

Template Mapping Using Adaptive Splines and Optimal Parameterization

Svajūnas Sajavičius, Bert Jüttler and
Jaka Špeh

G+S Report No. 76

September 2018

Template Mapping Using Adaptive Splines and Optimal Parameterization

Svajūnas Sajavičius, Bert Jüttler and Jaka Špeh

Abstract We consider the construction of a spline map (a volumetric deformation) that transforms a template, which is given in the domain, into a target shape. More precisely, the domain is equipped with a set of surface patches (the template skeleton) and target patches for some of them (which are called the constraining patches) are specified. The constructed spline map approximately transforms the constraining patches into the associated target patches. Possible applications include isogeometric segmentation and parameterization of the computational domain. In particular, the approach should be useful when performing isogeometric segmentation and parameterization for a large class of computational domains possessing similar shapes. We present a solution approach, which is based on least-squares fitting. In order to deal with the influence of the parameterization, the well-established approaches of point and tangent distance minimization are employed for the iterative solution of the resulting nonlinear optimization problems. Additionally, we enrich the approach with spline space refinement. The efficiency and performance of the approach are investigated experimentally. The proposed template mapping approach is also applied to a case of industrial interest, as well as to a volumetric example.

1 Introduction

In this paper, we focus on the template mapping problem. The problem consists of the construction of a spline map (a volumetric deformation) that transforms the template domain into the target domain. The template map is constructed taken into account restrictions on mapping certain surface patches from the template domain: the template map transforms these surfaces (the constraining patches) into other predefined surfaces (the target patches).

Institute of Applied Geometry, Johannes Kepler University, Altenberger Str. 69, 4040 Linz, Austria, e-mail: svajunas.sajavicius@gmail.com, bert.juettler@jku.at, jaka@jakascorner.com

The template mapping possesses applications in isogeometric segmentation and parameterization (see [28, 39]). These two steps are essential as preprocessing steps for NURBS-based numerical simulation, i.e., for isogeometric analysis (IGA) [19]. Consequently, they have attracted substantial interest from the scientific community. We summarize some of the related literature:

IGA-suitable spline parameterizations of swept volumes are described in paper [1]. The paper [7] analyzes aspects of parameterization quality for geometric modeling in IGA. Solid modeling and domain parameterization using trivariate T-splines is discussed in [8, 25, 35].

The papers [14, 36] describe optimization based techniques for planar and volumetric domain parameterization in IGA. High-quality constructions for multi-patch NURBS parameterizations are introduced in [40]. The paper [9] uses inverse harmonic mappings for planar domain parameterization using truncated hierarchical B-splines (THB-splines), while [31] employs Powell–Sabin spline representations and [26] uses the Teichmüller mapping.

Unstructured spline spaces for IGA on manifolds are introduced in [30] (see also [21] for G^1 smooth discretizations). The conversion of the boundary representation (B-Rep) models into domain parameterizations for IGA is addressed in [2], while the construction of multi-patch parameterizations is discussed in [6]. Scaled boundary parameterizations for IGA are considered in [3]. The recent paper [17] adopts elliptic grid generation principles for IGA applications. Another approach to the computation of IGA-suitable planar parameterizations via PolySquare-enhanced domain partition is investigated in [38]. An approach for constructing low-rank parameterizations of planar domains is proposed in [27]. The paper [37] focuses on generating high-quality high-order Bézier triangular and tetrahedral elements for IGA on triangulations. The recent paper [16] gives an overview of isogeometric segmentation and parameterization and provides additional references.

The present paper is devoted to the creation of a template mapping, which should be a useful tool when solving the isogeometric domain segmentation and parameterization problem for a large class of computational domains possessing similar shapes and equivalent topologies. Such classes occur naturally in engineering, e.g., when one is trying to identify the optimal design for a specific application. In this setting it appears to be a promising approach to transfer a pre-defined segmentation and parameterization from a (simplified) master domain (the “template”) to each particular instance of the domain.

We solve the template mapping problem by applying an iterative procedure, which is based on least-squares fitting. In particular, we minimize an objective function that involves several terms representing the geometric error, smoothing and regularization.

For computational purposes, we need to discretize the objective function. The standard method to discretize the geometric error term is the *point distance minimization* (PDM) error term [18]. The optimization procedure based on the PDM error term is equivalent to an alternating optimization method. Though this is quite robust and leads to significant improvements of the initial results, it is also known to have a low rate of convergence. Another approach, which employs the *tangent dis-*

tance minimization (TDM) error terms [4], is a Gauss–Newton-type method, thus providing quadratic convergence for zero-residual problems.

The method of *squared distance minimization* (SDM), which was introduced in [29], provides an alternative to the above-mentioned discretization methods for the geometric error terms (see [5] and the references cited therein). The SDM error terms are curvature-dependent, thus requiring C^2 smoothness, and the method lacks clear theoretical advantages with respect to the TDM method. A detailed discussion of the SDM method is therefore beyond the scope of the present paper.

In this paper, we adapt the PDM/TDM methods to the template mapping problem. In addition, we enrich the iterative approach with spline space refinement. Besides various other observations that we obtain during the experimental study, we also demonstrate that the local (adaptive) spline space refinement used instead of the global (uniform) refinement helps to save computational resources. The local refinement allows achieving a similar accuracy of the results with significantly reduced computational effort.

The rest of the paper is organized as follows. Sect. 2 presents the basic notation, definitions, formulates the template mapping problem and shortly describes possible applications to isogeometric segmentation and parameterization. In Sect. 3, the template mapping problem is discretized. We give a detailed description of the presented iterative approach for the template mapping problem in Sect. 4. The results of the experimental study are presented in Sect. 5. Finally, Sect. 6 summarizes the paper by formulating conclusions and identifying directions for future work.

2 Template Mapping Problem

In this section, we introduce the basic notation and definitions. Before formulating the template mapping problem in dimension independent form, possible applications to isogeometric segmentation and parameterization are shortly described.

2.1 Template Mapping for Isogeometric Segmentation and Parameterization

Our interest in the template mapping originated in its possible applications to isogeometric segmentation and parameterization. To clarify the practical meaning of the theoretical concepts, which will be introduced below, we first summarize our approach to isogeometric segmentation and parameterization.

Assume that we have a B-Rep model of the computational domain, which we need to parameterize for the purpose of applications of IGA. If the domain is too complicated to be parameterized as a single patch, it needs to be segmented into quadrilateral or hexahedral subdomains (patches) that can be parameterized separately (multi-patch parameterization) in subsequent steps.

More precisely, we consider a template domain, described by a B-Rep model, which is topologically equivalent to the original computational domain (target domain). It is assumed that a segmentation of this domain can be defined in a natural way and subdomains can be parameterized easily using some standard technique (for example, Coons patches). If, in addition, we have a map that deforms the template domain into B-Rep of the original computational domain, we arrive at a multi-patch parameterization of the original computational domain.

2.2 Template Skeleton and Target Patches

Let us assume that $\hat{\Omega}$ is a domain with piecewise smooth boundary in $\mathbb{R}^d$ ($d = 2$ or $d = 3$). We call this domain the *template domain*. In the template domain $\hat{\Omega}$, the surface patches $\hat{\Gamma}^k$, $k \in \mathcal{K} = \{1, 2, \dots, M\}$, are given. Each of these patches is parameterized as

$$\hat{\gamma}^k : (0, 1)^{d-1} \rightarrow \hat{\Gamma}^k.$$

The surfaces $\hat{\Gamma}^k$, $k \in \mathcal{K}^* = \{1, 2, \dots, N\}$, $N < M$, are called the *constraining patches*, and the remaining ones ($k \in \mathcal{K} \setminus \mathcal{K}^*$) are referred to as the *free patches*. The union of the constraining and free patches forms the *template skeleton* $\hat{\Gamma}$:

$$\hat{\Gamma} = \bigcup_{k \in \mathcal{K}} \hat{\Gamma}^k.$$

Note that in the rest of the paper we use the notion *surface* to denote curves ($d = 2$) or two-dimensional surfaces ($d = 3$). In addition, we use hats ($\hat{\cdot}$) for the symbols that are related to the template domain or skeleton.

Each *constraining* patch $\hat{\Gamma}^k$ corresponds to the *target patch* $\Gamma^k \subset \mathbb{R}^d$ ($k \in \mathcal{K}^*$). The target patches Γ^k are parameterized as

$$\gamma^k : (0, 1)^{d-1} \rightarrow \Gamma^k.$$

We assume that the topological structure of the intersection of the constraining patches $\hat{\Gamma}^k \subset \hat{\Gamma}$ matches that of the target patches Γ^k .

2.3 Template Map

Our aim is to construct the *template map*

$$\mathbf{s} : \hat{\Omega} \rightarrow \mathbb{R}^d$$

that satisfies the requirements

$$\mathbf{s} \circ \hat{\gamma}^k \circ \varrho^k(\mathbf{t}) = \gamma^k(\mathbf{t}), \quad \mathbf{t} \in (0, 1)^{d-1}, \quad (1)$$

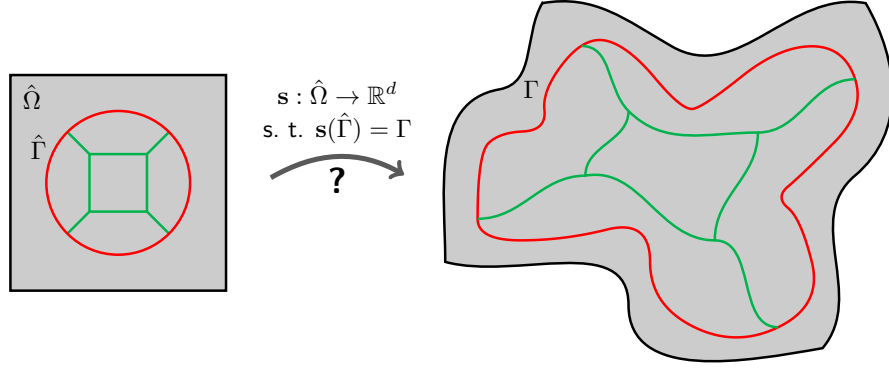


Fig. 1 The template mapping problem: find the map (a volumetric deformation) $\mathbf{s}$ that transfers the template skeleton (red and green, left) in such a way that the constraining patches (red, left) are transformed into the target patches (red, right). The mapping of the free patches (green, left) is not constrained

for $k \in \mathcal{K}^*$. Here $\varrho^k: (0, 1)^{d-1} \rightarrow (0, 1)^{d-1}$ are the *reparameterizations* of the constraining patches $\hat{\Gamma}^k$. In general, the reparameterization functions ϱ^k can be chosen in different ways but they should always be bijective and regular. The requirements (1) ensure that the template map $\mathbf{s}$ transforms each constraining patch $\hat{\Gamma}^k$ to the corresponding target patch Γ^k , $k \in \mathcal{K}^*$. We emphasize that the requirements (1) are formulated for the constraining and target patches only. For application purposes, the template map $\mathbf{s}$ is also supposed to be injective and regular in $\hat{\Omega}$.

The surfaces $\tilde{\Gamma}^k$ parameterized as

$$\tilde{\gamma}^k: (0, 1)^{d-1} \rightarrow \tilde{\Gamma}^k: \hat{\mathbf{t}} \mapsto \mathbf{s} \circ \hat{\gamma}^k \circ \varrho^k(\hat{\mathbf{t}})$$

are called the *mapped (transformed) constraining patches* (for $k \in \mathcal{K}^*$) or the *mapped (transformed) free patches* (for $k \in \mathcal{K}$). We expect that (cf. (1))

$$\gamma^k(\mathbf{t}) \approx \tilde{\gamma}^k(\mathbf{t}), \quad \mathbf{t} \in (0, 1)^{d-1},$$

is valid for each target patch Γ^k and the corresponding mapped constraining patch $\tilde{\Gamma}^k$ ($k \in \mathcal{K}^*$).

A schematic description of the template mapping problem is presented in Fig. 1.

2.4 Variational Formulation of the Template Mapping Problem

We construct the template map $\mathbf{s}$ satisfying the constraints (1) by solving a nonlinear optimization (the least-squares) problem [22]. Indeed, we define and minimize the least-squares objective function:

$$F = \sum_{k \in \mathcal{K}^*} \|\gamma^k - \tilde{\gamma}^k\|_{L_2([0,1]^{d-1})}^2 + Q(\mathbf{s}) \rightarrow \min. \quad (2)$$

The term Q in the objective function stands for the smoothing and regularization. This term involves functionals known as the *quality measures*. For further information the interested reader is referred to [9, 14], where these measures are applied in the context of isogeometric domain parameterization.

3 Discretization

In Sect. 2 we formulated the template mapping problem in an abstract way. A proper template map is expected to satisfy certain requirements. In this section, the template mapping problem is discretized.

3.1 Spline Approximation of the Template Map

We assume that the template map $\mathbf{s}$ is a spline function represented in a hierarchical spline space as a linear combination of the spline basis functions τ_j^ℓ defined by coefficients $\mathbf{c}_j^\ell \in \mathbb{R}^d$:

$$\mathbf{s}(\mathbf{u}) = \sum_{\ell=1}^L \sum_{j=1}^{J^\ell} \mathbf{c}_j^\ell \tau_j^\ell(\mathbf{u}), \quad \mathbf{u} \in \hat{\Omega}. \quad (3)$$

Here the upper index ℓ stands for the hierarchical level, L denotes the total number of levels of refinement, and J^ℓ is the numbers of spline basis functions of level ℓ . For the simplicity of notation, all the coefficients $\mathbf{c}_j^\ell$ are collected in one matrix

$$\mathbf{c} = (\dots, \mathbf{c}_j^\ell, \dots) \in \mathbb{R}^{d \times n},$$

where $n = \sum_{\ell=1}^L J^\ell$ is the number of hierarchical spline basis functions in all levels.

3.2 Discretization of Geometric Input Data

The target patches Γ^k are discretized by sampling points $\{\mathbf{p}_i^k\}_{i \in \mathcal{I}}$ corresponding to the initial parameter values $\{\mathbf{t}_i^k\}_{i \in \mathcal{I}}$:

$$\mathbf{p}_i^k = \gamma^k(\mathbf{t}_i^k),$$

where $\mathbf{t}_i^k \in (0,1)^{d-1}$ and $\mathcal{I}$ is an index set.

For the purpose of the closest point computation (see Sect. 3.3), we also discretize the constraining patches $\hat{\Gamma}^k$, $k \in \mathcal{K}^*$. Each constraining patch is discretized by sampling points $\{\hat{\mathbf{p}}_i^k\}_{i \in \hat{\mathcal{J}}}$ corresponding to the parameter values $\{\hat{\mathbf{t}}_i^k\}_{i \in \hat{\mathcal{J}}}$:

$$\hat{\mathbf{p}}_i^k = \hat{\gamma}^k(\hat{\mathbf{t}}_i^k),$$

where $\hat{\mathbf{t}}_i^k \in (0, 1)^{d-1}$ and $\hat{\mathcal{J}}$ is an index set. These sampled points are used for the initialization of the closest point computation. The number of points sampled on the constraining patches is expected to be much larger than the number of points sampled on the target patches ($|\hat{\mathcal{J}}| \gg |\mathcal{J}|$).

In the rest of the paper we assume that $\hat{i} \in \hat{\mathcal{J}}$, $i \in \mathcal{J}$ and $k \in \mathcal{K}^*$.

3.3 Discrete Representation of the Reparameterization Functions

Let us assume we have the template map $\mathbf{s}$. For each target patch Γ^k and each sampled point $\mathbf{p}_i^k$, the reparameterization function ϱ^k defines the parameter value $\hat{\mathbf{t}}_i^{k,*}$ such that the mapped point $\tilde{\mathbf{p}}_i^{k,*} = \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^{k,*})$ is the closest point to the point $\mathbf{p}_i^k$ from all the points on the mapped constraining patch $\tilde{\Gamma}^k$:

$$\hat{\mathbf{t}}_i^{k,*} := \arg \min_{\hat{\mathbf{t}} \in (0, 1)^{d-1}} \|\mathbf{p}_i^k - \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}})\|.$$

In this way, the reparameterization functions ϱ^k are represented discretely by the *optimal parameter values* $\{\hat{\mathbf{t}}_i^{k,*}\}$:

$$\varrho^k: (0, 1)^{d-1} \rightarrow (0, 1)^{d-1} : \hat{\mathbf{t}}_i^k \mapsto \hat{\mathbf{t}}_i^{k,*}.$$

The optimal parameter values are expected to ensure the difference vectors $\mathbf{p}_i^k - \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^{k,*})$ to be parallel to the unit normal vectors at the points $\mathbf{p}_i^k$ on the target patch Γ^k (the *parameter optimality constraint*). The points $\tilde{\mathbf{p}}_i^{k,*} = \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^{k,*})$ are called the *closest points* (or *foot points*) of the points $\mathbf{p}_i^k$ on the mapped constraining patch $\tilde{\Gamma}^k$.

3.4 Discretized Optimization Problem

Let $\{\mathbf{p}_i^k\}$ be a discrete set of points sampled on each target patch Γ^k . The objective function defined by (2) is discretized as

$$\tilde{F} = \sum_k \sum_i e_i^k + \tilde{Q}, \quad (4)$$

where $e_i^k = \min_{\mathbf{t} \in (0,1)^{d-1}} \|\mathbf{p}_i^k - \tilde{\gamma}^k(\hat{\mathbf{t}})\|^2$ are the *squared orthogonal distances* between the points $\mathbf{p}_i^k$ and the mapped target patch $\tilde{\Gamma}^k$, and $\tilde{Q}$ is a discretized smoothing and regularization term. The smoothing and regularization term is added to the objective function in order to avoid situations when this linear system becomes singular and self-overlappings appear in the resulting map.

We minimize the objective function $\tilde{F}$ with respect to the reparameterization functions $\mathbf{q}^k$ and the coefficients $\mathbf{c}$:

$$\tilde{F} \rightarrow \min_{\mathbf{q}^k, \mathbf{c}}. \quad (5)$$

This nonlinear least-squares problem is a *separable* and *constrained* optimization problem. Indeed, the reparameterization functions $\mathbf{q}^k$ and the coefficients $\mathbf{c}$ can be treated as two separate groups of optimization variables, and the optimization problem can be formulated as the minimization of the objective function (4) with respect to the coefficients $\mathbf{c}$, i.e.

$$\tilde{F} \rightarrow \min_{\mathbf{c}},$$

subject to the parameter optimization constraints (see Sect. 3.3). For the solution of this optimization problem we will apply an iterative procedure presented in the next section.

4 Iterative Solution Procedure

Our aim is to minimize the objective function (4) with respect to the coefficients of the discretized template map, and the reparameterization functions. We apply an iterative procedure for the solution of this optimization problem. In this section, we give a general outline of the procedure as well as describe its steps in detail.

4.1 General Outline of the Procedure

As it was mentioned in Sect. 3.4, the optimization problem (5) is separable. Therefore, we treat the unknown coefficients $\mathbf{c}$ of the map $\mathbf{s}$ and the reparameterization functions $\mathbf{q}^k$ in separate steps. The outline of the iterative procedure for the template mapping problem solution is the following:

- Step 1:* Discretization and initialization
- Step 2:* Control point computation
- Step 3:* Closest point computation
- Step 4:* Checking termination and refinement criteria:
 - (4a) Termination criterion
 - (4b) Refinement criterion
- Step 5:* Spline space refinement

In the next sections we will describe all the steps in detail.

4.2 Step 1: Discretization and Initialization

We assume the target patches Γ^k are discretized by sampled points $\{\mathbf{p}_i^k\}$ (see Sect. 3.2). In addition, we also construct the initial version of the template map $\mathbf{s}$. The initial template map is defined by initial tensor-product spline basis and initial values of the coefficients $\mathbf{c}$.

As the initial map for the iterative procedure, we use the identity map. This choice is suitable in cases when the constraining patches are quite similar to the target patches. Another possibility is to execute one iteration of the control point computation step (Sect. 4.3) with a simplified objective function (in order to obtain a simpler quadratic optimization problem).

4.3 Step 2: Control Point Computation

Suppose that the template map $\mathbf{s}$ defined by (3) is an initial map, or the current map generated in the previous iteration of the procedure. In this step, we minimize the objective function $\tilde{F}$ with respect to the coefficients $\mathbf{c}$. Clearly, the new values of the coefficients $\mathbf{c}$ define an updated map $\mathbf{s}$.

For the discretization of the squared orthogonal distances (error terms) e_i^k in the discretized objective function (4) we implemented and examined the *point distance minimization* (PDM) and the *tangent distance minimization* (TDM) procedures (see e.g. [34] and references therein). The error terms e_i^k in the PDM are expressed as

$$e_{\text{PDM},i}^k = \|\mathbf{p}_i^k - \tilde{\gamma}^k(\hat{\mathbf{t}}_i^k)\|^2,$$

while in the TDM procedure they are approximated as

$$e_{\text{TDM},i}^k = \left[\left(\mathbf{p}_i^k - \tilde{\gamma}^k(\hat{\mathbf{t}}_i^k) \right)^\top \cdot \mathbf{N}_i^k \right]^2,$$

where $\mathbf{N}_i^k$ are the unit normal vectors at the points $\mathbf{p}_i^k$ on the target patch Γ^k . Both PDM and TDM procedures can be combined together. Then the squared orthogonal distances are expressed as

$$e_i^k \approx \omega_{\text{PDM}} e_{\text{PDM},i}^k + \omega_{\text{TDM}} e_{\text{TDM},i}^k,$$

where ω_{PDM} and ω_{TDM} are the weights controlling the influence of PDM and TDM errors terms.

The template map coefficients $\mathbf{c}$ are obtained from the linear system arising after the differentiation of the objective function (4) with respect to unknown coefficients $\mathbf{c}$,

$$\frac{\partial \tilde{F}}{\partial \mathbf{c}} = 0.$$

In our implementation, the smoothing and regularization term is assumed to be

$$\tilde{Q} = \omega_r \tilde{Q}_r + \omega_u \tilde{Q}_u, \quad (6)$$

where ω_r and ω_u are user-defined non-negative weights for Tikhonov regularization term

$$\tilde{Q}_r(\mathbf{c}) = \sum_{\ell=1}^L \sum_{j=1}^{J^\ell} \|\mathbf{c}_j^\ell - \bar{\mathbf{c}}_j^\ell\|^2$$

and the discretized uniformity functional $\tilde{Q}_u$ (see [9]), while $\bar{\mathbf{c}}_j^\ell$ are the coefficients from the previous iteration.

The uniformity and the parametric length functionals [9] are quadratic. However, the rest of the quality measures mentioned in Sect. 2.4 are non-quadratic functionals. They are expressed as integrals of squares of functions depending on the first and second derivatives of the map. The objective functions involving non-quadratic functionals can be minimized using *Gauss–Newton-type* methods [22].

4.4 Step 3: Closest Point Computation

In this step, we change (update) the parameter values $\{\hat{\mathbf{t}}_i^k\}$ to the parameter values $\{\hat{\mathbf{t}}_i^{k,*}\}$ in such a way that the mapped points $\tilde{\mathbf{p}}_i^k = \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^{k,*})$ are the closest points (see Sect. 3.3) to the corresponding points $\mathbf{p}_i^k$ on the target patch Γ^k . Taking into account the discretization of the constraining patches (see Sect. 3.2), the initial parameter values are computed as

$$\hat{\mathbf{t}}_i^{k,*} := \arg \min_i \|\mathbf{p}_i^k - \tilde{\mathbf{p}}_i^k\|,$$

where $\tilde{\mathbf{p}}_i^k = \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k)$.

The initial parameter values are improved by applying Newton steps:

$$\hat{\mathbf{t}}_i^{k,*} := \hat{\mathbf{t}}_i^{k,*} + \Delta \hat{\mathbf{t}}_i^{k,*},$$

where

$$\Delta \hat{\mathbf{t}}_i^{k,*} = \frac{\left(\mathbf{p}_i^k - \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \right)^\top \cdot \left(\mathbf{s}_{u_1} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \right)}{\|\mathbf{s}_{u_1} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k)\|^2}$$

in planar ($d = 2$) case, and

$$\Delta \hat{\mathbf{t}}_i^{k,*} = \begin{bmatrix} \|\mathbf{s}_{u_1} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k)\| & (\mathbf{s}_{u_1} \mathbf{s}_{u_2}) \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \\ (\mathbf{s}_{u_1} \mathbf{s}_{u_2}) \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) & \|\mathbf{s}_{u_2} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k)\| \end{bmatrix}^{-1} \begin{Bmatrix} \left(\mathbf{p}_i^k - \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \right)^\top \cdot \left(\mathbf{s}_{u_1} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \right) \\ \left(\mathbf{p}_i^k - \mathbf{s} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \right)^\top \cdot \left(\mathbf{s}_{u_2} \circ \hat{\gamma}^k(\hat{\mathbf{t}}_i^k) \right) \end{Bmatrix}$$

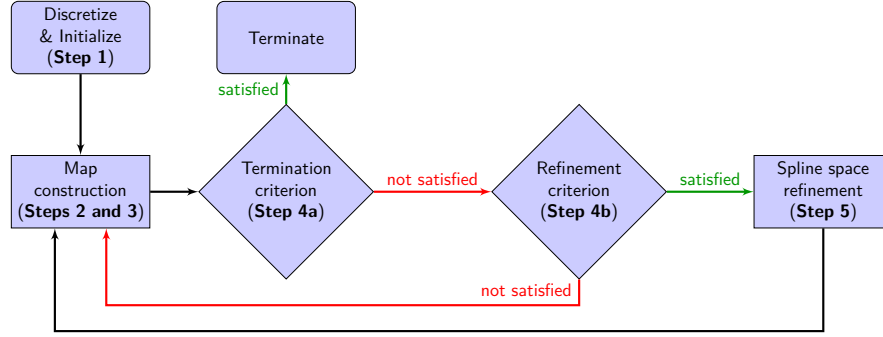


Fig. 2 The iterative approach for template mapping problem

in volumetric ($d = 3$) case.

4.5 Step 4: Checking Termination and Refinement Criteria

Steps 2 and 3 are repeated iteratively until a certain termination criterion is satisfied, e.g., the prescribed accuracy is achieved or the maximal number of iterations is reached. The procedure also terminates if the template map $\mathbf{s}$ becomes unacceptably irregular. We monitor the determinant of the Jacobian at certain points in $\hat{\Omega}$ and terminate the procedure if the percentage of points with negative values of Jacobian determinant exceeds a prescribed threshold.

In addition, the proposed approach for the template mapping problem can be enriched with the refinement of the spline space in which the template map $\mathbf{s}$ is represented. After several PDM/TDM iterations the accuracy of the results can be estimated and, if necessary, the spline space can be refined before executing another series of PDM/TDM iterations. As an indicator for the spline space refinement, the difference of the error in two successive iterations can be used: the spline space is refined whenever the error no longer changes significantly.

The flowchart representing the iterative approach is depicted in Fig. 2. The effectiveness of the approach combining PDM/TDM iterations and the spline space refinement will be demonstrated in Sect. 5. The details on the spline space refinement step will be given in the next section.

4.6 Step 5: Spline Space Refinement

The standard tensor-product constructions of the multivariate splines provide the possibility of the global (uniform) refinement only. This means that the insertion of a new knot into one of the knot vectors refines the entire column or row of cells

in the mesh. In order to overcome this, various generalizations of tensor-product splines were proposed (see [11] and the references cited therein).

In our implementation of the approach, the local refinement is based on THB-splines [11, 12], which form another basis for the space of hierarchical splines [10, 15, 24]. In addition to the possibility of the local refinement, the THB-splines possess numerous nice mathematical properties. Compared to the hierarchical splines introduced in [24], THB-splines form a non-negative partition of unity and have the same or a smaller support. In addition, THB-splines are linearly independent and strongly stable with respect to the maximum norm [13].

In order to select hierarchical mesh cells that should be refined, various strategies can be applied. We can mention the *absolute threshold* and *relative threshold* approaches [23]. In the first strategy, the points where the error exceeds a user-defined threshold are marked for the refinement, while the latter approach marks a certain percentage of points with the largest errors.

The size of the refined area and, consequently, the size of the corresponding THB-spline basis can be affected by properly adjusting the *extension* parameter [23].

As we will see in computational experiments, the local refinement of the spline space allows to significantly reduce the amount of computational resources, which are required in the case of the global refinement.

5 Experimental Study

In this section we set up a test example and use it to analyze and compare different versions of the considered approach for the template mapping problem in terms of the accuracy and computational complexity (the amount of required computational resources). The experimental study consists of four experiments. In one of them, we apply the approach to a case of industrial interest. An experiment demonstrating the approach applicability to volumetric cases is also presented.

5.1 Implementation Details

The template skeleton and target patches, which will be used in Experiments 1 and 2 of our experimental study, are presented in Fig. 3. For simplicity, in the first two experiments we analyze the template skeleton without free patches (all the patches are constraining). The template skeletons and the target patches of industrial and volumetric examples will be introduced in Sects. 5.4 and 5.5.

The iterative procedure is initialized by defining an initial map and sampling points $\mathbf{p}_i^k$ on each target patch Γ^k . In our experiments, as the initial map we use the identity map defined in a space of tensor-product B-splines of bi-degree (3,3) (in planar cases) or tri-degree (3,3,3) (in volumetric case). The B-spline basis is defined using uniform knot vectors with 11 inner knots in each direction.

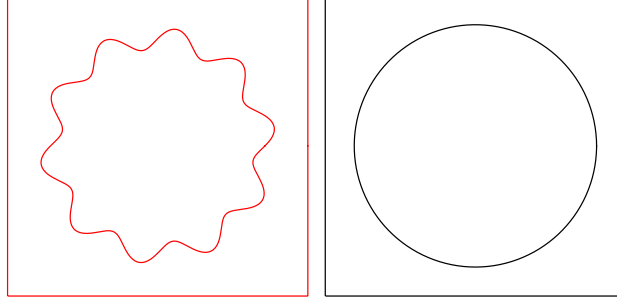


Fig. 3 Target patches (left) and constraining patches (right) used in Experiments 1 and 2

On each target patch in planar examples (Experiments 1–3), we sample 200 points corresponding to the parameters uniformly distributed on the unit interval. For the closest point computation, we discretize each constraining patch $\hat{\Gamma}^k$ by sampling 10^4 points. Then, we map these points using the template map $\mathbf{s}$. From all these mapped points we find an initial closest point on the target patch Γ^k for each point $\mathbf{p}_i^k$. An initial closest point then is improved by executing two Newton steps (Sect. 4.4).

The selection of appropriate weights ω_r and ω_u of the Tikhonov regularization term and the uniformity functional in the discretized objective function (4) is one of the main challenges of the proposed approach. These weights affect both the accuracy and quality of the final result (see e.g. [23] for the detailed discussion) and, therefore, they should be selected very carefully. In Experiments 1, 2 and 4 we will only apply the Tikhonov regularization with the weight $\omega_r = 2 \times 10^{-2}$. In Experiment 3, additionally we will use the uniformity functional with the weight $\omega_u = 2 \times 10^{-2}$.

In our experiments, the hierarchical mesh cells for the local refinement are marked using the absolute threshold strategy with the threshold of the pointwise maximal L_2 -error being constant and equal to 10^{-5} . The marked cells then are refined using the dyadic cell refinement.

The accuracy of the results is measured using the squared l_2 -error:

$$E_{l_2} = \sum_k \sum_i \|\mathbf{p}_i^k - \tilde{\mathbf{p}}_i^k\|^2.$$

Before the error estimation, the closest point computation (parameter optimization) step is executed.

All the steps of the examined approach have been implemented in G+SMO (Geometry + Simulation modules) C++ library for IGA [20]. This library includes an efficient implementation of the THB-splines.

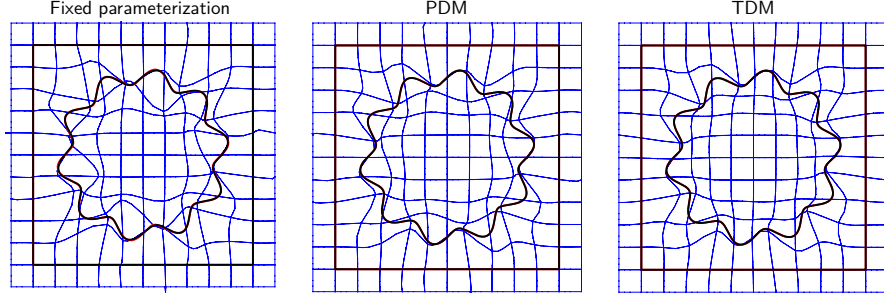


Fig. 4 The template mapping obtained after 20 iterations of different approaches: fixed parameterization (left), PDM (center) and TDM (right) (Experiment 1)

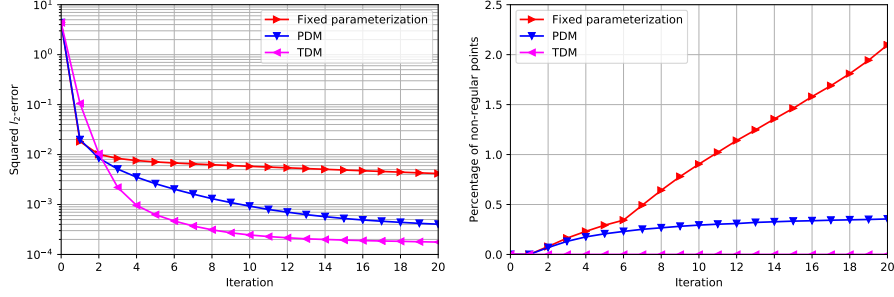


Fig. 5 Squared l_2 -errors (left) and the percentage of points with negative values of the Jacobian determinant (right) (Experiment 1)

Table 1 Squared l_2 -errors and the percentage of points with negative values of the corresponding map Jacobian determinant obtained after 20 iterations of the procedure with fixed parameterization and PDM/TDM procedures with the closet point computation step (Experiment 1)

	l_2 -error	Non-regular points
Fixed parameterization	4.16585×10^{-3}	2.0969 %
PDM	4.03419×10^{-4}	0.3552 %
TDM	1.76909×10^{-4}	0.0000 %

5.2 Experiment 1: Comparison of PDM and TDM

In the first experiment, we are going to demonstrate that the closest point computation (parameter optimization) step (Step 3) is an essential component of the approach. Moreover, we compare convergence rates of the procedures and the regularity of the resulting template maps.

In Fig. 4 we demonstrate the template mapping results obtained using procedure with the fixed parameterization, as well as the approach based on PDM or TDM with the parameter optimization step. In this experiment, the template map $\mathbf{s}$ is con-

structed in the initial tensor-product spline space (the spline space refinement is not applied).

From Fig. 4 we can see that the resulting map $\mathbf{s}$ obtained using the fixed parameterization has self-overlappings. To estimate the regularity of the template map $\mathbf{s}$, we computed the determinant of the map Jacobian at 10^6 points sampled in the parametric domain. The squared l_2 -errors obtained during 20 iterations of the procedure as well as the percentage of points with negative values of the Jacobian determinant are presented in Fig. 5 (see also Table 1 for the corresponding numerical values after the last of 20 iterations). The most accurate results were obtained using the approach based on TDM. The PDM with the closest point computation step produced slightly less accurate results.

In the case of the procedure with fixed parameterization, the percentage of non-regular points (points with negative values of Jacobian determinant) exceeds 2%. The closest point computation in PDM procedure reduces this percentage to around 0.36%. In the map produced by the procedure based on TDM, we do not identify non-regular points at all.

Clearly, the approach benefits from the closest point computation (parameter optimization) step. Therefore, we will use the closest point computation in the rest of this study. Moreover, since TDM seems to have clear advances with respect to PDM, we limit ourselves to TDM in the remaining experiments.

5.3 Experiment 2: Comparison of the Global and Local Spline Space Refinement Strategies

The goal of the second experiment is to compare the global and local spline space refinement strategies (see Sect. 4.6) in terms of the accuracy of the results and the amount of the required computational resources.

In this experiment, the spline space refinement is done after each single TDM iteration. The template mapping results are presented in Fig. 6. By using the same map regularity testing procedure as in Experiment 1, we do not identify any points with negative values of the Jacobian determinant. From Fig. 7 (left) we see that the number of degrees of freedom (the size of the spline space) grows exponentially if the spline space is refined globally. In case of the local spline space refinement, this number grows only linearly. This observation is expected and complies with the results obtained in [23].

From Fig. 7 (right) we also see that the local refinement leads to a similar accuracy of the results when using significantly coarser spline spaces (with much fewer degrees of freedoms in the spline map representation). Consequently, in order to achieve a similar accuracy, the local refinement strategy requires much less additional computational resources (computational time and memory) in comparison with the global one.

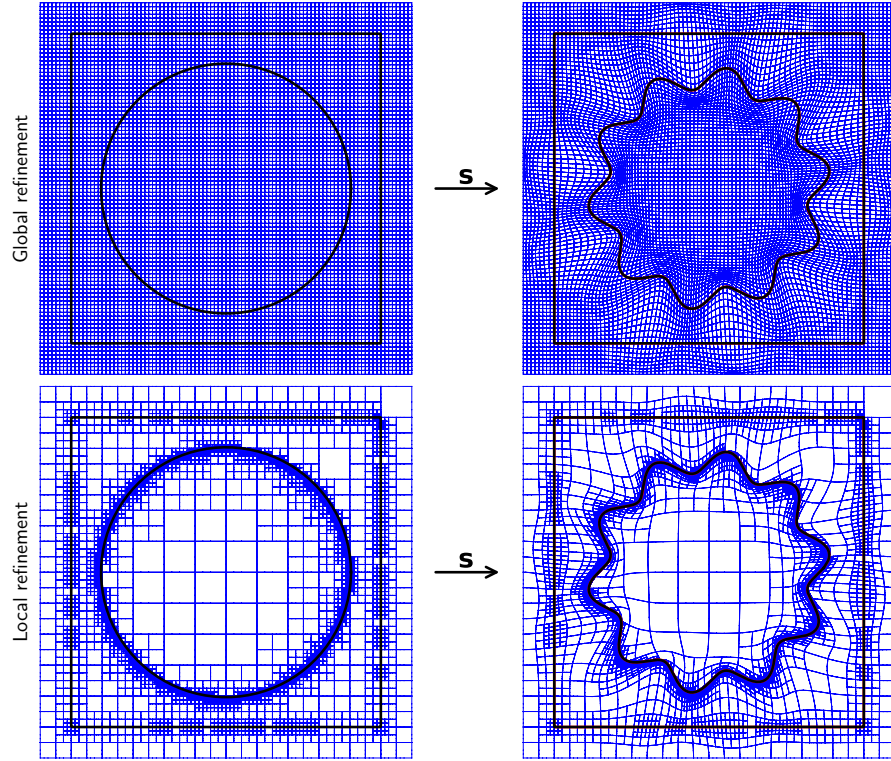


Fig. 6 The template mapping obtained after three iterations of global refinement (top) or seven iterations of local refinement (bottom) of the initial spline space (Experiment 2)

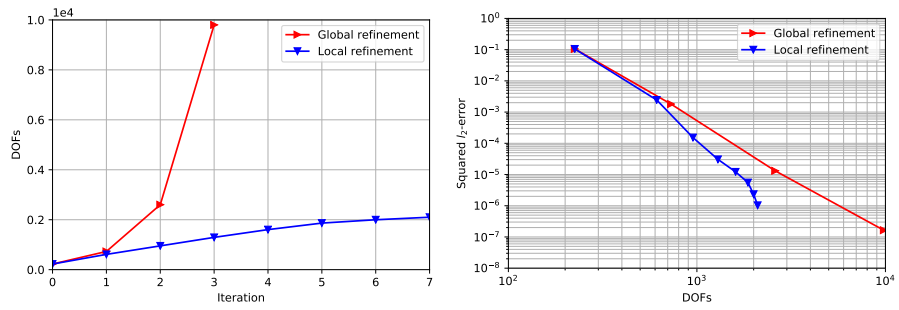


Fig. 7 The number of degrees of freedom (DOFs) (left), and the squared l_2 -errors vs. the number of degrees of freedom (right) obtained during the global and local refinement of the spline space (Experiment 2)

5.4 Experiment 3: Industrial example

In this experiment, the presented approach is applied to two cases corresponding to the target patches, which represent the profiles of twin screw compressor rotors with four (male rotor) and six (female rotor) lobes (Fig. 8, top), see [32, 33] for additional details. The corresponding template skeletons are depicted in Fig. 8 (bottom). In this case, the template skeletons contain not only constraining patches but free ones too.

By this experiment we aim to demonstrate the approach applicability to the cases with industrial input and, in addition, investigate the possibility to combine series of TDM iterations and the local spline space refinement. We combine TDM procedure with local refinement of the spline space, i.e., the spline space refinement iteration is executed after each series of five TDM iterations. In addition to Tikhonov regularization ($\omega_r = 2 \times 10^{-2}$), the uniformity functional with the weight $\omega_u = 2 \times 10^{-2}$ is also used in this experiment. The identity map is used for the initialization of the iterative procedure.

The knot configurations in parametric and physical domains are exhibited in Figs. 9 and 10. The maps presented in these figures have no non-regular points. In addition, Fig. 11 demonstrates how the errors change after each iteration of TDM procedure. We see that after several iterations error decay slows down. The refinement of the spline space helps to speed up the convergence and obtain more accurate final results. We have already demonstrated in Experiment 2 that the local refinement strategy allows saving computational resources in comparison with the global one.

5.5 Experiment 4: Volumetric example

So far, we applied the template mapping approach for the planar ($d = 2$) cases only. In the last experiment, we demonstrate the approach applicability to volumetric ($d = 3$) cases. The input data and the results are presented Fig. 12.

The set of the target patches consists of four faces from a rectangular hexahedron and a patch made by extruding and rotating a curve from Experiments 1 and 2 (Fig. 3). Correspondingly, the set of template patches consists four faces from a rectangular hexahedron and a one-sheeted hyperboloid (cylindrical patch made by extruding and rotating a circle).

In Fig. 12 we demonstrate how the mapped template patches match the target patches. The target patches were mapped using a volumetric template map obtained after two series of ten TDM iterations combined with one iteration of the adaptive spline space refinement. The initial template map is represented by 6,859 degrees of freedom, while after one iteration of the local refinement this number increased to 12,248. In case of the global spline space refinement, we would need 42,875 degrees of freedom. The map produced by the iterative procedure has no non-regular points.

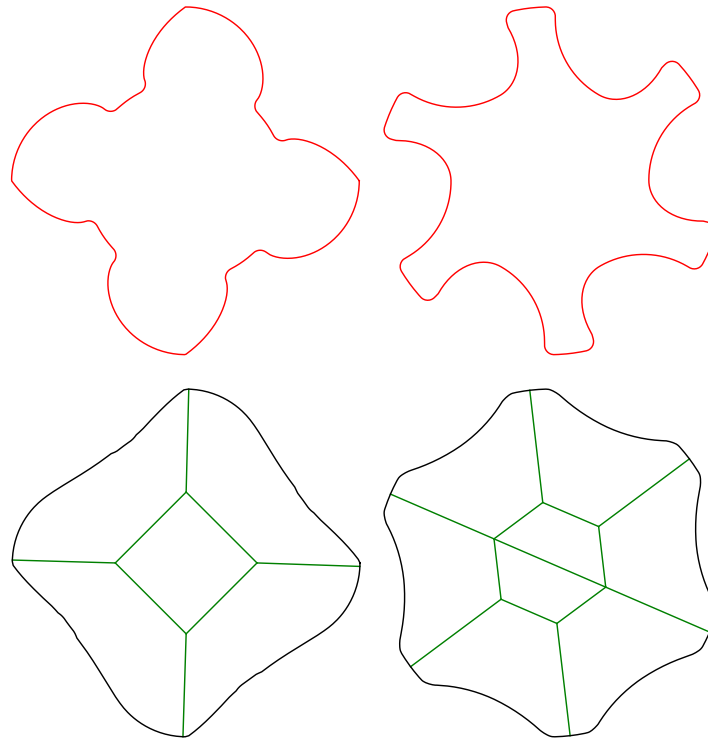


Fig. 8 Target patches (top) and template skeletons (bottom) used in Experiment 3. The target patches represent the profiles of a male rotor (left) and a female rotor (right) in a twin screw compressor. The template skeletons contain both the constraining patches (black) and free patches (green)

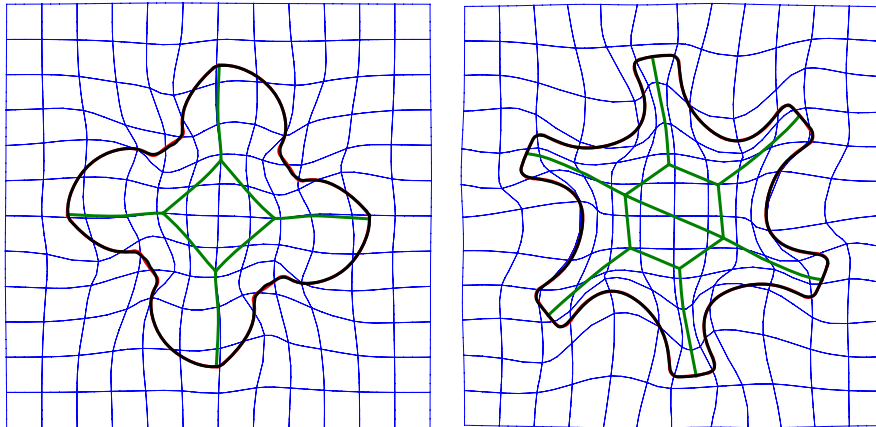


Fig. 9 The template mapping obtained after 20 iterations using TDM (Experiment 3)

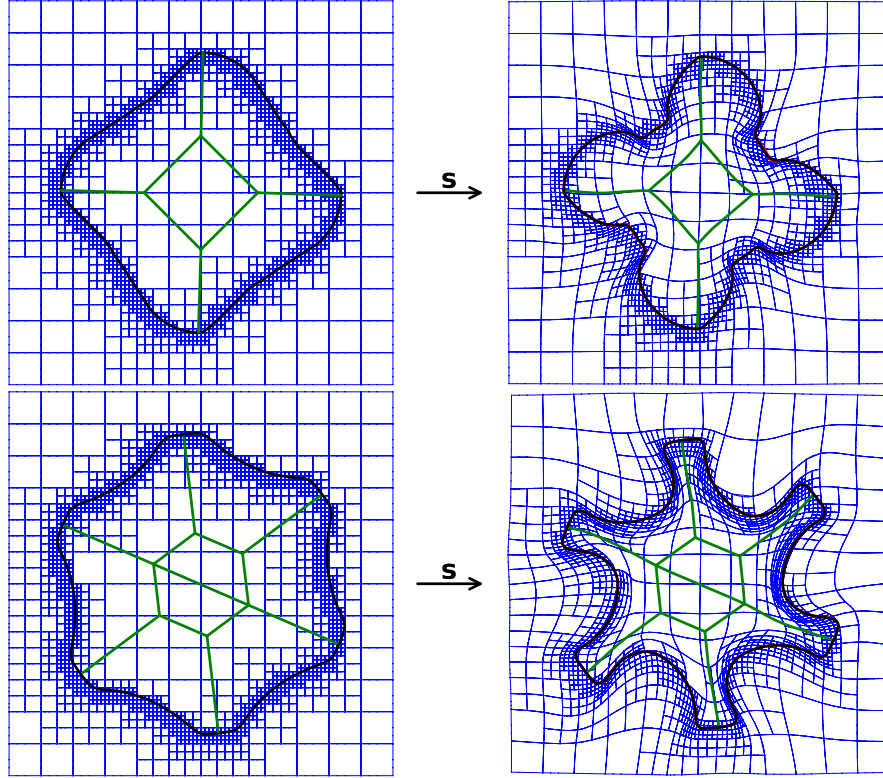


Fig. 10 The template mapping obtained after 20 iterations using TDM combined with three iterations of local refinement of the spline space (Experiment 3)

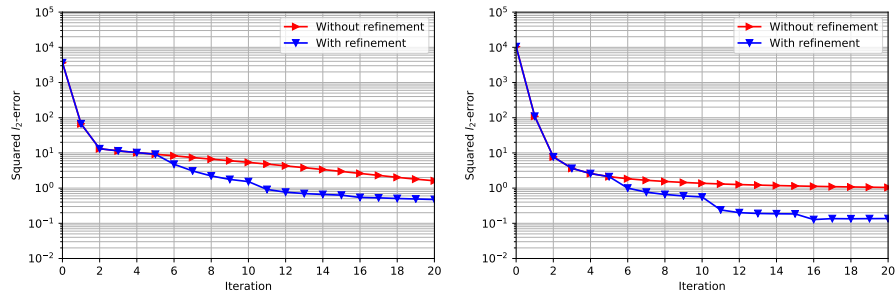


Fig. 11 Squared l_2 -errors obtained by combining TDM and spline space refinement (Experiment 3): the profiles of twin screw compressor male rotor (left) and female rotor (right)

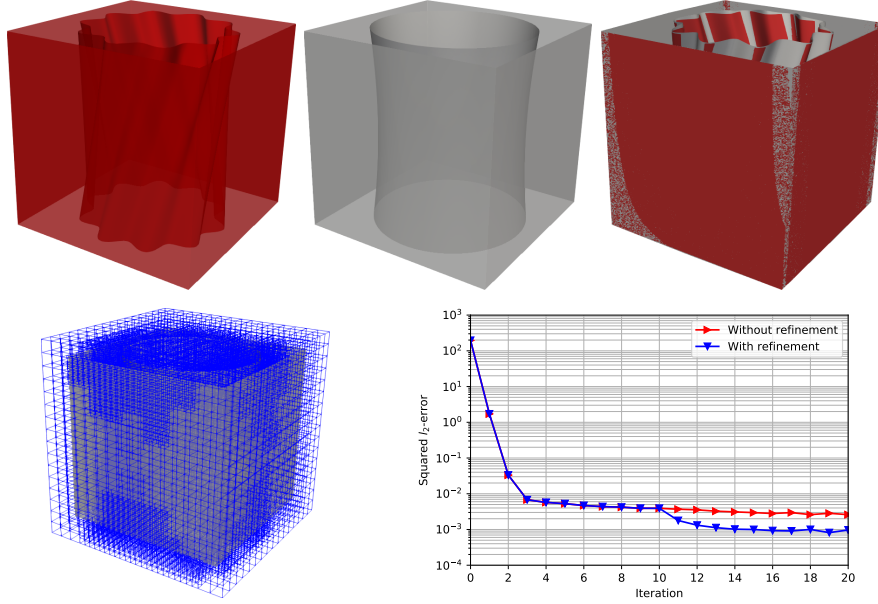


Fig. 12 Volumetric example of the template mapping (Experiment 4). Top row: target patches (left), template patches (middle), target patches and mapped template patches (right). Bottom row: hierarchical mesh after one iteration of adaptive spline space refinement (left) and squared l_2 -errors (right)

We also compared the convergence speed of procedures with and without spline space refinement. From Fig. 12 (bottom, right) we see that the local spline space refinement iteration, executed when the decay of the error already is slow, can slightly speed up the convergence. Although in this case the speedup is not extremely high, already first TDM iteration after spline space refinement gives the results, which are more accurate in comparison to those obtained after the last iteration of TDM procedure without spline space refinement. Therefore, this experiment once again confirm that the adaptive spline space refinement allows reducing the number of required iterations and, consequently, the amount of computational resources.

6 Concluding Remarks

We introduced the template mapping problem and presented an iterative adaptive approach for solving it. Based on the results of the experimental investigation, we arrive at the following conclusions:

- The closest point computation (parameter optimization) is beneficial for solving the template mapping problem. The iterative procedure based on TDM converges faster than the procedure based on PDM.

- In comparison with the global refinement of the spline space, the local refinement allows achieving the similar accuracy with significantly less amount of computational resources.
- The iterative minimization of the objective function can be efficiently combined with the spline space refinement.
- The approach is applicable to volumetric data. This is very important from the practical point, in view of applications to isogeometric segmentation and parameterization.

Acknowledgements Supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 678727 (the MOTOR project). We are grateful to our project partners from Dortmund for providing the data for the industrial example.

References

1. Aigner, M., Heinrich, C., Jüttler, B., Pilgerstorfer, E., Simeon, B., Vuong, A.V.: Swept volume parameterization for isogeometric analysis. In: E.R. Hancock, R.R. Martin, M.A. Sabin (eds.) *Mathematics of Surfaces XIII: 13th IMA International Conference, Lecture Notes in Computer Science*, vol. 5654, pp. 19–44. Springer, Berlin, Heidelberg (2009)
2. Al Akhras, H., Elguedj, T., Gravouil, A., Rochette, M.: Towards an automatic isogeometric analysis suitable trivariate models generation—Application to geometric parametric analysis. *Comput. Methods Appl. Mech. Engrg.* **316**, 623–645 (2017)
3. Arioli, C., Shamanskiy, A., Klinkel, S., Simeon, B.: Scaled boundary parameterizations in isogeometric analysis. *ArXiv e-prints*, arXiv:1711.05760 (2017)
4. Blake, A., Isard, M.: *Active Contours*. Springer, London (1998)
5. Bo, P., Ling, R., Wang, W.: A revisit to fitting parametric surfaces to point clouds. *Comput. Graph.* **36**(5), 534–540 (2012)
6. Buchegger, F., Jüttler, B.: Planar multi-patch domain parameterization via patch adjacency graphs. *Comput.-Aided Des.* **82**, 2–12 (2017)
7. Cohen, E., Martin, T., Kirby, R.M., Lyche, T., Riesenfeld, R.F.: Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.* **200**(5–8), 334–356 (2010)
8. Escobar, J.M., Cascón, J.M., Rodríguez, E., Montenegro, R.: A new approach to solid modeling with trivariate T-splines based on mesh optimization. *Comput. Methods Appl. Mech. Engrg.* **200**(45–46), 3210–3222 (2011)
9. Falini, A., Špeh, J., Jüttler, B.: Planar domain parameterization with THB-splines. *Comput. Aided Geom. Design* **35–36**, 95–108 (2015)
10. Forsey, D.R., Bartels, R.H.: Hierarchical B-spline refinement. *SIGGRAPH Comput. Graph.* **22**(4), 205–212 (1988)
11. Giannelli, C., Jüttler, B., Kleiss, S.K., Mantzaflaris, A., Simeon, B., Špeh, J.: THB-splines: An effective mathematical technology for adaptive refinement in geometric design and isogeometric analysis. *Comput. Methods Appl. Mech. Engrg.* **299**, 337–365 (2016)
12. Giannelli, C., Jüttler, B., Speleers, H.: THB-splines: The truncated basis for hierarchical splines. *Comput. Aided Geom. Design* **29**(7), 485–498 (2012)
13. Giannelli, C., Jüttler, B., Speleers, H.: Strongly stable bases for adaptively refined multilevel spline spaces. *Adv. Comput. Math.* **40**(2), 459–490 (2014)
14. Gravesen, J., Evgrafov, A., Nguyen, D.M., Nørtoft, P.: Planar parametrization in isogeometric analysis. In: M. Floater, T. Lyche, M.L. Masure, K. Mørken, L.L. Schumaker (eds.) *Mathematical Methods for Curves and Surfaces: 8th International Conference, Lecture Notes in*

- Computer Science*, vol. 8177, pp. 189–212. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
15. Greiner, G., Hormann, K.: Interpolating and approximating scattered 3D-data with hierarchical tensor product B-splines. In: A. Le Méhauté, C. Rabut, L.L. Schumaker (eds.) *Surface Fitting and Multiresolution Methods*, pp. 163–172. Vanderbilt University Press, Nashville, TN (1997)
 16. Haberleitner, M., Jüttler, B., Scott, M.A., Thomas, D.C.: Isogeometric analysis: Representation of geometry. In: *Encyclopedia of Computational Mechanics*. John Wiley and Sons, Inc. (2017)
 17. Hinz, J., Möller, M., Vuik, C.: Elliptic grid generation techniques in the framework of isogeometric analysis applications. *Comput. Aided Geom. Design* (2018). DOI 10.1016/j.cagd.2018.03.023
 18. Hoschek, J.: Intrinsic parametrization for approximation. *Comput. Aided Geom. Design* **5**(1), 27–31 (1988)
 19. Hughes, T.J.R., Cottrell, J.A., Bazilevs, Y.: Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput. Methods Appl. Mech. Engrg.* **194**(39–41), 4135–4195 (2005)
 20. Jüttler, B., Langer, U., Mantzaflaris, A., Moore, S., Zulehner, W.: Geometry + Simulation Modules: Implementing isogeometric analysis. *PAMM* **14**(1), 961–962 (2014)
 21. Kapl, M., Sangalli, G., Takacs, T.: Construction of analysis-suitable G^1 planar multi-patch parameterizations. *Comput.-Aided Des.* **97**, 41–55 (2018)
 22. Kelley, C.T.: Iterative Methods for Optimization, *Frontiers in Applied Mathematics*, vol. 18. Society for Industrial and Applied Mathematics (1999)
 23. Kiss, G., Giannelli, C., Zore, U., Jüttler, B., Großmann, D., Barner, J.: Adaptive CAD model (re-)construction with THB-splines. *Graph. Models* **76**(5), 273–288 (2014)
 24. Kraft, R.: Adaptive and linearly independent multilevel B-splines. In: A. Le Méhauté, C. Rabut, L.L. Schumaker (eds.) *Surface Fitting and Multiresolution Methods*, pp. 209–218. Vanderbilt University Press, Nashville, TN (1997)
 25. Liu, L., Zhang, Y., Hughes, T.J.R., Scott, M.A., Sederberg, T.W.: Volumetric T-spline construction using Boolean operations. *Eng. Comput.* **30**(4), 425–439 (2014)
 26. Nian, X., Chen, F.: Planar domain parameterization for isogeometric analysis based on Teichmüller mapping. *Comput. Methods Appl. Mech. Engrg.* **311**, 41–55 (2016)
 27. Pan, M., Chen, F., Tong, W.: Low-rank parameterization of planar domains for isogeometric analysis. *Comput. Aided Geom. Design* **63**, 1–16 (2018)
 28. Pauley, M., Nguyen, D.M., Mayer, D., Špeh, J., Weeger, O., Jüttler, B.: The isogeometric segmentation pipeline, *Lecture Notes in Computational Science and Engineering*, vol. 107, pp. 51–72. Springer International Publishing (2015)
 29. Pottmann, H., Leopoldseder, S.: A concept for parametric surface fitting which avoids the parametrization problem. *Comput. Aided Geom. Design* **20**(6), 343–362 (2003)
 30. Sangalli, G., Takacs, T., Vázquez, R.: Unstructured spline spaces for isogeometric analysis based on spline manifolds. *Comput. Aided Geom. Design* **47**, 61–82 (2016)
 31. Speleers, H., Manni, C.: Optimizing domain parameterization in isogeometric analysis based on Powell–Sabin splines. *J. Comput. Appl. Math.* **289**, 68–86 (2015)
 32. Utri, M., Brümmer, A.: Improvement of the efficiency of twin-screw refrigeration compressors by means of dual lead rotors. In: *International Compressor Engineering Conference*, Paper 1428 (2016)
 33. Utri, M., Brümmer, A.: Energy potential of dual lead rotors for twin screw compressors. *IOP Conf. Ser.: Mater. Sci. Eng.* **232**(012018) (2017)
 34. Wang, W., Pottmann, H., Liu, Y.: Fitting B-spline curves to point clouds by curvature based squared distance minimization. *ACM Trans. Graph.* **25**(2), 214–238 (2006)
 35. Wang, W., Zhang, Y., Liu, L., Hughes, T.J.R.: Trivariate solid T-spline construction from boundary triangulations with arbitrary genus topology. *Comput.-Aided Des.* **45**(2), 351–360 (2013)
 36. Wang, X., Qian, X.: An optimization approach for constructing trivariate B-spline solids. *Comput.-Aided Des.* **46**, 179–191 (2014)

37. Xia, S., Qian, X.: Generating high-quality high-order parameterization for isogeometric analysis on triangulations. *Comput. Methods Appl. Mech. Engrg.* **338**, 1–26 (2018)
38. Xiao, S., Kang, H., Fu, X.M., Chen, F.: Computing IGA-suitable planar parameterizations by PolySquare-enhanced domain partition. *Comput. Aided Geom. Design* **62**, 29–43 (2018)
39. Xu, G., Li, M., Mourrain, B., Rabzuck, T., Xu, J., Bordas, S.P.A.: Constructing IGA-suitable planar parameterization from complex CAD boundary by domain partition and global/local optimization. *Comput. Methods Appl. Mech. Engrg.* **328**, 175–200 (2018)
40. Xu, G., Mourrain, B., Galligo, A., Rabczuk, T.: High-quality construction of analysis-suitable trivariate NURBS solids by reparameterization methods. *Comput. Mech.* **54**(5), 1303–1313 (2014)