tuple (ξ, η) and the geometric variables by (\bar{x}, \bar{y}) , the equation takes the form

$$\begin{cases} \Delta\xi(\bar{x}, \bar{y}) = 0 \\ \Delta\eta(\bar{x}, \bar{y}) = 0 \end{cases} \quad \text{s.t. } \bar{\mathbf{x}}^{-1}|_{\partial\Omega} = \partial\hat{\Omega}. \quad (5)$$

Since one is generally not interested in $\bar{\mathbf{x}}^{-1}$, the above problem is inverted for the components $x(\xi, \eta)$ and $y(\xi, \eta)$ of $\bar{\mathbf{x}}$ and scaled in order to yield a mapping-generating equation that is suitable for a computational approach. The resulting equations read

$$\begin{cases} L(\bar{x}, \bar{y}, \bar{x}) = 0 \\ L(\bar{x}, \bar{y}, \bar{y}) = 0 \end{cases} \quad \text{s.t. } \bar{\mathbf{x}}|_{\partial\hat{\Omega}} = \partial\Omega, \quad (6)$$

with the functional $L(x, y, z)$ defined as in (4).

The reason behind the imposition of the Laplace equation on $\bar{\mathbf{x}}^{-1}$ as opposed to $\bar{\mathbf{x}}$ follows from the observation that the target space of $\bar{\mathbf{x}}^{-1}$ is convex, and hence, the solution of (6) is bijective (Azarenok, 2009).

Furthermore, it can be shown (Gravesen et al., 2012) that the solutions of (5) and (6) are equal to the unique minimizer of the Winslow functional from (3). Also, (6) justifies the usage of the ‘harmonic energy’ functional from (3). As the exact solution of (6) is bijective, algorithms based on the principles of EGG belong to the small subset of algorithms that can reliably produce bijective mappings without the need for constraining. To the best of our knowledge, the only other planar parameterization method with this property is the one from Nian and Chen (2016).

Generally, equation (6) cannot be solved analytically. In the following, we shall present a numerical solution technique for the approximate solution of (6) that is based on the principles of Isogeometric Analysis (IgA).

4. Discretization

Traditionally, the system of equations (6) is approximately solved utilizing a finite-difference approach, yielding a discrete set of grid points with known connectivity (see Thompson et al., 1998). This approach makes sense whenever a structured representation of the geometry with linear elements is desired as in most finite-difference (FD) and structured

finite-volume (FV) settings. A spline-based approach has been suggested in Lamby and Brakhage (2007) and Manke (1989) by the utilization of a collocation technique.

In an IgA-setting, the most natural approach of tackling (6) is a variational approach with some pre-chosen spline basis $\Sigma \subset C^1$. This can be either a tensor-product B-Spline basis or a (truncated) hierarchical spline basis (Giannelli et al., 2012), where the former has the advantage of being structured and the latter has the advantage of allowing for local refinement, potentially converging to a bijective mapping \mathbf{x} with fewer degrees of freedom (see Section 8). The approximation of $\bar{\mathbf{x}}$ is then of the form

$$\mathbf{x}(\xi, \eta) = \sum_i \mathbf{c}_i w_i(\xi, \eta), \quad (7)$$

where $\Sigma = \{w_1, \dots, w_N\}$ and $\mathbf{c}_i = (c_i^x, c_i^y)^T$. The spline-based mapping is denoted by \mathbf{x} as opposed to $\bar{\mathbf{x}}$ in order to stress that \mathbf{x} should be regarded as an approximation of $\bar{\mathbf{x}}$ since the condition $\bar{\mathbf{x}}|_{\partial\hat{\Omega}} = \partial\bar{\Omega}$ generally cannot be discretized without discretization errors.

The variational counterpart of (6) is given by

$$\forall w_i \in \Sigma_0 : \begin{cases} \int w_i L(x, y, x) d\xi = 0 \\ \int_{\hat{\Omega}} w_i L(x, y, y) d\xi = 0 \end{cases} \quad \text{s.t. } \mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega, \quad (8)$$

where $\partial\Omega$ is some approximation of $\partial\bar{\Omega}$ contained in the span of $\Sigma|_{\partial\hat{\Omega}}$ and $\Sigma_0 = \{w_i \in \Sigma \mid w_i|_{\partial\hat{\Omega}} = 0\}$. The components of \mathbf{x} , namely $x(\xi, \eta)$ and $y(\xi, \eta)$, are expressed as elements from the span of Σ and the condition $\mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega$ enters the governing equations as a Dirichlet boundary condition. With this approach, any solution of (8) will automatically be in the form of (7), as is mandatory in an IgA-setting. In the following, we shall assemble the *interior* control points, i.e. those that are not predetermined by the boundary parameterization, into one global vector of unknowns denoted by \mathbf{c} . Let $\mathbf{c}_i = (c_i^x, c_i^y)^T$, $\mathcal{I}_0 = \{i \in \mathbb{N} \mid w_i \in \Sigma_0\}$ and $\mathcal{I} = \{i \in \mathbb{N} \mid w_i \in \Sigma \setminus \Sigma_0\}$, we introduce the vectors

$$\begin{cases} \mathbf{c}^x = (\dots, c_i^x, \dots) \\ \mathbf{c}^y = (\dots, c_i^y, \dots) \end{cases}, \quad \text{s.t. } i \in \mathcal{I}_0, \quad (9)$$

and define the global vector of unknowns

$$\mathbf{c} = (\mathbf{c}^x, \mathbf{c}^y)^T. \quad (10)$$

With this notation, the sum from (7) can be split into two sums, corresponding to the unknown and known control points

$$\mathbf{x}(\xi, \eta) = \underbrace{\sum_{i \in \mathcal{I}_0} \mathbf{c}_i w_i}_{\text{unknown}} + \underbrace{\sum_{j \in \mathcal{I}} \mathbf{c}_j w_j}_{\text{known}}, \quad (11)$$

where the known control points follow from the discretization of the given boundary contours (see Section 5). With (11) in mind, we can formulate (8) as a coupled system of nonlinear equations $\mathbf{F}(\mathbf{c})$ whereby our objective is to determine its root, i.e.

$$\mathbf{F}(\mathbf{c}) = 0. \quad (12)$$

4.1. C^0 vs. C^1 discretizations

The condition $\Sigma \subset C^1$ is a result of the presence of second order derivatives in (8). In IgA, the choice of C^1 -continuous basis functions is straight forward, whereas the design of classical finite elements with C^1 -continuity is more complicated. It is of course possible to reformulate (8) such that a (locally) C^0 -continuous basis can be used. Since (5) and (6) are equivalent one might perform partial integration on the variational counterpart of (5) to obtain

$$\forall w_i \in \Sigma_0 : \begin{cases} \int \nabla w_i \cdot \nabla \xi d\mathbf{x} = 0 \\ \int_{\Omega} \nabla w_i \cdot \nabla \eta d\mathbf{x} = 0 \end{cases} \quad \text{s.t. } \mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega. \quad (13)$$

Here, the integral has been carried out over Ω as opposed to $\hat{\Omega}$, which is equally valid.

Transforming above system to local coordinates, we find

$$\forall w_i \in \Sigma_0 : \begin{cases} \int_{\Omega} \frac{1}{\det[J]} (w_{i,\xi} g_{22} - w_{i,\eta} g_{12}) d\xi & = 0 \\ \int_{\Omega} \frac{1}{\det[J]} (-w_{i,\xi} g_{12} + w_{i,\eta} g_{11}) d\xi & = 0 \end{cases} \quad \text{s.t. } \mathbf{x}|_{\partial\hat{\Omega}} = \partial\Omega. \quad (14)$$

As both (8) and (14) are nonlinear in the \mathbf{c}_i of (7), and iterative solution approach is mandatory for which an initial guess is required.

In the case of system (8), any initial guess \mathbf{c}^0 is valid. In (14), however, we see the Jacobian determinant $\det[J]$ appearing in the denominator so that the initial guess must satisfy $\forall(\xi, \eta) \in \hat{\Omega} : \det[J](\xi, \eta) \neq 0$. This leads to the conclusion that the initial guess already needs to be bijective itself which defeats the purpose of this approach. System (14) can, however, be utilized to improve the parametric properties of a bijective mapping that is, for instance, produced by (8). A better option in this case is to minimize the Winslow functional:

$$m_w(\mathbf{c}) = \frac{x_\xi^2 + x_\eta^2 + y_\xi^2 + y_\eta^2}{\det[J]}(\mathbf{c}). \quad (15)$$

The minimization problem takes the form

$$\int_{\hat{\Omega}} m_w(\mathbf{c}) \rightarrow \min_{\mathbf{c}}, \quad (16)$$

which can usually be done without constraining since $(\det[J])^{-1}$ naturally penalizes (nearly) unfeasible solutions. It is seen that (16), just as (14) allows for C^0 -continuities but requires a bijective initial guess as well. It is, however, superior to (14) in that it enables one to find the global minimizer of (16) over the span of the chosen spline space, which (14) does not guarantee.

In an application, where the objective is to find a bijective mapping \mathbf{x} , starting with a C^0 -continuous basis is thus not a viable choice. We shall discuss how to acquire a mapping containing C^0 -continuities from a mapping that is initially $C^{\geq 1}$ -continuous in Section 9.

5. Contour approximation and the choice of basis

In Section 3, we assumed that a parameterization of $\partial\bar{\Omega}$ by $\bar{\mathbf{x}}|_{\partial\hat{\Omega}}$ is given. This parameterization, however, does not have to be contained within the span of $\Sigma_0|_{\partial\hat{\Omega}}$. Denoting the southern, eastern, northern and western segments of $\partial\bar{\Omega}$ by $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4 , respectively, we shall assume that the corresponding boundary parameterizations, $\bar{f}_i : [0, 1] \rightarrow \Gamma_i$ that jointly make up $\partial\bar{\Omega}$ are non self-intersecting.

The \bar{f}_i can be given in terms of analytic functions or spline-functions. We will also consider the case in which $\bar{\mathbf{x}}|_{\partial\hat{\Omega}}$ is only discretely available as a collection of points $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\} \subset \mathbb{R}^2$.

Our objective in this section shall be to present techniques to choose spline spaces capable of resolving the given contour to any desired accuracy while satisfying certain properties whose applications will become apparent in later sections.

We shall restrict ourselves to tensor-product B-Spline bases for now and shall make some generalizations to hierarchical spline bases by the end of this section.

5.1. Contours given as analytic functions

In the case of a set of analytic functions as input, the obvious discretization technique is that of an L_2 -projection. Assuming we are starting with uniform knot-vectors Ξ_ξ and Ξ_η with corresponding bivariate $p \geq 2$ B-Spline basis Σ^0 with global $C^{\geq 1}$ continuity, we can sequentially perform a least-squares fit of the \bar{f}_i onto $\Sigma^0|_{\Gamma_i}$. Since this fit will generally not be exact, it is possible that the control points corresponding to the corners of the domain may receive unequal values. It is therefore mandatory to carry out a constrained fit instead, constraining control points corresponding to the corners of the domain to, for instance, the function evaluations of the \bar{f}_i in these corners.

Denoting by f_i^0 the constrained least-squares fit of \bar{f}_i onto Σ^0 , we can introduce the following element-wise average residual function

$$E(f_i^0) = \sum_{\epsilon_i} \frac{1}{|\epsilon_i|} \int_{\epsilon_i} \|f_i^0(\xi) - \bar{f}_i(\xi)\|^2 d\xi, \quad (17)$$

where the ϵ_i denote the elements corresponding to Ξ_ξ or Ξ_η , respectively, and $|\epsilon_i|$ are their length in the parametric domain. The residual function from (17) serves as an indicator for the quality of the fit both globally and locally. The local quality can be assessed by the contributions of each summand to the overall sum. As a result, it can serve as a local refinement criterion whenever the average residual on one element exceeds some threshold value δ . Looping over all the

elements ϵ_i , we can refine those on which the residual exceeds δ by adding a new knot in its center and perform the (constrained) least-squares fit again until the residual on each element is deemed sufficiently small. Once the algorithm terminates, a sequence of nested B-Spline bases $\Sigma^i, i \in \{1, \dots, m\}$ with corresponding boundary control points $\{\mathbf{c}_j^i\}$ and segmentations of $\partial\hat{\Omega}$ into disjoint elements is available.

We shall use the Σ^i and corresponding boundary control-points $\{\mathbf{c}_j^i\}$ to construct good initial guesses for a computation approach of the variational equations (8) in Subsection 6.3.

5.2. Contours given as B-Spline functions

Whenever the contours are given as B-Spline functions, a natural way to choose a basis is to use a tensor-product B-Spline basis that corresponds to the given contours. They may, however, not be compatible in the sense that the knot-vectors Ξ_1, Ξ_3 and Ξ_2, Ξ_4 corresponding to the f_i do not coincide. In this case one has to take the union of the knot-vectors corresponding to each coordinate direction and prolong the control points to the unified basis. This can lead to a basis of large cardinality, potentially making further computations less feasible. A remedy might be the usage of the principles from Subsection 5.1. One can use the same approach to find an approximation of the given contours to any desired accuracy, the only difference being that one should perform the projection over a set of elements on which both the input function as well as the projection basis is C^∞ -continuous (the projection matrix can be assembled over the smaller set of elements corresponding to the projection basis though). This has the potential disadvantage of being computationally expensive since the knot-vector corresponding to the input data may be quite dense. Considering, however, that the projection corresponds to a univariate integral, the computational costs are still negligible with respect to the expected costs of solving equation (8). With this approach, we were often able in our numerical solutions to reduce the amount of basis functions required by a factor of four or more with hardly any loss of quality.

5.3. Contours given as a point cloud

In many engineering applications, it is common to define the boundary contours discretely as an ordered set of points. Often the coordinates of the points are a function of a set of shape parameters $\alpha = (\alpha_1, \dots, \alpha_L)^T$.

Whenever this is the case, we split the input point-cloud into four point-clouds $P^i = \{\mathbf{p}_1^i, \mathbf{p}_2^i, \dots, \mathbf{p}_{K_i}^i\} \subset \mathbb{R}^2, i \in \{1, 2, 3, 4\}$, each corresponding to one part of $\partial\hat{\Omega}$. The first step in computing an approximate mapping \mathbf{x} is the construction of a spline fit. This fit can either be exact or approximate. Selecting any monotone increasing sequence $\{\xi_1 = 0, \xi_2, \dots, \xi_{K_i-1}, \xi_{K_i} = 1\}$, the objective is to construct four spline curves f_i for which $\|f_i(\xi_j) - \mathbf{p}_j^i\|, j \in \{1, \dots, K_i\}$ is either zero (exact) or below a certain threshold (approximate).

This is accomplished by recursively constructing f_i from the span of univariate spline bases of successively increasing cardinality, in a way similar to that proposed in Subsection 5.1 in the least-squares sense.

For this, we can define the error functional

$$E(f_i^k) = \sum_j \left\| f_i^k(\xi_j) - \mathbf{p}_j^i \right\|^2, \quad (18)$$

where k denotes the index of the current recursion. Again it is advisable to constrain the corner control points to the corner points of the point cloud.

Functional (18) can then be utilized not only as an indicator of the global quality of the interpolation but also as local quality indicator by taking the individual contributions of the summands to the overall sum into account. The local contributions then serve as a local refinement criterion.

In this setting, it can be noted that constructing a collocation $\bar{C}(\xi)$ of the point-cloud that satisfies $\|\bar{C}(\xi_i) - p_i\| < \epsilon_0$ (exact for $\epsilon_0 = 0$) and approximating it utilizing the principles from Subsection 5.2 constitutes a possible strategy, too. Thanks to the triangle inequality, it follows from

$$\forall j \in \{1, \dots, K_i\} : \left\| f_i^k(\xi_j) - \bar{C}(\xi_j) \right\| < \epsilon, \quad (19)$$

that

$$\forall j \in \{1, \dots, K_i\} : \left\| f_i^k(\xi_j) - \mathbf{p}_j^i \right\| < \epsilon_0 + \epsilon. \quad (20)$$

The construction of $\bar{C}(\xi)$ can be accomplished with one of the FITPACK (Dierckx, 1995) routines. The question remains that if the objective is to push the residual below a certain threshold value, why not set ϵ_0 to the desired threshold and use this collocation right away? This has two main reasons:

1. A direct collocation does not yield a hierarchy of spline approximations which can be utilized to improve the robustness of the computation of the final mapping (see Section 6.3).

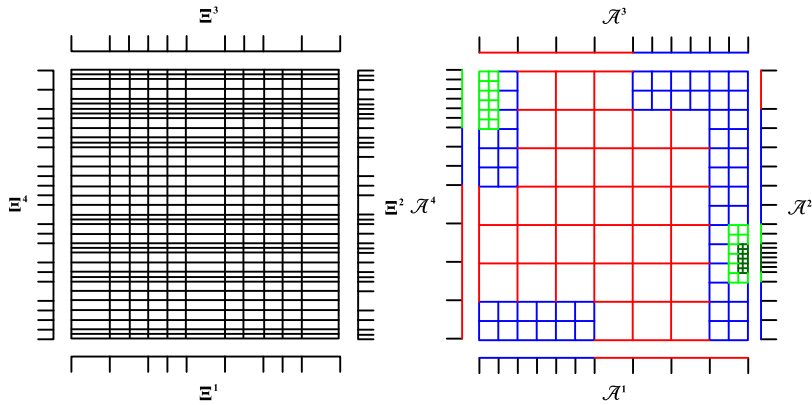


Fig. 2. Example of a segmentation of the computational domain into elements resulting from the four knot-vectors $\Xi_i, i \in \{1, 2, 3, 4\}$ that correspond to the bases utilized in the approximation of the contours by spline curves (left). The segmentation results from taking the union of the knot-vectors corresponding to the same coordinate directions (the tuples (Ξ_1, Ξ_3) and (Ξ_2, Ξ_4)) without knot repetitions. The left picture shows a segmentation of $\hat{\Omega}$ into elements that follows from the univariate hierarchical segmentations of the unit interval $\mathcal{A}_i, i = 1, 2, 3, 4$ into elements and is carried out in the canonical way shown above. The various colors indicate the various levels in the element hierarchy. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

2. In Subsection 11.1, we shall encounter applications in which it is beneficial to base the contour approximation on a basis with a predefined structure. The refinement based on the halving of element sizes dictates exactly where knots can be located. This is advantageous when two or more mappings that have different knot-vectors need to be pulled onto a unified knot-vector because then the amount of knots one needs to add to each of the mappings tends to be far less than when the knots can be arbitrarily placed.

This approach has the advantage of being computationally more stable, since the (rare) possibility of the local amount of spline functions exceeding the local amount of points is excluded. It is, however, computationally more expensive since it requires a fit through the (possibly large) set of points P.

5.4. Choice of basis

Assigning an index to the southern, eastern, northern and western boundaries of the geometry in ascending order, we have four sequences $\{f_i^j\}, (i, j) \in \{1, 2, 3, 4\} \times \{1, \dots, m\}$ of successively more accurate descriptions of the boundary contours. These descriptions are utilized to form a hierarchy of segmentations of $\hat{\Omega}$ into elements with corresponding canonical spline bases. In the case of an ordinary spline basis, the mesh hierarchy is formed by simply taking the tensor-product of the unified knot-vectors corresponding to the northern and southern boundaries as well as the western and eastern boundaries (see Fig. 2 (left)).

5.5. Generalization to hierarchical spline-bases

The variational equations (8) are also compatible with a $C^{\geq 1}$ hierarchical spline basis. The approximation of the boundary contours can be accomplished by starting with a coarse initial (tensor-product) basis Σ^0 and performing hierarchical refinements at the boundary at places where the residual is too large in a way depicted in Fig. 2 (right).

This has the potential advantage of being computationally less expensive since refinement is local in nature and can be tuned such that it is carried out only where needed. Apart from the refinements at the boundaries that can be based on pushing the residual below a predefined threshold, it is difficult to make an a priori prediction of which interior elements need to be refined. This, in turn, increases the chances of the solution of the governing equations (8) leading to unsatisfactory results.

In case an approach with a hierarchical spline-basis is preferred, we recommend the usage of one of the algorithms suggested in Falini et al. (2015).

6. Computational approach

After the meshing procedure from Section 5 has been completed, we have a nested sequence of domain segmentations $\mathcal{H}_i, i \in \{1, 2, \dots, m\}$ with corresponding (tensor- or hierarchical-) spline bases $\Sigma^i, i \in \{1, 2, \dots, m\}$ and steadily improving approximations of the boundary contours from the spans of the Σ^i .

In Subsections 6.1 and 6.2, we shall propose two possible computational approaches for solving equation (12), whereas in Section 6.3, we present ways to construct good initial guesses for the proposed approaches. The computational approaches have been implemented in the Python-library Nutils (van Zwieten et al., 2016).

6.1. Truncated Newton-approach

The variational equations (8) or (14), with some appropriately chosen spline-basis Σ , will lead to a root-finding problem of the form

$$\mathbf{F}(\mathbf{c}) = 0, \quad (21)$$

where \mathbf{c} refers to the internal (unknown) degrees of freedom.

Due to the nonlinear nature of this equation, an iterative approach to finding its root \mathbf{c} is mandatory. Even though algorithms based on the principles of EGG traditionally employ a Picard-based approach (as in Lamby and Brakhage, 2007 and Manke, 1989), we shall present an approach that is based on Newton's method. Defining

$$\mathbf{F}^k \equiv \mathbf{F}(\mathbf{c}^k) \quad (22)$$

and

$$\left(\frac{\partial \mathbf{F}}{\partial \mathbf{c}}\right)^k \equiv \left.\frac{\partial \mathbf{F}}{\partial \mathbf{c}}\right|_{\mathbf{c}=\mathbf{c}^k}, \quad (23)$$

a Newton-step is calculated as follows

$$\Delta \mathbf{c}^k = - \left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{c}}\right)^k \right]^{-1} \mathbf{F}^k. \quad (24)$$

The next iterate then becomes

$$\mathbf{c}^{k+1} = \mathbf{c}^k + \delta \Delta \mathbf{c}^k, \quad (25)$$

where $\delta \leq 1$ is a damping factor. The Nutils-internal Newton-solver is equipped with sophisticated line-search capabilities. Assuming that

$$\mathbf{F}(\mathbf{c}^k + \delta \Delta \mathbf{c}^k) = A + B\delta + C\delta^2 + D\delta^3, \quad (26)$$

the constants A, B, C, D are estimated from the current and updated tangents and residuals and δ is selected such that (26) is minimized in the Euclidean norm in order to reduce the amount of function evaluations required in \mathbf{F} .

The above procedure is repeated until

$$\|\mathbf{F}(\mathbf{c}^n)\| < \epsilon, \quad (27)$$

where ϵ is some convergence threshold.

6.2. Pseudo time-stepping

Should the Newton-approach fail to converge, the solver can fall back on the technique of pseudo time-stepping (Coffey et al., 2003) which has an even higher chance of convergence at the expense of an increased computational effort. In practice, we have observed that the need to fall back on this approach is a seldom occurrence thanks to the reliability of the truncated Newton-approach (in combination with a good initial guess, see Subsection 6.3).

The idea of pseudo time-stepping is to look for the steady-state solution of

$$\frac{\partial}{\partial t} \mathbf{x}(t) = -\mathbf{G}(\mathbf{x}(t)), \quad (28)$$

where

$$\mathbf{G}(\mathbf{x}(t)) = \begin{bmatrix} g_{22}x_{\xi\xi}(t) - 2g_{12}x_{\xi\eta}(t) + g_{11}x_{\eta\eta}(t) \\ g_{22}y_{\xi\xi}(t) - 2g_{12}y_{\xi\eta}(t) + g_{11}y_{\eta\eta}(t) \end{bmatrix} \quad (29)$$

We discretize in space utilizing a Galerkin approach and in time utilizing a backward Euler scheme which leads to the following equation

$$\begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} \frac{\mathbf{c}^{n+1} - \mathbf{c}^n}{\Delta t_n} = -\mathbf{F}^{n+1}, \quad (30)$$

where M denotes the mass matrix of the basis utilized.

The drawback of this discretization is its non-linear right-hand side term. We circumvent this issue with a first order expansion around \mathbf{F}^n

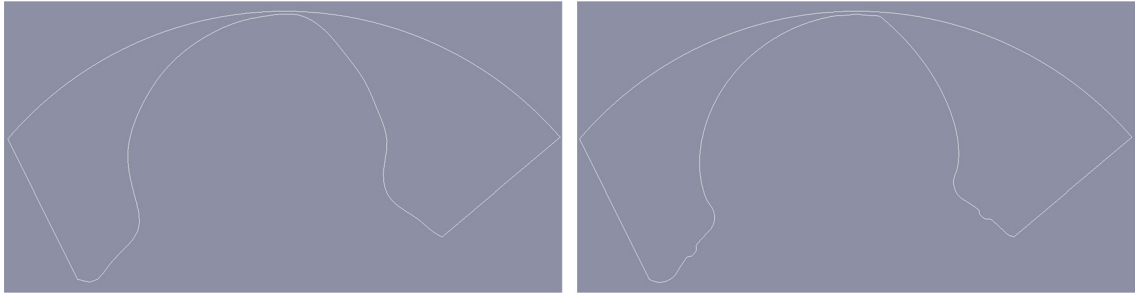


Fig. 3. Contours corresponding to the first (left) and last level in the hierarchy (right).

$$\mathbf{F}^{n+1} \simeq \mathbf{F}^n + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{c}} \right)^n \Delta \mathbf{c}^n. \tag{31}$$

Substitution and rearrangement yields

$$\left[\frac{1}{\Delta t_n} \begin{pmatrix} M & 0 \\ 0 & M \end{pmatrix} + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{c}} \right)^n \right] \Delta \mathbf{c}^n = -\mathbf{F}^n. \tag{32}$$

As proposed in Kelley and Keyes (1998), the time step selection is based on the following recursive formula

$$\Delta t_n = \Delta t_{n-1} \frac{\|\mathbf{F}^{n-1}\|}{\|\mathbf{F}^n\|}. \tag{33}$$

Again, the iteration is terminated once

$$\|\mathbf{F}(\mathbf{c}^n)\| < \epsilon. \tag{34}$$

6.3. Choice of the initial guess

The computational approaches from Subsections 6.1 as well as 6.2 are iterative and thus require an initial guess \mathbf{c}^0 . The quality of this initial guess can make a tremendous difference in the success or failure of the approach as well as the amount of iterations needed. We shall dedicate this section to constructing good initial guesses.

The preprocessing step from Section 5 leaves us with the nested sequence of spline spaces $\Sigma_i, i \in \{1, \dots, m\}$ and corresponding known boundary control points. Each of these spline spaces leads to a nonlinear variational problem of the form

$$\mathbf{F}_i(\mathbf{c}_i) = 0, \tag{35}$$

where the subscript i indicates the level in the hierarchy.

It is advisable to utilize this hierarchical structure to our advantage. The idea is to solve $\mathbf{F}_1(\mathbf{c}_1) = 0$ and manipulate its solution to serve as an initial guess for the problem $\mathbf{F}_m = 0$. An initial guess for \mathbf{F}_1 is constructed utilizing transfinite-interpolation (Coons, 1967) and solved either with the approach proposed in Sections 6.1 or 6.2. Depending on the complexity of the problem, we have observed that the truncated Newton-approach converges within about 4–6 iterations. As the cardinality of Σ_1 is significantly lower than that of Σ_m , a large amount of iterations does not compromise the feasibility of the algorithm since one iteration is cheap. Assuming that we have found the root \mathbf{c}_1 , we prolong the mapping from the first level in the hierarchy to the basis Σ^m and impose the discrepancy between the boundary control points of the prolongation and the boundary control points corresponding to the m -th level as a Dirichlet boundary condition on a linear-elasticity (Falk, 1939) problem. In cases where a large number of boundary control points is needed in order to accurately resolve the target contours (i.e. m is large), it is recommended not to jump from the first to the last level in the hierarchy right away but to solve some of the intermediate variational problems in order to arrive with a better initial guess at the m -th level. A good guideline is to ensure that the cardinality of the bases in the hierarchy quadruples after each level.

In the following we shall illustrate this approach by applying it to the target contours depicted in Fig. 3. The left and right pictures show a plot of the least-squares fit of the target contours with respect to Σ^1 and Σ^m , respectively. Fig. 4 shows a plot of the mapping resulting from the first level in the hierarchy. The mapping is prolonged to the canonical B-Spline basis corresponding to the domain segmentation depicted in Fig. 5 (left).

Fig. 5 (right) shows the mapping generated from the linear elasticity problem, which serves as an initial guess to $\mathbf{F}_m = 0$. The plot of the Jacobian determinant reveals that the initial guess succeeds in being bijective and is therefore already analysis-suitable. Solving the variational equations thus only serves as to improve the parametric properties in this case. Fig. 6 (right) shows the parametric properties of the initial guess by the narrow gap of the geometry compared to an inferior prolongation method (left) (Fig. 7).

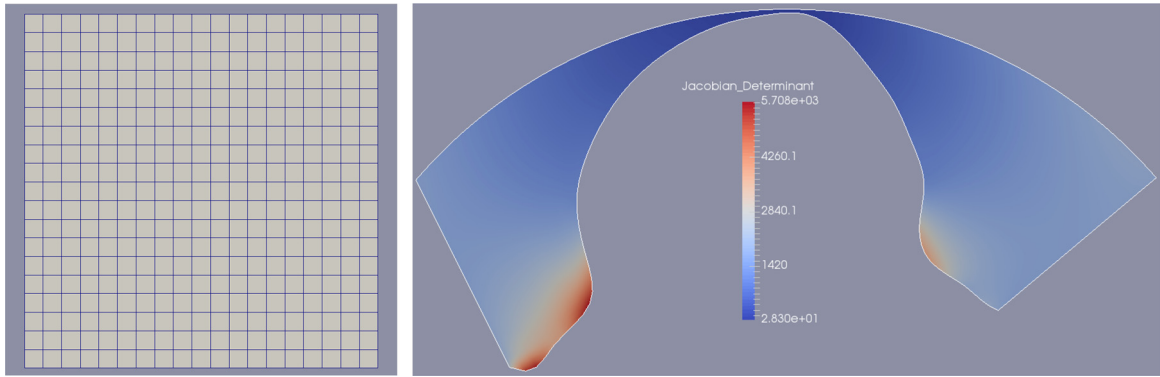


Fig. 4. The domain segmentation (left) and mapping (right) resulting from the first level in the hierarchy. The color depicts the value the Jacobian determinant assumed at each point.

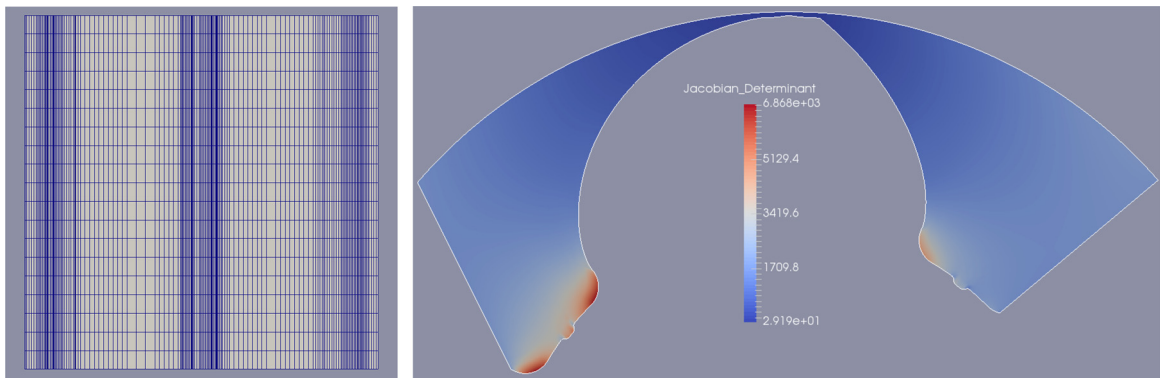


Fig. 5. Domain segmentation corresponding to the highest level in the hierarchy (left), the initial mapping acquired with the linear elasticity method (right).

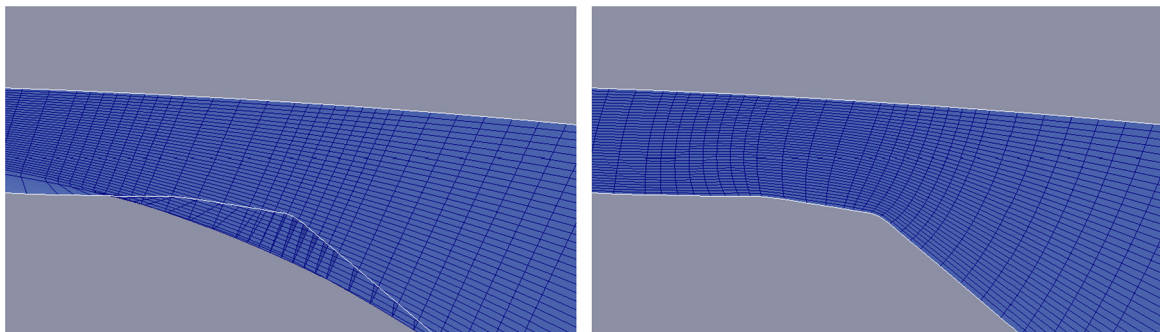


Fig. 6. The parametric properties by the narrow gap of the geometry of the initial guess acquired by acting with the canonical prolongation matrix on the internal DOF's and replacing the boundary DOF's (left) vs. with linear elasticity (right). Note that the initial guess depicted on the left fails to be bijective.

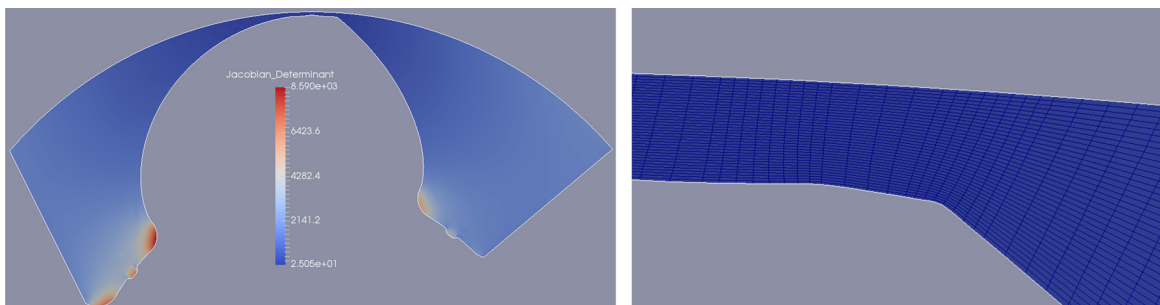


Fig. 7. The final mapping (left) and its parametric properties by the narrow gap (right).

In practice, the solver typically converges within 3–4 iterations on the finest level and convergence failure is extremely uncommon. We have found this hierarchical approach to greatly improve the robustness as well as the efficiency of the algorithm. In cases where the boundary control points are not the result of one of the methods from Section 5 but are fixed from the e.g. CAD-input, we have often successfully tackled the variational equations with an artificially created hierarchy. This is done by removing knots from the input knot-vectors and performing a least-squares projection of the Dirichlet-extension onto the coarsened knot-space. We have found this approach to lead to more robustness than a direct approach with transfinite interpolation.

Heuristically, the initial guess acquired with the linear-elasticity approach usually succeeds in being bijective. Since it requires one to solve one negligibly small non-linear problem as well as a linear problem of the size of Σ_0^m , we conclude that its complexity is comparable with linear parameterization techniques while yielding superior results. Furthermore, it constitutes a (quasi-)linear parameterization technique that may be able to produce feasible initial guesses for constrained optimization approaches such as Xu et al. (2011), Gravesen et al. (2012). One may also choose to use it for simulation without further optimization, in that case we might regard the linear-elasticity mapping as a perturbation of the optimized mapping. Distortions in the mapping have a less severe effect on its quality from an analysis-aware perspective for higher values of p (Lipton et al., 2010).

7. Reparameterization techniques

The parametric properties of the contours have a significant influence on the quality of the resulting mapping since they enter the equations as a Dirichlet boundary-condition. In the following we shall discuss various parameterization techniques and give recommendations about which technique should be used in which setting.

7.1. Arc-length based parameterization techniques

For an arbitrary mapping, the best option is typically to start with an (approximate) arc-length parameterization. In the case of a point-cloud $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_K\}$, this is easily accomplished by a proper choice of the collocation abscissa $\{0 = \xi_1, \xi_2, \dots, \xi_K = 1\}$. Defining l_i recursively by

$$l_i = l_{i-1} + \|\mathbf{p}_{i+1} - \mathbf{p}_i\|, \tag{36}$$

starting with $l_1 = 0$, and ending on l_K , an approximate arc-length parameterization (cord-length parameterization) can be accomplished by choosing

$$\xi_i = \frac{l_{i-1}}{l_K}. \tag{37}$$

In case of a parametric curve $f(\xi)$ as input, an option is to generate a point-cloud P from it by performing a large number of function evaluations. A reparameterization function $g(\xi)$ is then generated by performing a monotone increasing interpolation between the ξ_i acquired with the above procedure. An alternative way to approximately reparameterize is to introduce the univariate reparameterization basis $\sigma = \{B_1, \dots, B_n\}$ (ideally with $p \geq 2$) and define

$$g(\xi) = \sum_i c_i B_i, \tag{38}$$

with $c_0 = 0$ and $c_n = 1$. An approximate arc-length parameterization is then achieved by minimizing

$$A(c_2, \dots, c_{n-1}) = \int_0^1 \left(x_\xi^2 + y_\xi^2 \right) d\xi, \tag{39}$$

where $f(g(\xi)) = (x, y)^T$, over the c_i .

For an arbitrary geometry this parameterization is the default choice. When it is known that the geometry is subject to extreme aspect ratios, an arc-length parameterization might be insufficient. In this case it is beneficial to choose the ξ_i such that grid-lines make right angles with the contours (see Fig. 8). For this purpose we will propose an automated constrained arc-length parameterization method in the next subsection.

7.2. Constrained cord-length techniques

In many settings, we have observed that the arc-length based parameterization leads to unsatisfactory results.

Since the boundary contours are imposed strongly as a Dirichlet boundary condition, their parametric properties have a profound influence on the resulting grid. Fig. 10 shows the contours of a challenging geometry. This geometry is the result of a spline interpolation of a very dense point cloud. The challenge arises as a result of a quick alternation between very wide and very narrow gaps.

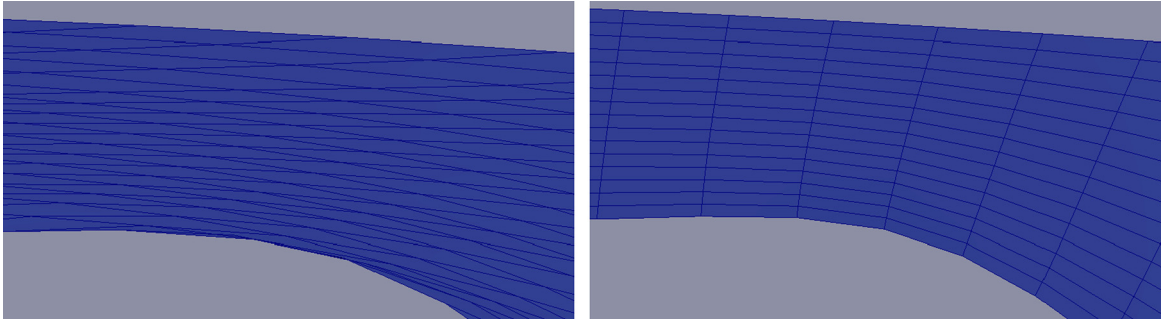


Fig. 8. The mappings resulting from an arc-length parameterization (left) and one where the ξ_i have been chosen so as to minimize the deviation from right angles by a narrow gap (right). The reparameterization has a profound influence on the quality of the resulting mapping. The unparameterized mapping (left) not only exhibits mathematically less favorable properties but also fails to be bijective even though a comparable mesh resolution is used.

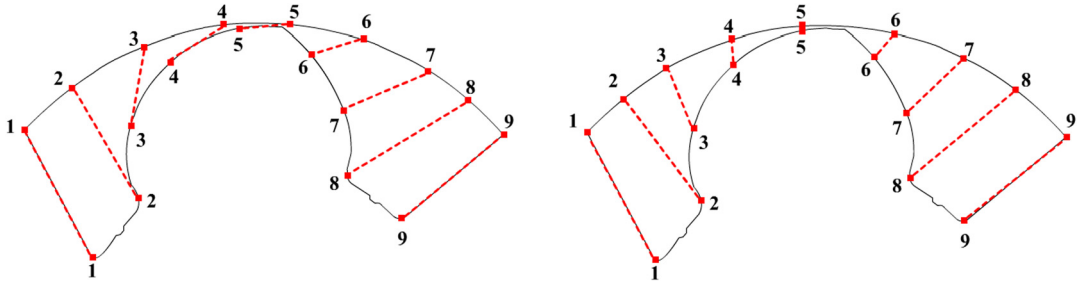


Fig. 9. Depiction of the expected grid lines for cord-length parameterized (left) and constrained cord-length parameterized (right) boundary contours. The parametric properties of the boundary contours are of major importance whenever extreme aspect ratios are involved. The cord-length parameterization can be expected to lead to unsatisfactory results, whereas the more flexible constrained cord-length parameterization produces superior results.

As discussed in 2, a lot of effort has gone into IgA-compatible parameterization techniques in recent years. To the best of our knowledge, however, there only exist a small number of articles dealing with the impact of the parametric properties of the boundary contours on the final mapping. Xu et al. (2014) discusses a technique to change the parametric properties of a curve while retaining its shape. This is accomplished by representing the curve in a related NURBS-basis over a different knot-vector. The parametric properties can be tuned to some quality functional with one given degree of freedom. On the other hand, Xu et al. (2018) attempts to tackle challenging geometries by segmenting them into a large number of Bézier patches via quadrangulation on the Bézier control net. The parametric properties of each patch are then optimized based on a local quality functional.

In the following, we shall present a reparameterization technique that is especially tuned for tube-like shapes subject to extreme aspect ratios as the one in Fig. 10. For these kinds of challenging geometries it is important that the same parametric value is assumed by the narrow gaps on opposite boundaries in order to ensure that emanating grid lines are short and orthogonal to the boundary contours (see Fig. 9). This is beneficial for both the convergence properties of the algorithm as well as the quality of the resulting mapping from a numerical perspective.

To address the aforementioned concerns, we have often successfully employed a more flexible counterpart of the cord-length parameterization, which we shall henceforth refer to as the ‘constrained cord-length parameterization’. In the following we shall assume that the boundary contours are given in terms of a point cloud. Should this not be the case, a point cloud can easily be formed by performing a large number (uniformly spaced) function evaluations over the parametric domain $[0, 1]$.

Based on the observation that it is crucial that the same parametric value is attained on opposite boundaries in the presence of narrow gaps, we have developed an algorithm that forms a cord-length parameterization while satisfying these additional constraints. Given two point clouds $P_1 = \{\mathbf{p}_1^1, \mathbf{p}_2^1, \dots, \mathbf{p}_n^1\}$ and $P_2 = \{\mathbf{p}_1^2, \mathbf{p}_2^2, \dots, \mathbf{p}_m^2\}$, the first step is to form a cord-length parameterization of P_1 (see Subsection 7.1), leaving us with the parametric values $\Xi_1 = \{\xi_1^1 = 0, \xi_2^1, \dots, \xi_n^1 = 1\}$. The objective of the algorithm is to select parametric values of $\Xi_2 = \{\xi_1^2 = 0, \xi_2^2, \dots, \xi_m^2 = 1\}$ such that the aforementioned properties are satisfied. Given a threshold-distance $\epsilon > 0$ and assuming that a tuple $(\mathbf{p}_i^1, \mathbf{p}_j^2)$ with $\|\mathbf{p}_i^1 - \mathbf{p}_j^2\| \leq \epsilon$ exists, let

$$(n_0, m_0) = \min_{(i,j) \in \{1, \dots, n\} \times \{1, \dots, m\}} \|\mathbf{p}_i^1 - \mathbf{p}_j^2\|. \quad (40)$$

After (n_0, m_0) has been found, we set $\xi_{m_0}^2 = \xi_{n_0}^1$. We continue in the same fashion on the two subsets of $P_1 \times P_2$ with lower and higher indices as (n_0, m_0) , respectively. In practice, we usually remove a few points from the resulting subsets such that (in terms of cord-length) two matched tuples of points have a predefined minimum distance with respect to one

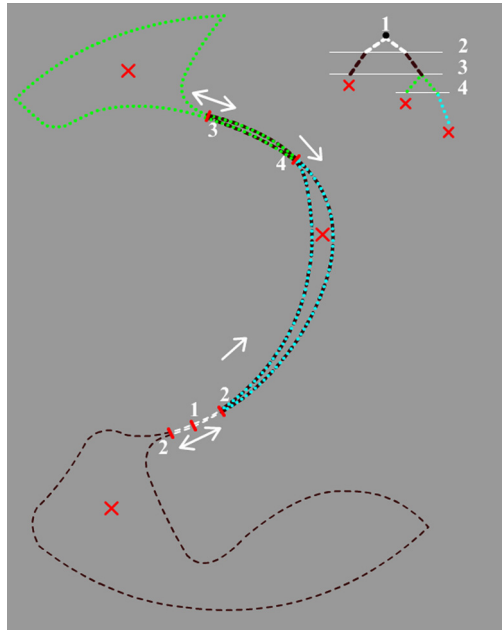


Fig. 10. The matching that the algorithm yields.

another. This process is repeated in a hierarchical fashion until no more points that are eligible for matching are found. The procedure is illustrated in Fig. 10.

After the algorithm terminates, the set Ξ_2 is partially known. It remains to determine the values of the ξ_i^2 with unmatched index. Let $\mathcal{I}_2^m = \{i \in \mathbb{N} \mid \xi_i^2 \text{ is known}\}$. It is possible to assign a value to the remaining ξ_i^2 by carrying out a spline-interpolation over the set of tuples $T_m^2 = \{(i, \xi_i^2) \mid i \in \mathcal{I}_2^m\}$. For this approach we recommend a monotone cubic interpolation (Fritsch and Carlson, 1980).

Another possibility of assigning values to the $\{\xi_i^2 \mid i \notin \mathcal{I}_2^m\}$ is by performing cord-length parameterizations between the consecutive tuples $t \in T_m^2$. Let $t_i = (\alpha, \xi_\alpha^2)$ and $t_{i+1} = (\beta, \xi_\beta^2)$ be two consecutive tuples from T_m^2 . The idea is to form a cord-length parameterization (as in Subsection 7.1) over the set $\{\mathbf{p}_i^2 \in P_2 \mid \alpha \leq i \leq \beta\}$. This will yield a set of abscissa $\tilde{\Xi}_i^2 = \{\tilde{\xi}_1 = 0, \tilde{\xi}_2, \dots, \tilde{\xi}_{\beta-\alpha+1} = 1\} \subset [0, 1]$. The idea is to manipulate this set as follows

$$\forall \tilde{\xi}_i \in \tilde{\Xi}_i^2 : \tilde{\xi}_i \implies \xi_\alpha + \tilde{\xi}_i(\xi_\beta - \xi_\alpha). \tag{41}$$

The sets $\tilde{\Xi}_i^2$ can be combined into the monotone increasing ordered set Ξ_2 by taking the set union (removing duplicates)

$$\Xi_2 = \bigcup_i \tilde{\Xi}_i^2. \tag{42}$$

After completion of the algorithm, Ξ_1 and Ξ_2 are utilized to parameterize and approximate the contours as discussed in Section 5.

Fig. 11 shows the geometry corresponding to the contour from Fig. 10 with cord-length parameterization and constrained cord-length parameterization. The plot of the Jacobian determinant reveals that the cord length parameterized geometry fails to be bijective, whereas bijectivity is achieved with the constrained approach.

A zoom-in onto the narrow gap of the geometry reveals the properties of the grid lines resulting from the unconstrained as well as the constrained approach. The grid lines in the unconstrained case have to be deemed unsatisfactory, whereas the constrained approach yields an outcome that can be considered superior both computationally (bijectiveness is achieved) and numerically (the grid lines should be orthogonal to the boundary) (Fig. 12). Let us mention that the B-Spline basis Σ utilized to construct the mapping is comprised of 2366 functions in the unconstrained case, whereas the basis resulting from the constrained case contains 3952 basis functions. This is likely due to the parametric velocity not being (approximately) constant over the parametric interval $[0, 1]$, requiring a locally higher density of basis functions to reach the approximation tolerance. The drawback is that the resulting computational costs are larger, defect correction (see Subsection 8.2), however, is not required and the overall numerical quality of the resulting mapping is vastly superior.

In our experience, the parametric properties of the boundary contours are as important for the quality of the resulting mapping as the meshing technique employed. Ideally, the contours should be so well-parameterized that the usage of transfinite-interpolation alone suffices to acquire a folding-free mapping. In practice, however, this is seldom achieved. The

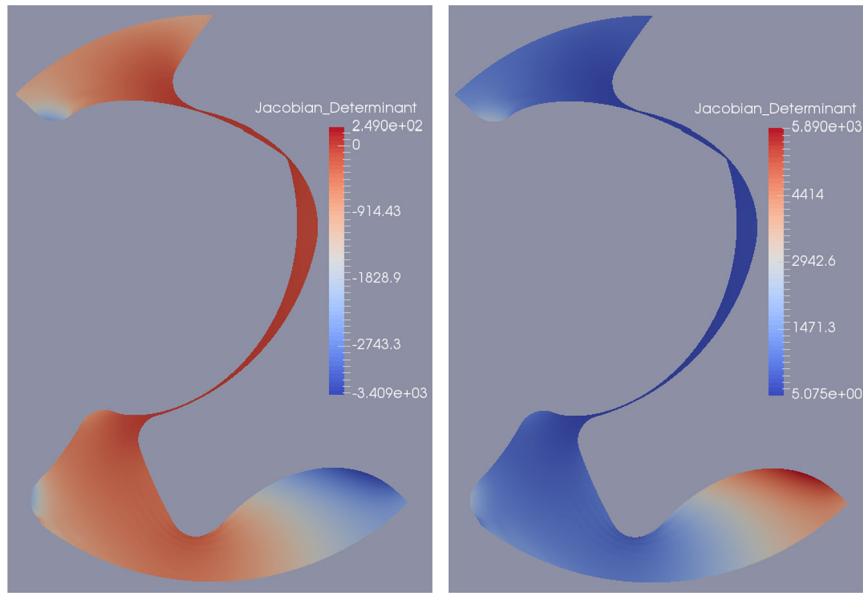


Fig. 11. The resulting mapping without (left) and with (right) constrained cord-length parameterization.

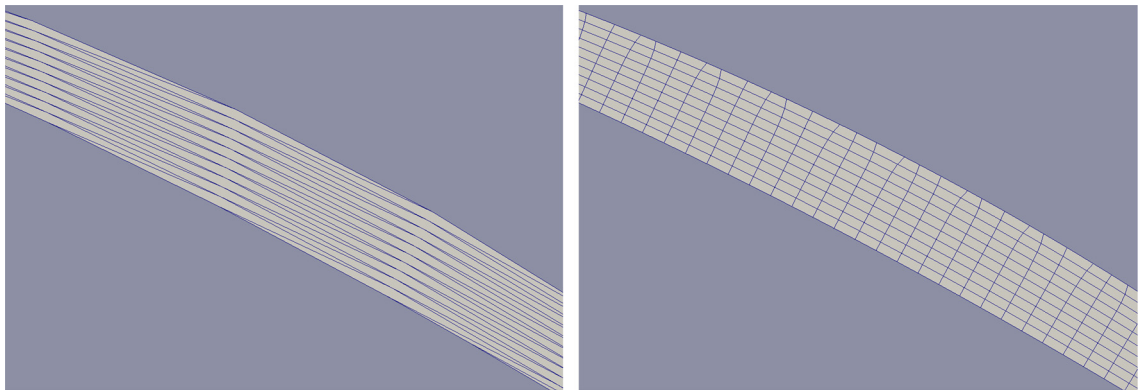


Fig. 12. A zoom in on the narrow part of the geometry shows its grid lines before and after parameterization.

reparameterization techniques employed depend on the application and most applications require a custom-made reparameterization procedure. The technique discussed in this subsection yields the best results in a wide range of applications. Since it operates on a discrete point-cloud, however, it is an inherently discrete process and the resulting reparameterization cannot be a continuous function of the input shape parameters α which is, however, a desired property in many applications.

8. Post-processing techniques

Upon convergence, the mapping operator $\mathbf{x}(\xi, \eta)$ is passed on to the post-processor. Due to the approximate nature of the variational approach, numerical errors can compromise the theoretically predicted bijection-property of the exact solution of the governing differential equation. The task of the post-processor is to detect defects caused by numerical errors. Defects manifest themselves in sign-changes in the determinant of the Jacobian-matrix of the mapping

$$\det[J] = \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}. \quad (43)$$

8.1. Defect detection

Assuming the topological properties of the mapping are orientation-preserving, the objective of the post-processor is thus to detect sign changes in $\det[J]$.

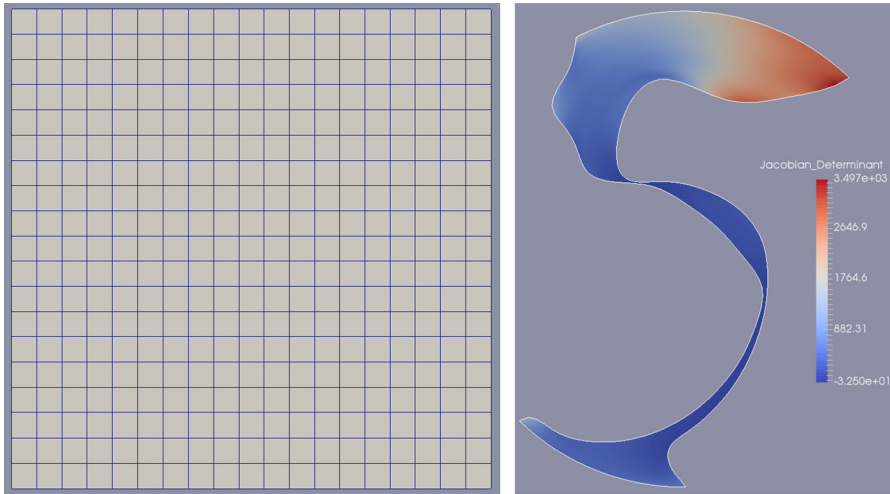


Fig. 13. A challenging geometry (right) and the corresponding segmentation of $\hat{\Omega}$ into elements (left). On first glance the parameterization seems fine, the plot of the Jacobian determinant, however, reveals that it is defective.

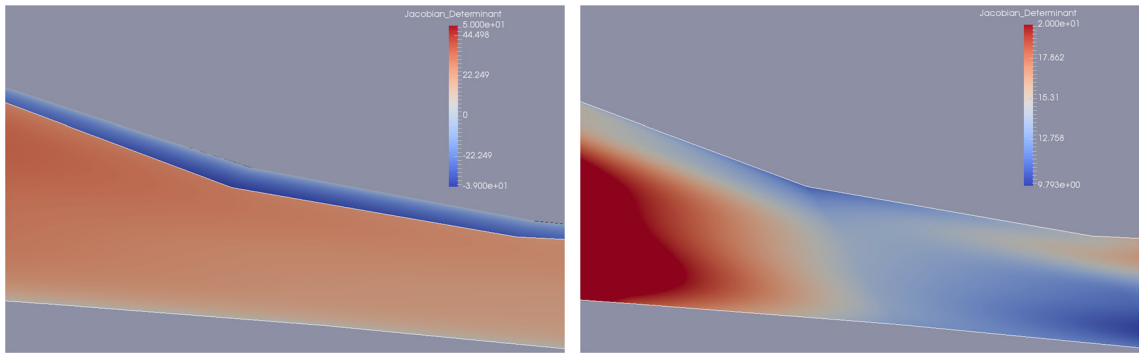


Fig. 14. Defective part of the geometry (left) and the same part after defect correction (right).

In practice, this is accomplished by projecting $\det[J]$ onto a suitable spline-space that contains it as in Gravesen et al. (2012) and require that all the projection weights d_i are positive. Thanks to the positivity of spline functions, this is a sufficient condition for bijectivity but not a necessary one. Heuristically, we have observed that $\det[J] > 0$ implies $d_i > 0$ in nearly all cases, which is why we deem a mapping analysis-suitable whenever $d_i > 0$.

8.2. Defect correction

After the above procedure has completed either the mapping is deemed analysis-suitable or a defect has been detected. If a defect has been detected the parameterization is locally refined in a small region centered around the region where a defect is located and another level $m + 1$ is added to the hierarchy. The defective mapping is prolonged to the refined basis and passed back to the solver, serving as an initial guess. The variational equations are solved again and this process is repeated until no defects are detected. We have often successfully corrected defects while freezing weights corresponding to basis functions located sufficiently far away from the location of the defects. With this approach, a lower amount of unknowns has to be recomputed which makes the approach computationally less expensive.

As an example, we consider the geometry depicted in Fig. 13. A projection of $\det[J]$ onto the spline-space that contains it reveals that a defect is located at the upper part of the narrow gap of the geometry. A zoom in onto this part in Fig. 14 (left) shows the geometry before defect correction and after (right). Fig. 15 depicts the corresponding refinements in the computational domain.

9. Including C^0 continuities

As discussed in Subsection 4.1, the usage of arbitrary bases (in particular bases with C^0 -continuities) can be computationally impractical (requiring bijective initial guesses), which is why we established that the basis used for the variational approach should be at least globally C^1 -continuous. In certain settings, however, a steep angle in the boundary contour is best resolved by the usage of a basis that allows for a C^0 -continuity there. A possible way to incorporate C^0 continuities

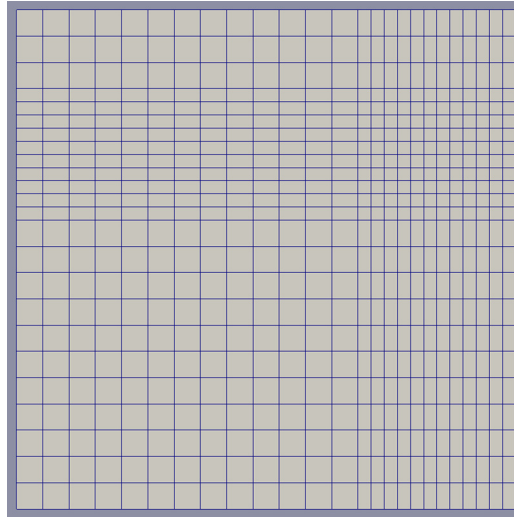


Fig. 15. The computational domain after refinement.

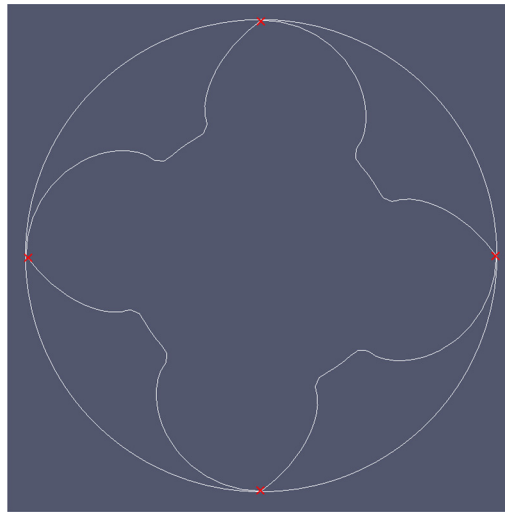


Fig. 16. Target contours with C^0 -continuities.

is to perform a segmentation of the geometry and compute parameterizations patch-wise. It is, however, not always evident how to segment and we are interested in generating parameterizations with a minimum amount of patches. Fig. 16 will serve as an example geometry for the discussion that follows. The depicted contours of this O-type geometry already represent the least-squares projection onto a $p = 3$ B-spline basis. Knots have been added in the parametric domain that correspond to the crosses in the Figure. They have been repeated thrice in order to yield a (local) C^0 -continuity. The basis is thus not compatible with the variational equations (8). The idea is to lower the knot-multiplicity by one wherever it exceeds $p - 1$ and perform a least-squares projection of the given contour approximation onto the newly formed basis. Henceforth we shall refer to the two bases formed in this way by Σ^0 and Σ^1 . Fig. 17 shows a zoomed-in picture of the C^0 -continuity before and after smoothing. We proceed by solving the variational equations corresponding to the smoothed contours and to manipulate the resulting mapping so that it can serve as a bijective initial guess to (16). With Subsection 6.3 in mind, we prolong the solution of the smoothed problem to the basis Σ^0 and the discrepancy between the boundary control points is imposed as a Dirichlet boundary condition to a linear elasticity problem. Should bijectivity be retained we solve (16) with the given initial guess to further improve the parametric properties. In practice we have often successfully applied this technique thereby observing that its success rate is strongly correlated to the quality of the parameterization of the boundary contours (see Section 7).

Should the initial guess constructed upon completion of the linear elasticity problem fail to be bijective, we can perform the above steps with a Dirichlet boundary condition that is dampened by a factor $\alpha < 1$. This process is repeated until the target contours are attained.

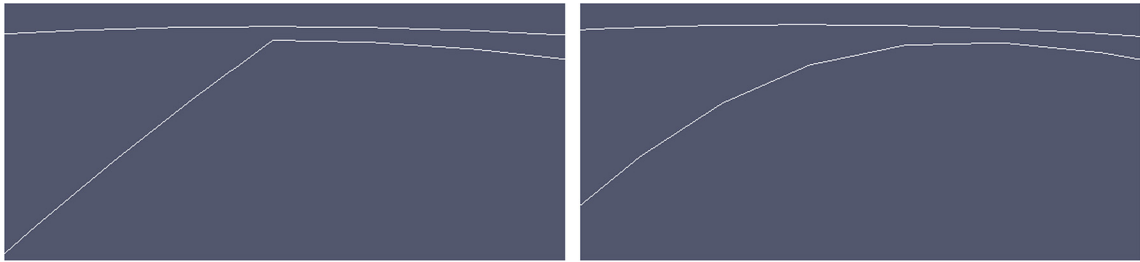


Fig. 17. Part of the boundary contours before and after smoothing.

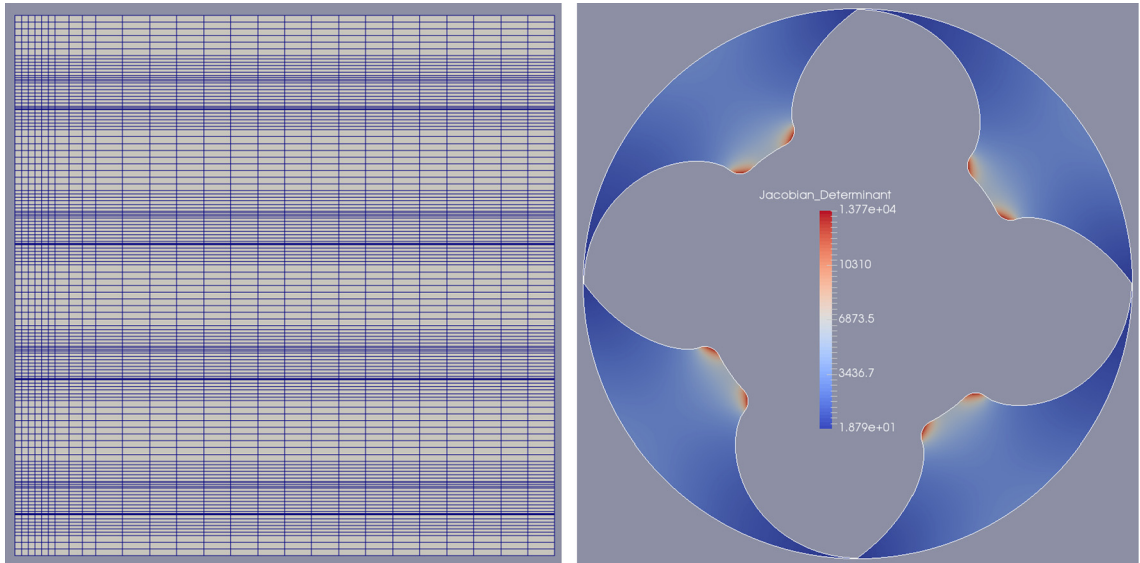


Fig. 18. The mapping corresponding to the contours from Fig. 16 acquired by performing above steps.

Finally, we have often successfully applied this technique while only recomputing a subset of DOFs in the vicinity of the boundary control points that correspond to the C^0 -continuities. This makes this approach much more feasible from a computational point of view while it is challenging to decide which DOFs to recompute and which to freeze.

Fig. 18 depicts the geometry that corresponds to the contours from Fig. 16.

10. Comparison to other methods

In this section we will compare the proposed method to some of the established unconstrained methods that are known to yield bijective parameterizations in a wide range of applications. Furthermore, we shall investigate whether it can reliably produce feasible initial guesses for the constrained methods. Quality assessment shall be based on whether or not bijectivity is achieved, the amount of iterations required until convergence and the quality of the solution of a benchmark problem. As a benchmark for grid quality, we will employ the following benchmark problem

$$\begin{aligned}
 -\Delta u &= -\Delta u^*, \quad \text{in } \Omega \\
 u|_{\partial\Omega} &= u^*, \quad \text{on } \partial\Omega,
 \end{aligned}
 \tag{44}$$

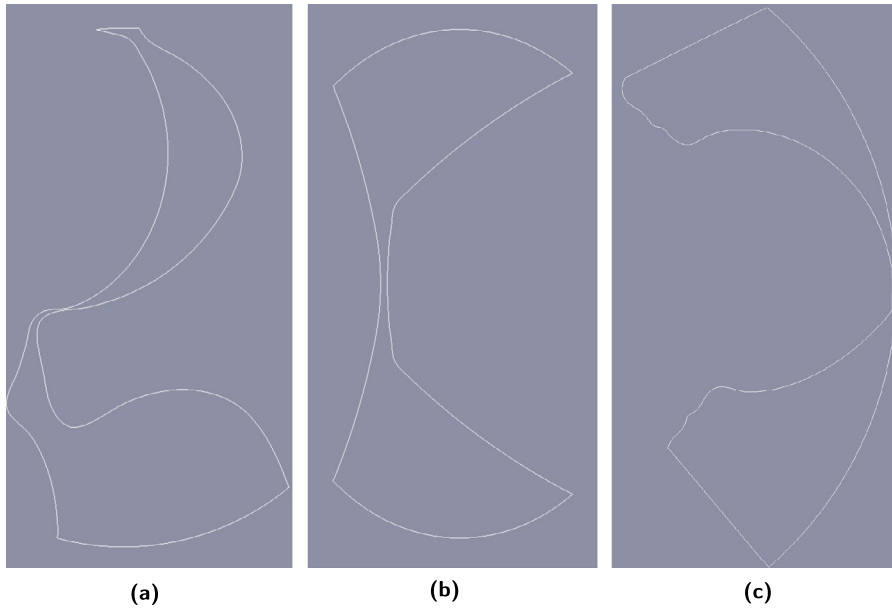
where $u^* = \sin\left(\frac{\pi x}{a_1}\right) \sin\left(\frac{\pi y}{a_2}\right)$, with a_1, a_2 chosen based on the characteristic length scale of the geometry. Note that the exact solution of (44) is given by $u = u^*$. We shall solve the weak counterpart of (44) with the parameterization \mathbf{x} to compute an approximation $u^h(\mathbf{x})$ of u . To assess the quality of the solution, we utilize

$$Q(\mathbf{x}) = \sqrt{\frac{\int_{\Omega} (u^h(\mathbf{x}) - u^*)^2 d\mathbf{x}}{\int_{\Omega} (u^*)^2 d\mathbf{x}}}.
 \tag{45}$$

Table 1

Results corresponding to Fig. 19(a), 19(b) and 19(c).

(a)	IT	BIJ	$\frac{Q(\mathbf{x})}{Q(\mathbf{x}_G)}$	W	(b)	IT	BIJ	$\frac{Q(\mathbf{x})}{Q(\mathbf{x}_G)}$	W	(c)	IT	BIJ	$\frac{Q(\mathbf{x})}{Q(\mathbf{x}_G)}$	W
G	4	yes	1	38.7	G	4	yes	1	18.8	G	4	yes	1	31.3
VH	18	no	x	∞	VH	38	yes	0.96	18.9	VH	26	yes	1.08	31.7

**Fig. 19.** Benchmark problems for the comparison with VHM.

10.1. Variational harmonic method

The ‘Variational harmonic method’ (Xu et al., 2013) (VHM) is based on the minimization of

$$\text{VH}(\mathbf{c}) = \int_{\hat{\Omega}} L(x, y, x)^2 + L(x, y, y)^2 + \lambda_1 \left(\|\mathbf{x}_{\xi\xi}\|^2 + \|\mathbf{x}_{\eta\eta}\|^2 + \|\mathbf{x}_{\xi\eta}\|^2 \right) + \lambda_2 \left(g_{11}^2 + g_{22}^2 \right) d\xi, \quad (46)$$

which constitutes a linear combination of the harmonic energy, uniformity and area orthogonality functionals. For $\lambda_1 = \lambda_2 = 0$, the approach is comparable to ours, which is why in the following, we shall set $\lambda_1 = \lambda_2 = 0$. We minimize (46) by applying the principles from Subsection 6.1 to $\frac{\partial \text{VH}(\mathbf{c})}{\partial \mathbf{c}}$. In the following we will compare the amount of iterations ‘IT’ and the ratio of the grid quality estimates (45), where \mathbf{x}_G denotes the Galerkin-solution. ‘BIJ’ indicates whether the geometry succeeds in being bijective. Finally, by ‘W’ we denote the evaluation of the winslow functional (16) both of which they attempt to minimize.

Table 1 shows the results for the geometries from Fig. 19.

10.2. Quasi-conformal method

In this section, we compare various geometries acquired with our method to geometries acquired with the method from Nian and Chen (2016). Unlike in the article, we minimize the functional

$$T(\mathbf{x}) = \int_{\hat{\Omega}} \left(f - k^2 \right)^2 d\xi, \quad (47)$$

with

$$f = \frac{(x_\xi - y_\eta)^2 + (y_\xi + x_\eta)^2}{(x_\xi + y_\eta)^2 + (y_\xi - x_\eta)^2} \quad (48)$$

over $\mathbf{x}(\mathbf{c}), k$ using the optimization environment IPOPT (Wächter and Biegler, 2006). Since we are using different solution methods, in the following, we shall only compare the grid quality. Table 2 contains the results of the grid quality comparison for the various test cases from Fig. 20.

Table 2
Grid quality for the various geometries from Fig. 20.

Fig. 20	(a)	(b)	(c)
$Q(\mathbf{x})/Q(\mathbf{x}_G)$	21.0	0.89	0.77

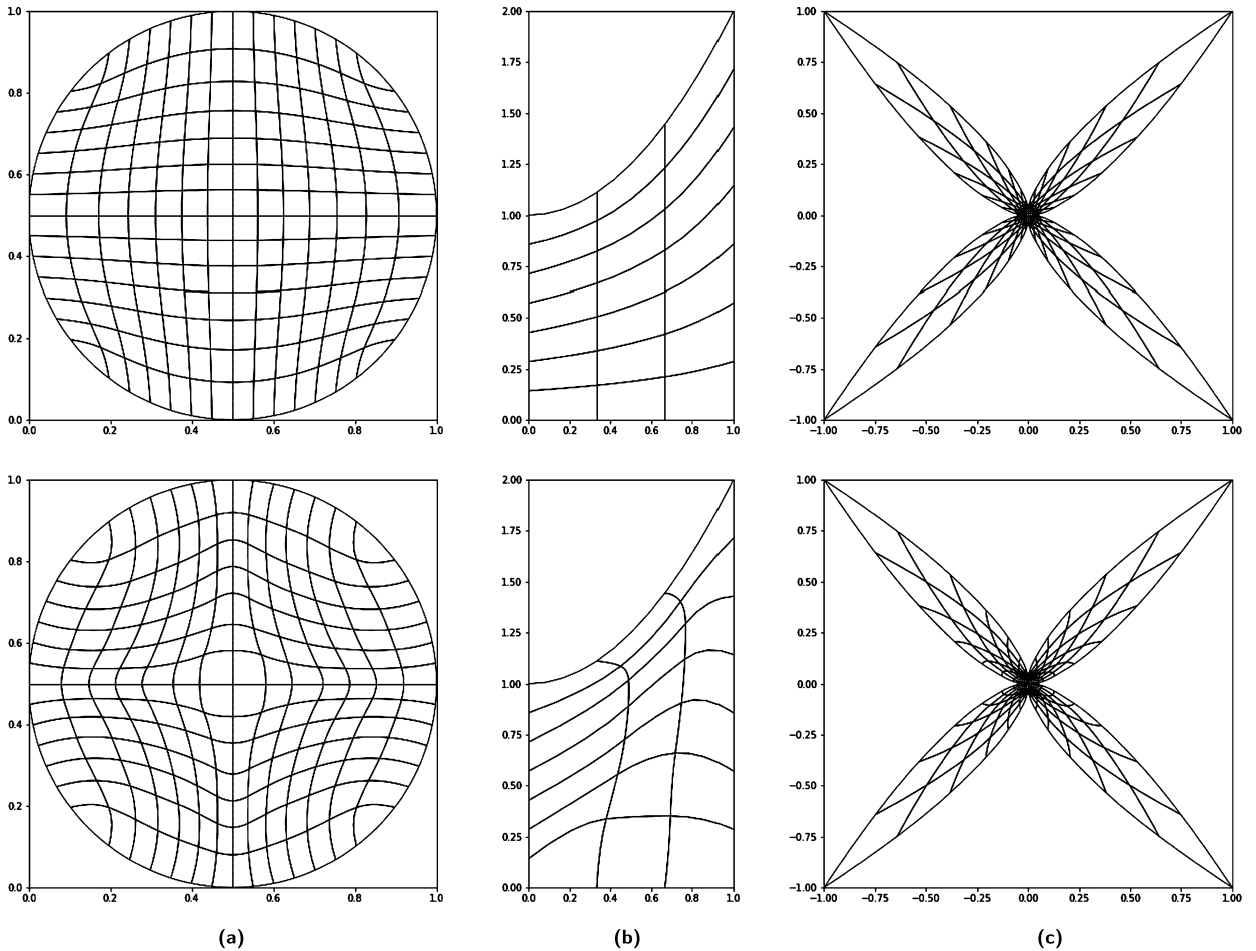


Fig. 20. Results of various geometries with our approach (above) and the minimization of (47) (below).

10.3. Application to constrained methods

In Gravesen et al. (2012), the inner control points chosen based upon the optimization of various quality functionals under the constraint that the projection of $\det[J]$ onto a spline space in which it is contained only carries positive weights. A major hurdle is the finding of a feasible initial guess. Let $\mathbf{d}(\mathbf{c})$ be the vector of projection weights, a feasible initial guess is computed as follows

$$\begin{aligned} & \max_{\mathbf{c}} z, \\ & \text{s.t. } \mathbf{d}(\mathbf{c}) \geq \mathbf{1}, \end{aligned} \tag{49}$$

where $\mathbf{1}$ is a vector of ones. As suggested in Gravesen et al. (2012), we use IPOPT to tackle (49). We are considering the test cases depicted from Fig. 21. The solvers are initialized with the same (unfeasible) initial guess.

Table 3 shows the results, where the letter ‘G’ stands for Galerkin (our approach) and ‘Z’ for the approach from (49). ‘L’, ‘W’ and ‘AO’ stand for the Liao, Winslow and area orthogonality functionals, respectively.

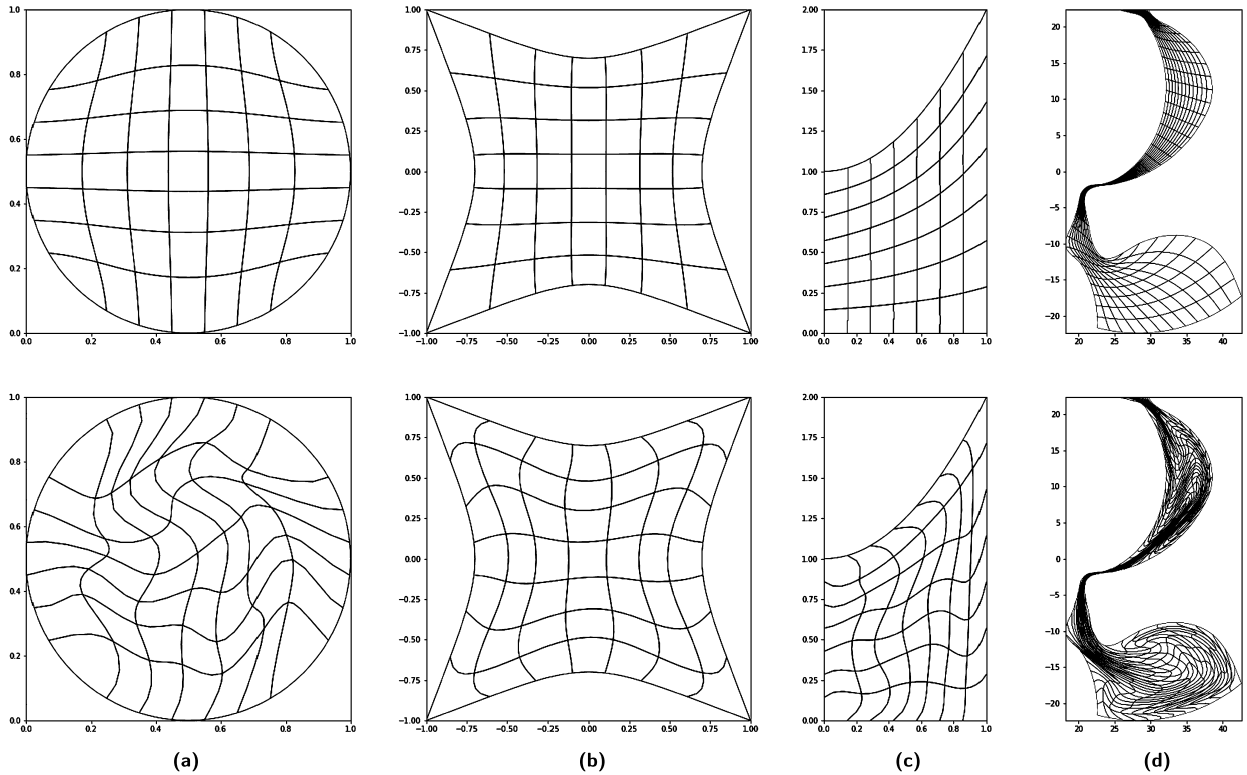


Fig. 21. Results of various geometries with our approach (above) and the minimization of (47) (below).

Table 3

Comparison of the iteration number and initial grid quality with respect to the various quality functionals for the two initialization methods.

	$\Pi(G)$	$\Pi(Z)$	$L(G)$	$L(Z)$	$W(G)$	$W(Z)$	$AO(G)$	$AO(Z)$
(a)	25	27	1.92	6.02	2.23	3.36	0.70	0.99
(b)	10	18	19.03	24.46	2.05	2.34	8.47	7.92
(c)	5	19	10.66	17.17	2.40	3.01	3.06	3.60
(d)	5	27	1.25×10^7	2.88×10^8	38.73	70.25	5.54×10^5	5.76×10^6

10.4. Conclusion

The numerical experiments from Subsection 10.1 show that the parameterizations produced by VHM with $\lambda_1 = \lambda_2 = 0$ and our method are of similar quality. Our method succeeds in producing a bijective mapping in all three test cases whereas the VHM fails to do so in one. The evaluations of the Winslow functional are close, too. Preliminary numerical comparisons indicate that the proposed Galerkin approach (combined with a Newton solver) outperforms the VHM by a factor of ~ 10 in nonlinear iterations. A more in depth analysis is of course needed to check if other nonlinear solvers could speed up the VHM. It is likely that the Newton-approach suggested in this paper is not a suitable method for the VHM.

Tackling (47) with the approach from Subsection 6.1 generally did not lead to convergence, which is why we had to fall back on IPOPT. We observed that the amount of iterations needed until convergence significantly exceeded those required in our method. As the computational approach suggested in Nian and Chen (2016) is custom-made for this problem, it is likely that the quality of the resulting parameterizations will be better, especially since similar methods (such as Lui et al., 2014 and Weber et al., 2012) can produce quite different results, despite being based on the same principle. Furthermore, the amount of iterations may be reduced significantly. Preliminary numerical experiments from Subsection 10.2, however, show that the quasi-conformal approach produces parameterizations of higher quality in two out of the three test cases, despite the relative simplicity of the approach based on IPOPT. This may indicate that methods based on this principle can produce mappings of similar or better quality than those based on EGG, without the need for constraining.

Finally, the numerical experiments from Subsection 10.3 indicate that our approach may serve as an excellent way to generate feasible initial guesses for the constrained methods. This is especially true when comparing the iteration number. Since IPOPT requires the computation of the constraint-gradient during each iteration, the complexity of such an iteration should be comparable to one Newton-iteration. From this, we may conclude that our method can produce feasible initial guesses with less computational effort. Furthermore, the initial guesses produce smaller values in the evaluations of all the

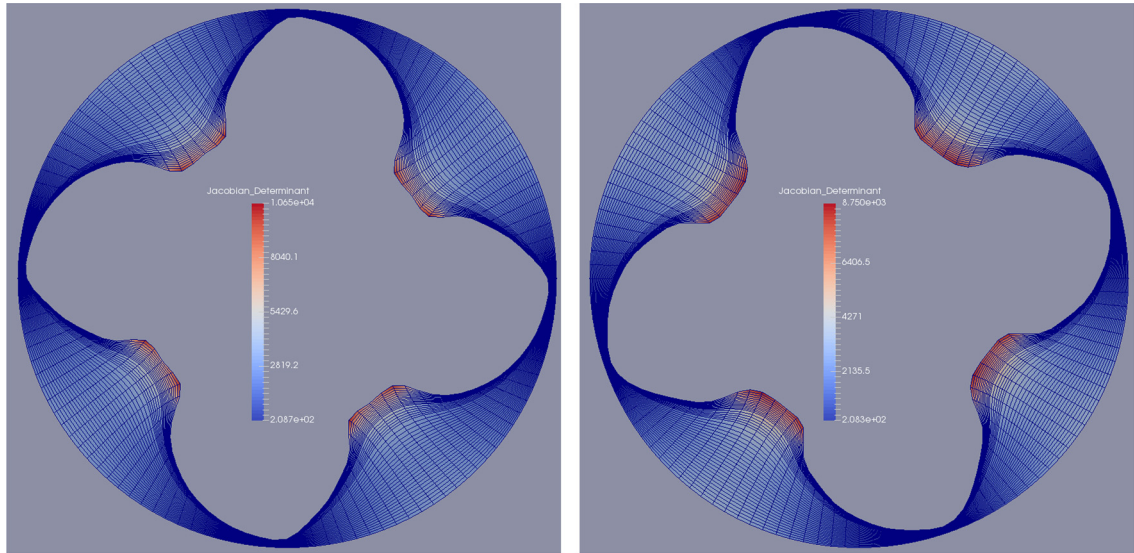


Fig. 22. The sliding O-grid geometry after 0 and 125 iterations.

quality functionals considered in Gravesen et al. (2012) in all cases but one. This may serve as to reduce the typical iteration number of ~ 30 that is required with an initialization based on (49) (especially when the Winslow-functional is chosen).

We can conclude that our algorithm, combined with the solution techniques from Subsections 6.1 and 6.2, constitutes a computationally effective and reliable way to generate analysis-suitable parameterizations of decent quality. Due to being based on EGG only, it is inherently less flexible than methods based on the minimization of other quality functionals. More flexibility, however, may be achieved by combining it with one of the constrained optimization approaches.

11. Applications

In this section, we will discuss possible applications of the algorithm within an IgA-setting and within classical meshing problems.

11.1. Time-dependent geometries and volumetric applications

We are again considering the geometry from Fig. 18.

Assuming a flow-simulation of the interior is carried out utilizing an IgA-approach in the spatial components and finite-difference (with fixed time-step) in the temporal component, part of the computational approach is the generation of an analysis-suitable mapping for the geometry at the current time-instance. For the O-grid, the northern and southern boundaries are disregarded and a periodic knot-vector is utilized in the η -direction. Having to perform the all the steps from Section 6 at each time-instance is computationally expensive which is why one should look for better options. Assuming the grid to be fixated at the eastern boundary (casing) and sliding along the western boundary (rotor), we can construct high-quality initial guesses by extrapolating the mappings from previous time-steps to the current time-step. Assuming Δt corresponds to some fixed angular increment $\Delta\theta$, after each iteration we act with the canonical rotation matrix on the inner contour-interpolation $\mathbf{c}_r(\eta)$ and reparameterize to let $\eta = 0$ coincide with $\mathbf{c}_r(\eta)_y = 0$. After the first mapping $\mathbf{x}_1(\xi, \eta)$ has been computed utilizing the principles from Section 6, the internal DOFs of \mathbf{x}_1 are utilized as an initial guess for those of \mathbf{x}_2 . This constitutes a zeroth order extrapolation. Once more mappings become available, higher-order extrapolations can be constructed. Here, we restrict ourselves to a sequence of length six, i.e. we construct an initial guess for \mathbf{x}_n from $\{\mathbf{x}_{n-6}, \dots, \mathbf{x}_{n-1}\}$ or a subset thereof. A sequence of length l is utilized to construct a $(l - 1)^{\text{th}}$ -order spline extrapolation for each internal DOF which is then evaluated at the current time-instance and utilized as an initial guess. This operation is computationally efficient since each DOF can be treated independently. Note also that this operation is compatible with variable time steps.

Fig. 22 shows a realization of the sliding O-grid algorithm from Section 11.1 with $\Delta\theta = 0.003$ after 0 and 125 iterations. In practice, the truncated Newton iteration converges after only one iteration once enough grids for a 5th-order extrapolation are available (the initial residual is of order 10^{-4} in the Euclidean norm and the convergence threshold is 10^{-7}). This is a remarkable result since it implies that EGG is on par with linear meshing techniques in terms of the computational complexity while yielding far superior results.

As an example of how in certain applications the planar EGG-approach can serve as a means to generate trivariate parameterizations, we replace the temporal axis t by the z -axis and perform first-order transfinite interpolation on the

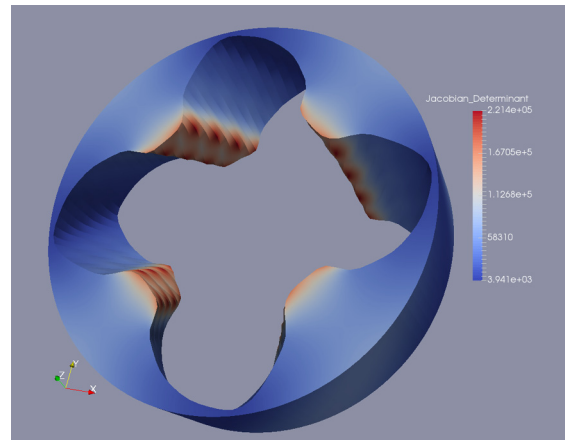


Fig. 23. First order interpolation between the first 126 cross-sections.

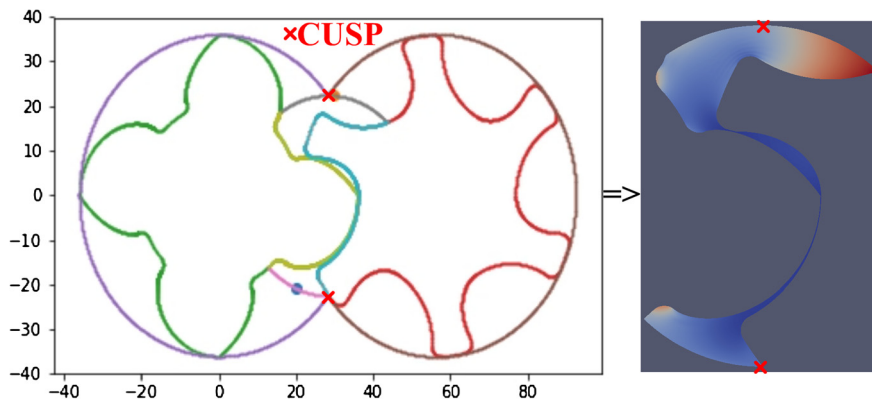


Fig. 24. Contours of a challenging geometry that can only be parameterized using two or more patches.

$\{\mathbf{x}_1, \dots, \mathbf{x}_{126}\}$ in ascending order. Fig. 23 depicts the resulting volumetric geometry. The positivity of the Jacobian determinant of the mapping reveals that the interpolation, too, is analysis-suitable. The non-smooth nature of the first-order interpolation, however, is clearly visible in the picture. Higher-order (and smooth) interpolation is possible but has not yet been implemented.

Should the geometry acquired via transfinite-interpolation fail to be bijective despite all its constituent slices \mathbf{x}_i being bijective, we need to add more slices wherever the defects are detected. Fortunately we can utilize our database of m planar slices to construct excellent initial guesses for the additional required slices via interpolation in z . For this, the boundary control points of each slice should be continuous in z and the slices, centered at their corresponding z -coordinate should not intersect, which usually can only be guaranteed for swept volumes like the one presented in this section. At this point we would like to mention that unlike in the bivariate case, trivariate EGG does not guarantee bijectivity of the resulting solution. In Wang and Qian (2014), similar to the bivariate case from Gravesen et al. (2012) and Xu et al. (2011), this is solved by the constrained minimization of a quality functional that enforces bijectivity of the resulting parameterization. Again, a major hurdle is finding a bijective initial guess. The divide-and-conquer approach from this section may serve as a means to quickly generate feasible initial guesses. For an algorithm that mainly relies on heuristics but can produce good results in swept volume applications, see (Aigner et al., 2009).

11.2. Multi-patch parameterizations

Fig. 24 (left) shows the contours of a challenging geometry. The objective is to generate a parameterization of the interior between the two rotors shown and the casing represented by two circular arcs (meeting at the CUSP points) utilizing two or more mappings. The figure also shows two smaller circular arcs, that together with parts of the left and right rotors constitute the contours of a ‘separator’ geometry whose single-patch mapping is depicted on the right. The geometry was generated utilizing the Newton-approach from Subsection 6.1 in conjunction with the reparameterization technique from Subsection 7.2. As a next step we generate a separating line connecting the two CUSP-points of the casing by selecting the two points in the computational domain that correspond to them and connecting them via a curve. The most straight-forward choice is to simply connect the two points by a straight line, which already yields decent results. The

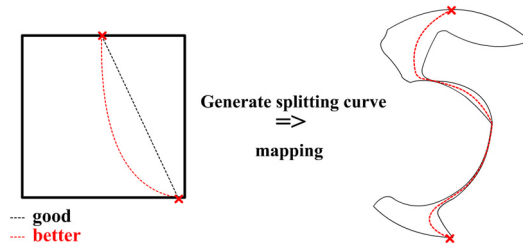


Fig. 25. Generating a high-quality splitting-curve utilizing the bijective mapping.

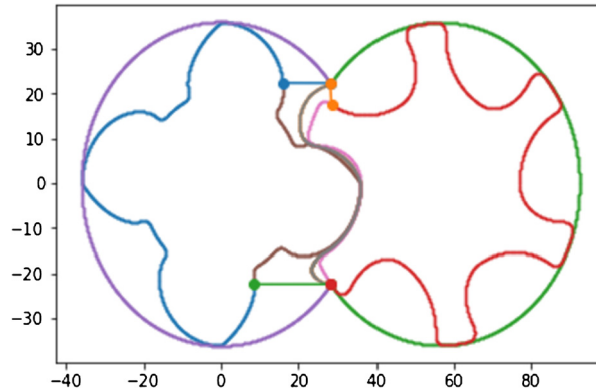


Fig. 26. Four-patch target geometry.

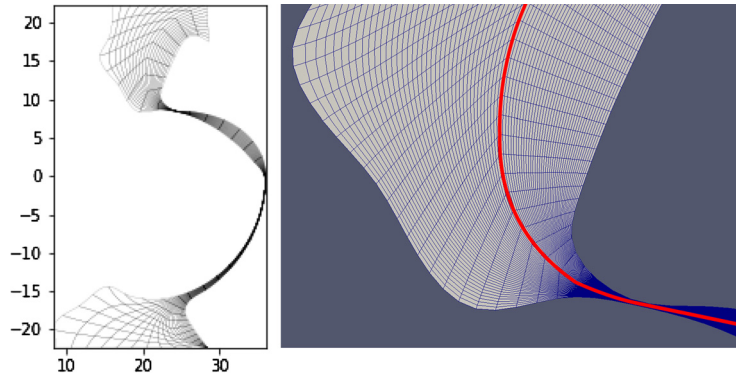


Fig. 27. Two-patch mapping for the middle part of the geometry depicted in Fig. 26 (left). The right picture shows the segment at which the two geometries meet. It is clearly seen that the two mappings are conforming but that they exhibit a C^0 -continuity where the mappings meet.

bijective mapping, however, also allows for a more flexible choice. In this case we traverse the computational domain in such a way that the geometry is split most-evenly into two parts by the narrow gaps of the separator. The procedure is illustrated in Fig. 25. The splitting-curve is exported as a point-cloud and utilized to split the target geometry into two equal parts. As a next step, two geometry contours are formed using the splitting-curve, parts of the left and the right rotors and connecting line segments (see Fig. 26).

Mappings for the geometries, fenced off by these contours, are generated using the same technique employed in the generation of the separator, whereby it has been made sure that both geometries utilize the same knot-vector in the η -direction. The two resulting mappings are depicted in Fig. 27 (left). In Fig. 27 (right) the red line shows the segment at which the two mappings meet. Since both mappings are the result of a basis with open knot-vectors, the shared boundary results in a C^0 -continuity. The idea is to combine the two mappings into one by combining the knot-vectors and letting the DOFs corresponding to the red splitting-curve float. As a next step, the steep angles that the elements make by the splitting-curve are smoothed by finding the root of system (14) which allows for C^0 -continuous bases but requires an bijective initial guess. As initial guess we simply use the initial state of the combined geometries (which is obviously bijective since both of its constituents are bijective). A zoom-in on the resulting mapping is shown in Fig. 28.

Finally, two C-type mappings are generated for the parts to the left and to the right of the center. The final geometry consisting of the mappings is depicted in Fig. 29.

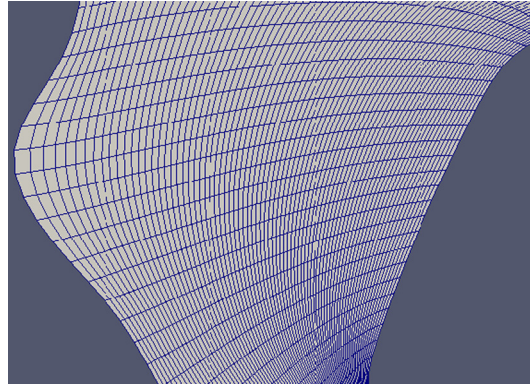


Fig. 28. Upon combination of the two mappings into one and remeshing, the steep angles are significantly smoothened.

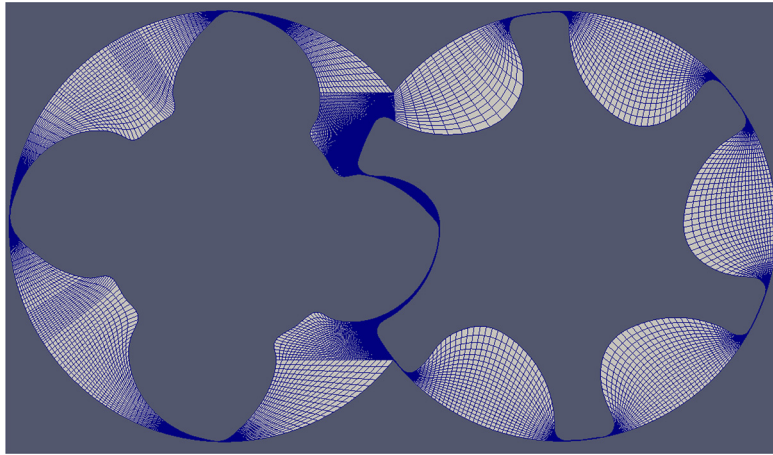


Fig. 29. Upon combination of the two mappings into one and remeshing, the steep angles are significantly smoothened.

The two C-grids and the middle-part are not conforming by the segment where they meet. The elements of the middle part are twice as dense as the elements of the two C-type grids. They can easily be made conforming by performing one uniform refinement on the knot-vector corresponding to the ξ -direction of the C-grids. A last possible step is to split the middle-part into two mappings again and combine the left and right parts with the left and right C-grid respectively to form two O-grids. The steep angles by the hand-made separating line can then be removed by finding the root of the system (14) in a way similar to the approach used in the removal of the splitting-curve.

11.3. Applications within classical meshing

Classical finite-difference or finite-volume settings, typically utilize structured grids comprised of linear elements. The classical counterpart of EGG is based on a finite-difference approach of the governing equations (see Thompson et al., 1998). Even though the approach proposed in this article is based on curved as opposed to linear elements, the mapping operator can be utilized to generate linear meshes of any desired accuracy.

Assuming the solver has converged to a bijective mapping while being within a predefined tolerance-band of the exact boundary contours $\partial\bar{\Omega}$, the curved mapping $\mathbf{x}(\xi, \eta)$ can be turned back into a linear mesh by performing a large number of function evaluations. Thanks to the flexibility of spline-functions, the tolerance of the boundary approximation tends to be attained with fewer elements than in the linear case which is potentially less computationally expensive. The desired accuracy of the contour approximation of the linear mesh is then easily attained by performing an adequate amount of function evaluations. The spacing of the evaluations can be locally tuned to reach the desired accuracy with less elements.

Another potential application of this principle is that of a tunable grid resolution. In many applications it is desirable to have a higher element density, for instance, along the boundary contours. As an example we again consider the familiar geometry from Fig. 30.

We form the sets $\Xi = \{0, 1/\Delta\xi, 2/\Delta\xi, \dots, 1\}$ with $\Delta\xi = 1/401$ and $\mathcal{H} = \{0, \eta'(\Delta\eta), \eta'(2\Delta\eta), \dots, 1\}$ with

$$\eta'(\eta) = \frac{1}{2} \left(\frac{\arctan\left(6\left(x - \frac{1}{2}\right)\right)}{\arctan(3)} + 1 \right) \quad (50)$$

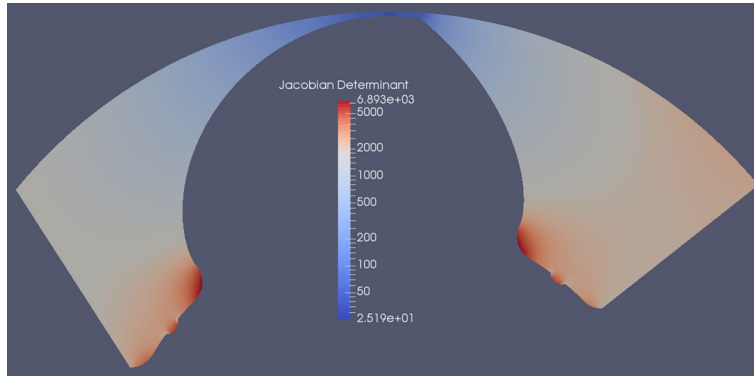


Fig. 30. The bijective geometry with curved elements.

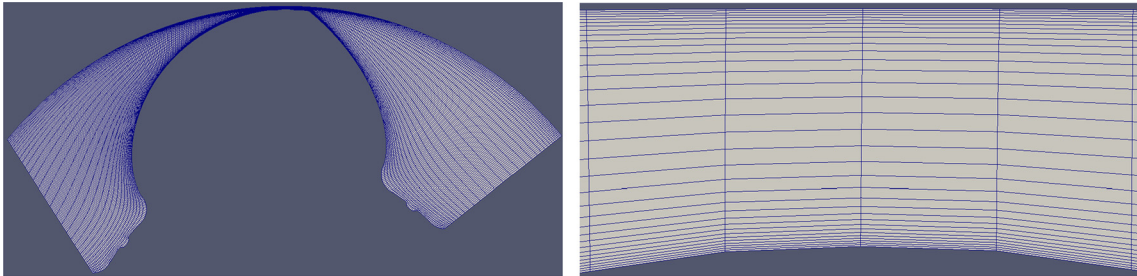


Fig. 31. A classical mesh acquired from the geometry depicted in Fig. 30 by performing a large number of function evaluations (left). The right picture shows a zoom-in on the narrow gap to show the boundary element distribution.

and $\Delta\eta = 1/35$. A linear mesh with boundary layers is acquired by performing function evaluations over the $(\xi, \eta) \in \Xi \times \mathcal{H}$ and connecting neighbouring nodes with edges. The resulting grid is depicted in Fig. 31.

The internal DOFs to generate the mapping from Fig. 30 amount to a total number of 14694, whereas the total amount of internal grid points is 26334. The mapping has been created with an extremely accurate boundary approximation, which is why the conversion to a classical mesh should be performed with an extremely large number of function evaluations (more than in the example given) to make the IgA-approach less computational expensive than a direct finite-difference based approach with the same amount of grid points. Initially, the large amount of DOFs to generate the mapping might seem unfeasible but considering that a classical mesh with an amount of elements that exceeds that of the mapping by several orders of magnitude can be generated with ease, it becomes apparent that the IgA-approach might be more feasible in this case.

Finally, we mention that the mapping can also be utilized to generate unstructured meshes. This is accomplished by performing (for instance) a triangulation in the computational domain and carrying out function evaluations in the vertices. The value of the Jacobian determinant in the vertices can in turn serve as a refinement criterion.

12. Outlook

In this article we presented an algorithm aimed at embedding the principles of EGG into an IgA-framework. This was accomplished by discretizing the governing equations utilizing a Galerkin-approach with $p \geq 2$ spline basis-functions and solving the resulting nonlinear system utilizing a truncated Newton-approach. Furthermore, we presented automatized reparameterization techniques to improve the parametric properties of challenging geometries like, for instance, geometries with extreme aspect ratios.

The hierarchical Newton-based approach proposed in this article greatly contributes to the overall stability of the algorithm. The algorithm reliably produces analysis-suitable single-patch parameterizations for a wide range of geometries and typically converges within 3–5 iterations.

Even though self-intersections due to numerical errors are uncommon, defects can be located on internal elements as a result of locally insufficient numerical resolution. So far, the basis is only chosen based upon a sufficiently accurate discretization of the boundary contours but not so as to prevent self-intersections. Currently, self-intersections are resolved by the defect-correction techniques from Section 8. At this point, we recommend against an approach with a hierarchical spline basis. Since it is difficult to predict which internal elements need to be hierarchically refined, from our experience, the amount of defect corrections required does not weigh up against the reduced amount of DOFs.

The strength of the *constrained cord-length* reparameterization technique introduced in Subsection 7.2 is evidenced by its successful application in the generation of parameterizations for geometries like the ones depicted in Figs. 8 and 11 as well as the multi-patch parameterization from Fig. 29. Unfortunately it has two major drawbacks:

- since one side has an (approximately) constant parametric velocity while the opposite side does not, the amount of DOFs required for an accurate boundary discretization will be dominated by the side with variable parametric velocity which implies that the cardinality of the basis required for an accurate resolution will generally exceed that of a purely arc-length based parameterization.
- The constrained cord-length technique has a discrete nature which is why it can neither be combined with methods that aim at reducing the computational complexity via extrapolation such as the sliding grid from 11.1 nor with swept volume parameterizations such as the one presented in the same subsection.

The aforementioned drawbacks imply that in most applications a custom-made reparameterization technique is required that results in a contour parameterization which is a continuous function of, for instance, the shape parameters. It would be convenient to possess a reparameterization technique that yields satisfactory results while not being inherently discrete.

Finally, despite its repeated successful application, the method for the inclusion of C^0 -continuities proposed in Section 9 is relatively involved. A computational approach that allows for C^0 -continuities while not requiring a bijective initial guess constitutes a possible topic for further research.

Acknowledgements

This project (MOTOR) has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 678727.

References

- Aigner, M., Heinrich, C., Jüttler, B., Pilgerstorfer, E., Simeon, B., Vuong, A.-V., 2009. Swept volume parameterization for isogeometric analysis. In: IMA Conference on the Mathematics of Surfaces. Springer, pp. 19–44.
- Azarenok, B.N., 2009. Generation of structured difference grids in two-dimensional nonconvex domains using mappings. *Comput. Math. Math. Phys.* 49 (5), 797–809.
- Becker, C., 2007. Strategien und Methoden zur Ausnutzung der High-Performance-Ressourcen moderner Rechnerarchitekturen für Finite-Element-Simulationen und ihre Realisierung in FEAST (Finite Element Analysis & Solution Tools). PhD thesis, Universität Dortmund, Logos Verlag, Berlin. ISBN 978-3-8325-1637-6. <http://www.logos-verlag.de/cgi-bin/buch?isbn=1637>.
- Coffey, T.S., Kelley, C., Keyes, D.E., 2003. Pseudotransient continuation and differential-algebraic equations. *SIAM J. Sci. Comput.* 25 (2), 553–569.
- Coons, S.A., 1967. Surfaces for Computer-Aided Design of Space Forms. Technical report, DTIC Document.
- Dierckx, P., 1995. *Curve and Surface Fitting with Splines*. Oxford University Press.
- Falini, A., Špeh, J., Jüttler, B., 2015. Planar domain parameterization with thb-splines. *Comput. Aided Geom. Des.* 35, 95–108.
- Falk, R., 1939. Finite element methods for linear elasticity. 01 2008.
- Fritsch, F.N., Carlson, R.E., 1980. Monotone piecewise cubic interpolation. *SIAM J. Numer. Anal.* 17 (2), 238–246.
- Giannelli, C., Jüttler, B., Thb-splines, H. Speleers, 2012. The truncated basis for hierarchical splines. *Comput. Aided Geom. Des.* 29 (7), 485–498.
- Gravesen, J., Evgrafov, A., Nguyen, D.-M., Nørtoft, P., 2012. Planar parametrization in isogeometric analysis. In: International Conference on Mathematical Methods for Curves and Surfaces. Springer, pp. 189–212.
- Hughes, T.J., Cottrell, J.A., Bazilevs, Y., 2005. Isogeometric analysis: cad, finite elements, nurbs, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Eng.* 194 (39), 4135–4195.
- Kelley, C.T., Keyes, D.E., 1998. Convergence analysis of pseudo-transient continuation. *SIAM J. Numer. Anal.* 35 (2), 508–523.
- Lamby, P., Brakhage, K., 2007. Elliptic grid generation by b-spline collocation. In: Proceedings of the 10th International Conference on Numerical Grid Generation in Computational Field Simulations. FORTH, Crete, Greece.
- Lipton, S., Evans, J.A., Bazilevs, Y., Elguedj, T., Hughes, T.J., 2010. Robustness of isogeometric structural discretizations under severe mesh distortion. *Comput. Methods Appl. Mech. Eng.* 199 (5), 357–373.
- Lui, L.M., Lam, K.C., Yau, S.-T., Gu, X., 2014. Teichmüller mapping (t-map) and its applications to landmark matching registration. *SIAM J. Imaging Sci.* 7 (1), 391–426.
- Manke, J., 1989. A Tensor Product b-Spline Method for Numerical Grid Generation. Technical report, Dept. of Applied Mathematics, Washington Univ., Seattle, WA (USA).
- Nian, X., Chen, F., 2016. Planar domain parameterization for isogeometric analysis based on Teichmüller mapping. *Comput. Methods Appl. Mech. Eng.* 311, 41–55.
- Schöberl, J., 1997. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Comput. Vis. Sci.* 1 (1), 41–52.
- Shewchuk, J.R., 1996. Triangle: engineering a 2d quality mesh generator and delaunay triangulator. In: Applied Computational Geometry Towards Geometric Engineering. Springer, pp. 203–222.
- Thompson, J.F., Soni, B.K., Weatherill, N.P., 1998. *Handbook of Grid Generation*. CRC Press.
- van Zwieten, G., Verhoosel, C., van Zwieten, J., van Opstal, T., Hoitinga, W., 2016. Nutils v2.0. Feb.
- Vuong, A.-V., 2012. *Adaptive Hierarchical Isogeometric Finite Element Methods*. Springer Science & Business Media.
- Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* 106 (1), 25–57.
- Wang, X., Qian, X., 2014. An optimization approach for constructing trivariate b-spline solids. *Comput. Aided Des.* 46, 179–191.
- Weber, O., Myles, A., Zorin, D., 2012. Computing Extremal Quasiconformal Maps. *Comput. Graph. Forum*, vol. 31. Wiley Online Library, pp. 1679–1689.
- Winslow, A.M., 1981. Adaptive-Mesh Zoning by the Equipotential Method. Technical report, Lawrence Livermore National Lab., CA (USA).
- Xu, G., Li, M., Mourrain, B., Rabczuk, T., Xu, J., Bordas, S.P., 2018. Constructing iga-suitable planar parameterization from complex cad boundary by domain partition and global/local optimization. *Comput. Methods Appl. Mech. Eng.* 328, 175–200.

- Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2010. Optimal analysis-aware parameterization of computational domain in isogeometric analysis. In: *International Conference on Geometric Modeling and Processing*. Springer, pp. 236–254.
- Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2011. Parameterization of computational domain in isogeometric analysis: methods and comparison. *Comput. Methods Appl. Mech. Eng.* 200 (23), 2021–2031.
- Xu, G., Mourrain, B., Duvigneau, R., Galligo, A., 2013. Constructing analysis-suitable parameterization of computational domain from cad boundary by variational harmonic method. *J. Comput. Phys.* 252, 275–289.
- Xu, G., Mourrain, B., Galligo, A., Rabczuk, T., 2014. High-quality construction of analysis-suitable trivariate nurbs solids by reparameterization methods. *Comput. Mech.* 54 (5), 1303–1313.