



HAL
open science

Optimal analysis-aware parameterization of computational domain in isogeometric analysis

Gang Xu, Bernard Mourrain, Régis Duvigneau, André Galligo

► **To cite this version:**

Gang Xu, Bernard Mourrain, Régis Duvigneau, André Galligo. Optimal analysis-aware parameterization of computational domain in isogeometric analysis. [Technical Report] 2010. inria-00440130v1

HAL Id: inria-00440130

<https://inria.hal.science/inria-00440130v1>

Submitted on 9 Dec 2009 (v1), last revised 30 Jun 2010 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal analysis-aware parameterization of computational domain in isogeometric analysis

Gang Xu¹, Bernard Mourrain¹, Régis Duvigneau², André Galligo³

¹GALAAD, INRIA Sophia-Antipolis, 2004 Route des Lucioles, 06902 Cedex, France

²OPALE, INRIA Sophia-Antipolis, 2004 Route des Lucioles, 06902 Cedex, France
Firstname.Lastname@sophia.inria.fr

³University of Nice Sophia-Antipolis, 06108 Nice Cedex 02, France
galligo@unice.fr

Abstract. In isogeometric analysis (IGA for short) framework, computational domain is exactly described using the same representation as that employed in the CAD process. For a CAD object, we can construct various computational domain with same shape but with different parameterization. One basic requirement is that the resulting parameterization should have no self-intersections. In this paper, a linear and easy-to-check sufficient condition for injectivity of planar B-spline parameterization is proposed firstly. By an example of 2D thermal conduction problem, we show that different parameterization of computational domain has different impact on the simulation result and efficiency in IGA. For problems with exact solutions, we propose a shape optimization method to obtain optimal parameterization of computational domain. The proposed injective condition is used to check the injectivity of initial parameterization constructed by discrete Coons method. Several examples and comparisons are presented to show the effectiveness of the proposed method. Compared with the initial parameterization during refinement, the optimal parameterization can achieve the same accuracy but with less degrees of freedom.

Key words: isogeometric analysis; analysis-aware parameterization of computational domain; injectivity; shape optimization; steepest descent method

1 Introduction

CAGD software usually relies on splines or NURBS representations, but the analysis software for CAD object uses mesh-based geometric descriptions (structured or unstructured). Therefore, in conventional approaches, several information transfers occur during the design phase, yielding approximations and non-linear transformations that can significantly deteriorate the overall efficiency of the design optimization procedure.

The isogeometric approach proposed by Hughes et al. [19] is employed to overcome this difficulty by using CAD standards as unique representation for all disciplines. The isogeometric analysis consists in developing methods that use NURBS representations for all design and analysis tasks:

- the geometry is defined by NURBS curves or surfaces;
- the computation domain is defined by planar NURBS surfaces or NURBS volumes instead of discrete meshes;
- the solution fields are obtained by using a finite-element approach that uses NURBS basis functions instead of classical Lagrange polynomials;
- the optimizer controls directly NURBS control points.

This framework allows to compute the analysis solution on the exact geometry (not a discretized geometry), obtain a more accurate solution (high-order approximation), reduce spurious numerical sources of noise that deteriorate convergence, avoid data transfers between the design and analysis phases. Moreover, NURBS representation is naturally hierarchical and allows to perform refinement operations to improve the analysis result.

In finite element analysis (FEA), mesh generation, which generates discrete geometry as computational domain from given CAD object, is a key and the most time-consuming step. In IGA framework, parameterization of computational domain, which corresponds to the mesh generation in FEA, also has some impact on analysis result and efficiency. Moreover, in FEA, one can perform arbitrary refinements on the computational mesh, but in IGA using tensor product B-splines, the refinement is not arbitrary, we can only perform refinement operations in u direction and v direction by knot insertion or degree evaluation. Hence, parameterization of computational domain is more important in IGA.

The parameterization of a computational domain in IGA is determined by control points, knot vectors and the degrees of B-spline objects. For IGA problem of two dimension, the knot vectors and the degree of computational domain are determined by the given boundary curves. Hence, finding the optimal placement of inner control points for a specified physical problem, is a key issue in IGA. A basic requirement of resulting parameterization for IGA is that it doesn't have self-intersections. In this paper, we first propose a linear and easy-to-test sufficient condition for injectivity of planar B-spline parameterization. Then we show that different parameterizations of computational domain has different impact on the simulation results in IGA. For problems with exact solutions, a shape optimization method is proposed to obtain an optimal parameterization of computational domain. Some examples and comparisons are presented based on the heat conduction problem to show the effectiveness of proposed method.

The remainder of the paper is organized as follows. Section 2 reviews the related work in isogeometric analysis. Section 3 proposes the linear sufficient conditions for injectivity of planar B-spline parameterization. Section 4 describes a test IGA model in this paper and shows the impact of different parameterizations of computational domain. Section 5 presents the shape optimization method to obtain an optimal parameterization of a computational domain. Some examples and comparisons are also presented in Section 5. Finally, we conclude this paper and outline the future work in Section 6.

2 Related work

In this section, we will review related works in IGA and parameterization of computational domains.

The concept of IGA was firstly proposed by T.R Hughes et al. [19] in 2005 to achieve the seamless integration of CAD and FEA. Since then, many researchers in the fields of mechanical engineering and geometric modeling were involved in this topic. The current work on isogeometric analysis can be classified into three categories: (1) application of IGA to various simulation problems [2,5,6,10,14,18,20,26,27]; (2) application of various geometric modeling tools to IGA [7,12,23]; (3) accuracy and efficiency improvement of IGA framework by reparameterization and refinement operation [1,3,8,9,15,21,24].

The topic of this paper belongs to the third field. As far as we know, there are few works on the parametrizations of computational domains for IGA. T.Martin et al. [24] proposed a method to fit a genus-0 triangular mesh by B-spline volume parametrization, based on discrete volumetric harmonic functions; this can be used to build computational domains for 3D IGA problems. A variational approach for constructing NURBS parameterization of swept volumes is proposed by M.Aigner et al [1]. Many free-form shapes in CAD systems, such as blades of turbines and propellers, are covered by this kind of volumes. E. Cohen et.al. [8] proposed the concept of *analysis-aware modeling*, in which the parameters of CAD models should be selected to facilitate isogeometric analysis. They also demonstrated the influence of parameterization of computational domains by several examples. In this paper, a method for generating optimal analysis-aware parameterization of computational domain is proposed based on shape optimization method.

3 A linear sufficient condition for injectivity of planar B-spline parameterization

The main idea of the isogeometric approach is to use the same representation for the geometry and the physical solutions we are interested in. Schematically, the geometry Ω involved in the physical problem can be a surface or a volume in a three-dimensional space $\subset \mathbb{R}^3$. Let us call $\mathbf{x} = (x, y, z)$ the coordinates associated to this space. In our case, this geometry will be represented by a parameterization σ for a domain \mathcal{P} of the parameter space. Let us call \mathbf{u} the coordinates of this parameter domain, which could be of dimension 2 for a surface or 3 for a volume. This parameterization will be given by B-spline functions with knots in \mathcal{P} and control points in \mathbb{R}^3 .

The concept of isogeometry consists in representing the physical quantities $\Phi \in \mathbb{R}^p$ on the geometry Ω using the same type of B-spline representation as for the geometry Ω . In other words, given a point $\mathbf{x} = \sigma(\mathbf{u}) \in \Omega$ with $\mathbf{u} \in \mathcal{P}$, we associate to it the physical quantities $\Phi(\mathbf{u})$ where $\Phi(\mathbf{u})$ is a B-spline function with nodes in \mathcal{P} and control points in \mathbb{R}^p . This means that the map $\mathbf{x} \in \Omega \mapsto \Phi \in \mathbb{R}^p$ is defined *implicitly* as $\mathbf{x} \mapsto \Phi \circ \sigma^{-1}(\mathbf{x})$.

Consequently, the framework of isogeometry is thus valid when the parameterization σ of the geometry is *injective* (or bijective on its image). We are going to describe sufficient and easy-to-check conditions for the injectivity of σ . We will consider this problem in the context of finding a “good” parameterization of a domain when its boundary is given. In [22], a general sufficient condition is proposed for injective parameterization.

Proposition 1. *Suppose that σ is a C^1 parameterization from a compact connected domain $\mathcal{P} \subset \mathbb{R}^n$ to a geometry $\Omega \subset \mathbb{R}^n$ and $\partial\mathcal{P}$ is connected. If σ is injective on the boundary $\partial\mathcal{P}$ of \mathcal{P} and its Jacobian J_σ does not vanish on \mathcal{P} , then σ is injective.*

For a parameterization σ from $[a, b] \times [c, d]$ to $\Omega \subset \mathbb{R}^2$, we define the boundary curves as the image of $\{a\} \times [c, d]$, $\{b\} \times [c, d]$, $[a, b] \times \{c\}$, $[a, b] \times \{d\}$ by σ . We say that σ defines a *regular boundary* if these curves do not intersect pairwise, except at their end points and if they have no self-intersection.

As a consequence of the previous proposition, we get the following injectivity test for standard B-spline tensor product parameterization of a planar domain.

Proposition 2. *Let σ be a C^1 parameterization from $[a, b] \times [c, d]$ to $\Omega \subset \mathbb{R}^2$ which defines a regular boundary. If its Jacobian J_σ does not vanish on $[a, b] \times [c, d]$, then σ is injective.*

These tests involve injectivity conditions on the boundary, which can be checked recursively using the same techniques, non-intersection tests for boundary curves and surfaces which are provided for instance by geometric (subdivision) algorithms and the local injectivity condition corresponding to the non-vanishing of the Jacobian. This last condition requires to test on all the domain Ω that the Jacobian does not vanish. Hereafter we propose a sufficient and easy-to-test condition to ensure the local injectivity condition.

We consider first the case of a planar parameterization

$$\sigma : \mathbf{u} \in \mathcal{P} := [a, b] \times [c, d] \mapsto \sigma(\mathbf{u}) := \sum_{0 \leq i \leq l_1, 0 \leq j \leq l_2} \mathbf{c}_{i,j} N_{i,j}(\mathbf{u}),$$

where $\mathbf{c}_{i,j} \in \mathbb{R}^2$ are the control points and $N_{i,j}(\mathbf{u})$ are the B-spline basis functions. The derivative of $\sigma(\mathbf{u})$ with respect to \mathbf{u}_1 can be expressed in terms of the differences $\Delta_{i,j}^1 := \mathbf{c}_{i+1,j} - \mathbf{c}_{i,j}$:

$$\partial_{u_1} \sigma(\mathbf{u}) := \sum_{0 \leq i \leq l_1 - 1, 0 \leq j \leq l_2} \omega_{i,j}^1 \Delta_{i,j}^1 N_{i,j}^1(\mathbf{u}),$$

where $N_{i,j}^1$ is the B-spline basis function with one degree less in \mathbf{u}_1 , $\omega_{i,j}^1$ is a positive factor. We denote by $\mathcal{C}_1(\mathbf{c})$ the convex cone of \mathbb{R}^2 generated by the half rays $\mathbb{R}_+ \cdot \Delta_{i,j}^1$.

Similarly, the derivative of $\sigma(\mathbf{u})$ with respect to \mathbf{u}_2 can be expressed in terms of the differences $\Delta_{i,j}^2 := \mathbf{c}_{i,j+1} - \mathbf{c}_{i,j}$:

$$\partial_{u_2} \sigma(\mathbf{u}) := \sum_{0 \leq i \leq l_1 - 1, 0 \leq j \leq l_2 - 1} \omega_{i,j}^2 \Delta_{i,j}^2 N_{i,j}^2(\mathbf{u}),$$

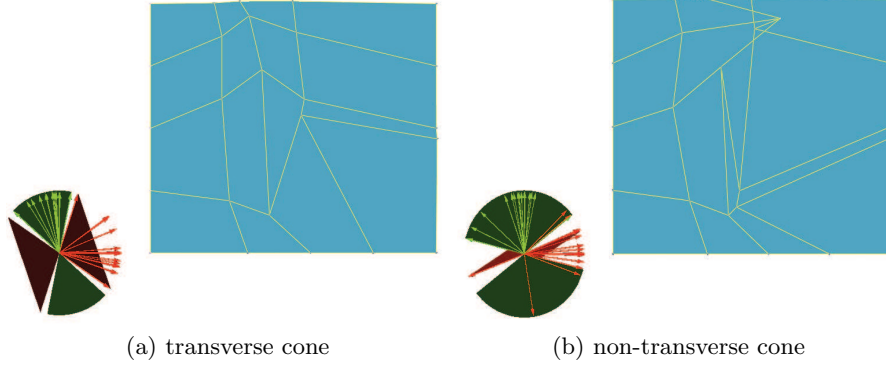


Fig. 1. Injectivity test by cones.

where $N_{i,j}^2$ is the B-spline basis with one degree less in \mathbf{u}_2 , $\omega_{i,j}^2$ is a positive factor. We denote by $\mathcal{C}_2(\mathbf{c})$ the convex cone of \mathbb{R}^2 generated by the half rays $\mathbb{R}_+ \cdot \Delta_{i,j}^2$. If there exist two opposite vectors, which are on a straight line, we define $\mathcal{C}_i(\mathbf{c})$ as a half-plane.

We say that two cones $\mathcal{C}_1, \mathcal{C}_2$ are *transverse* if $\mathbb{R} \cdot \mathcal{C}_1$ and $\mathbb{R} \cdot \mathcal{C}_2$ intersect only at $\{0\}$.

Proposition 3. *Let σ be a B-spline parametrisation, which is at least C^1 from $\mathcal{P} := [a, b] \times [c, d]$ to $\Omega \subset \mathbb{R}^2$ given by the control points \mathbf{c} . If the boundary curves do not intersect and have no self-intersection point and the cones $\mathcal{C}_1(\mathbf{c}), \mathcal{C}_2(\mathbf{c})$ are transverse, then σ is injective on \mathcal{P} .*

Proof. We check first that the transversality of the cones $\mathcal{C}_1(\mathbf{c}), \mathcal{C}_2(\mathbf{c})$ implies that the Jacobian of σ is not vanishing. This jacobian $J_\sigma(\mathbf{u})$ is obtained by taking the determinant $|\partial_{\mathbf{u}_1}\sigma, \partial_{\mathbf{u}_2}\sigma|$ which expands as

$$\sum_{0 \leq i \leq l_1 - 1, 0 \leq j \leq l_2} \sum_{0 \leq i' \leq l_1 - 1, 0 \leq j' \leq l_2 - 1} |\Delta_{i,j}^1, \Delta_{i',j'}^2| \omega_{i,j}^1 \omega_{i',j'}^2 N_{i,j}^1(\mathbf{u}) N_{i',j'}^2(\mathbf{u}).$$

Since the cone $\mathcal{C}_1(\mathbf{c})$ and $\mathcal{C}_2(\mathbf{c})$ are transverse, the determinants $|\Delta_{i,j}^1, \Delta_{i',j'}^2|$ have a constant sign for $\Delta_{i,j}^1 \in \mathcal{C}_1(\mathbf{c})$, $\Delta_{i',j'}^2 \in \mathcal{C}_2(\mathbf{c})$. As the basis functions and the factors are positive, the Jacobian $J_\sigma(\mathbf{u})$ cannot vanish at $\mathbf{u} \in \mathcal{G}$, except if all the $N_{i,j}^1(\mathbf{u}) N_{i',j'}^2(\mathbf{u})$ vanish, which is not possible.

The map σ is locally injective on \mathcal{P} . By Proposition 2, we deduce that σ is globally injective on \mathcal{P} . \square

Fig.1 shows two examples of the injectivity testing method. In Fig.1 (a), it satisfies the sufficient condition in our method, but it does not satisfy the sufficient condition of the method proposed in [17]. Hence, our method is an improved version of the method presented in [17].

Linear constraint for injectivity. This condition can be used to devise an algorithm which constructs an injective parameterization for given boundary control points. We consider first the planar case. Given four planar boundary curves described by the control points $\mathbf{c}_{i,0}, \mathbf{c}_{i,l_2}, \mathbf{c}_{0,j}, \mathbf{c}_{l_1,j}$, with $0 \leq i \leq l_1, 0 \leq j \leq l_2$, we define the boundary cone $\mathcal{C}_1^0(\mathbf{c})$ (resp. $\mathcal{C}_2^0(\mathbf{c})$) as the cone generated by the vectors $\Delta_{i,0}^1(\mathbf{c}), \Delta_{i,l_2}^1(\mathbf{c})$ for $0 \leq i \leq l_1 - 1$ (resp. $\Delta_{0,j}^2(\mathbf{c}), \Delta_{l_2,j}^2(\mathbf{c})$ for $0 \leq j \leq l_2 - 1$). We assume that these boundary curves form a regular boundary and that the two boundary cones $\mathcal{C}_1^0(\mathbf{c}), \mathcal{C}_2^0(\mathbf{c})$ are transverse. $\mathbb{R} \cdot \mathcal{C}_1^0(\mathbf{c})$ is the cone defined by $l_1^+(\mathcal{C}_1^0(\mathbf{c})) \leq 0, l_1^-(\mathcal{C}_1^0(\mathbf{c})) \leq 0$, where l_1^+ and l_1^- are the linear equations defining the boundary of $\mathbb{R} \cdot \mathcal{C}_1^0(\mathbf{c})$. We defined similarly l_2^+, l_2^- for $\mathcal{C}_2^0(\mathbf{c})$.

From Proposition 3, the inner control points $\mathbf{c}_{i,j}$ should satisfy the following linear constraints for injective parameterization:

$$\begin{cases} l_1^+(\mathbf{c}_{i+1,j} - \mathbf{c}_{i,j}) \leq 0, l_1^-(\mathbf{c}_{i+1,j} - \mathbf{c}_{i,j}) \leq 0, 0 \leq i < l_1, 0 < j < l_2 \\ l_2^+(\mathbf{c}_{i,j+1} - \mathbf{c}_{i,j}) \leq 0, l_2^-(\mathbf{c}_{i,j+1} - \mathbf{c}_{i,j}) \leq 0, 0 < i < l_1, 0 \leq j < l_2. \end{cases}$$

This condition provides an easy-to-check method for the injectivity of a parameterization. In Section 5, we will employ this method to check the injectivity of initial parameterization.

4 Isogeometric analysis and parameterization of computational domain

In this section, we aim at presenting the reasons why solutions from IGA depend strongly on the choice of the parameterization. This will be illustrated by considering problems from heat conduction field.

4.1 Test model — heat conduction problem

Given a domain Ω with $\Gamma = \partial\Omega_D \cup \partial\Omega_N$, we consider the following thermal conduction problem:

$$\begin{aligned} \nabla(\kappa(\mathbf{x})\nabla T(\mathbf{x})) &= f(\mathbf{x}) \quad \text{in } \Omega \\ T(\mathbf{x}) &= T_0(\mathbf{x}) \quad \text{on } \partial\Omega_D \\ \kappa(\mathbf{x})\frac{\partial T}{\partial \mathbf{n}}(\mathbf{x}) &= \Phi_0(\mathbf{x}) \quad \text{on } \partial\Omega_N, \end{aligned} \tag{1}$$

where \mathbf{x} are the Cartesian coordinates, T represents the temperature field and κ the thermal conductivity. Dirichlet and Neumann boundary conditions are applied on $\partial\Omega_D$ and $\partial\Omega_N$ respectively, T_0 and Φ_0 being the imposed temperature and thermal flux (\mathbf{n} unit vector normal to the boundary). f is a user-defined function that allows to generate problems with an analytical solution, by adding a source term to the classical heat conduction equation.

According to a classical variational approach, we seek for a solution $T \in H^1(\Omega)$, such as $T(\mathbf{x}) = T_0(\mathbf{x})$ on $\partial\Omega_D$ and:

$$\int_{\Omega} \nabla(\kappa(\mathbf{x})\nabla T(\mathbf{x})) \psi(\mathbf{x}) d\Omega = \int_{\Omega} f(\mathbf{x}) \psi(\mathbf{x}) d\Omega \quad \forall \psi \in H_{\partial\Omega_D}^1(\Omega),$$

where $\psi(\mathbf{x})$ are test functions. By integrating by parts and using boundary conditions, we obtain:

$$- \int_{\Omega} \kappa(\mathbf{x}) \nabla T(\mathbf{x}) \nabla \psi(\mathbf{x}) d\Omega + \int_{\partial\Omega_N} \Phi_0(\mathbf{x}) \psi(\mathbf{x}) d\Gamma = \int_{\Omega} f(\mathbf{x}) \psi(\mathbf{x}) d\Omega. \quad (2)$$

According to the IGA paradigm, the temperature field is represented using B-spline basis functions. For a 2D problem, we have:

$$\mathcal{T}(\xi, \eta) = \sum_{i=1}^{n_i} \sum_{j=1}^{n_j} \hat{N}_i^{p_i}(\xi) \hat{N}_j^{p_j}(\eta) T_{ij},$$

where \hat{N}_i functions are B-Spline basis functions and $\mathbf{u} = (\xi, \eta) \in \mathcal{P}$ are domain parameters. Then, we define the test functions $\psi(\mathbf{x})$ in the physical domain such as:

$$N_{ij}(\mathbf{x}) = N_{ij}(x, y) = N_{ij}(\mathcal{T}(\xi, \eta)) = \hat{N}_{ij}(\xi, \eta) = \hat{N}_i^{p_i}(\xi) \hat{N}_j^{p_j}(\eta).$$

The weak formulation Eq. 2 reads:

$$\sum_{k=1}^{n_k} \sum_{l=1}^{n_l} T_{kl} \int_{\Omega} \kappa(\mathbf{x}) \nabla N_{kl}(\mathbf{x}) \nabla N_{ij}(\mathbf{x}) d\Omega = \int_{\partial\Omega_N} \Phi_0(\mathbf{x}) N_{ij}(\mathbf{x}) d\Gamma + \int_{\Omega} f(\mathbf{x}) N_{ij}(\mathbf{x}) d\Omega$$

Finally, we obtain a linear system similar to that resulting from the classical finite-element methods, with a matrix and a right-hand side defined as:

$$\begin{aligned} M_{ij,kl} &= \int_{\Omega} \kappa(\mathbf{x}) \nabla N_{kl}(\mathbf{x}) \nabla N_{ij}(\mathbf{x}) d\Omega \\ &= \int_{\mathcal{P}} \kappa(\mathcal{T}(\mathbf{u})) \nabla_{\mathbf{u}} \tilde{N}_{kl}(\mathbf{u}) B(\mathbf{u})^T B(\mathbf{u}) \nabla_{\mathbf{u}} \tilde{N}_{ij}(\mathbf{u}) J(\mathbf{u}) d\mathcal{P} \\ S_{ij} &= \int_{\partial\Omega_N} \Phi_0(\mathbf{x}) N_{ij}(\mathbf{x}) d\Gamma + \int_{\Omega} f(\mathbf{x}) N_{ij}(\mathbf{x}) d\Omega \\ &= \int_{\partial\mathcal{P}_N} \Phi_0(\mathcal{T}(\mathbf{u})) \tilde{N}_{ij}(\mathbf{u}) J(\mathbf{u}) d\tilde{\Gamma} + \int_{\mathcal{P}} f(\mathcal{T}(\mathbf{u})) \tilde{N}_{ij}(\mathbf{u}) J(\mathbf{u}) d\mathcal{P}. \end{aligned}$$

where J is the Jacobian of the transformation, B^K is the transposed of the inverse of the Jacobian matrix. The above integrations are performed in the parameter space using classical Gauss quadrature rules.

Starting from a planar B-spline surface as computational domain, a general framework of an isogeometric solver for thermal conduction problem (1) has been implemented as plugins in the AXEL¹ platform, yielding a B-spline surface as solution field. Gauss-Seidel algorithm is employed to solve the linear system. In order to improve the simulation results, refinement operation can be performed for two parametric directions. Additional details concerning the methods employed can be found in [13].

¹ <http://axel.inria.fr/>

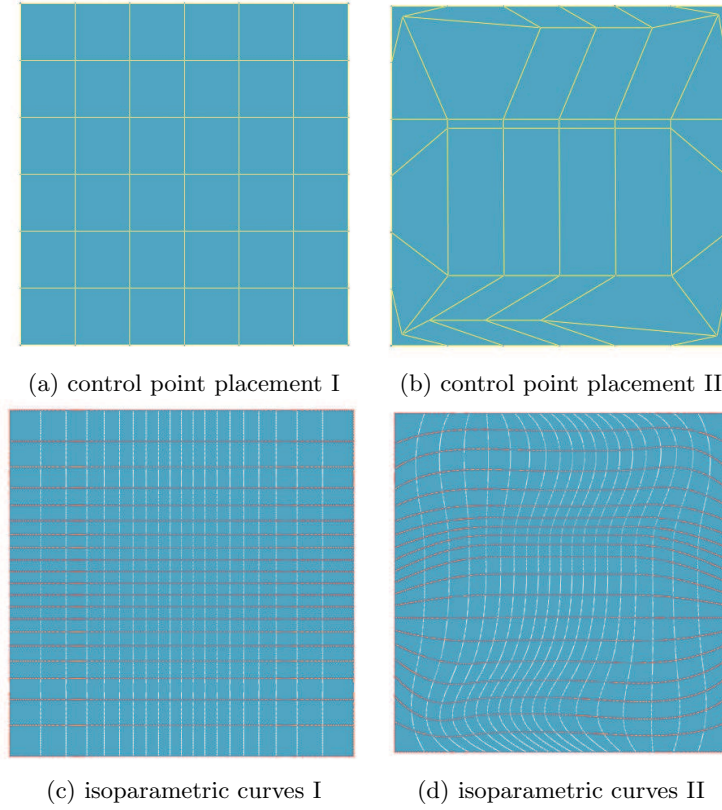


Fig. 2. Two different parameterization of computational domain. (a),(b): two different placements of inner control points. (c), (d): isoparametric curves on the computational domain with respect to the control points placements in (a) and (b).

4.2 Isogeometric analysis with different parameterization

As mentioned above, given four boundary planar B-spline curves, we can construct various planar B-spline surface with different parameterizations. For Example I in Fig. 2, we present two kinds of parameterization for a computational domain $\Omega(x, y) = [0, 6] \times [0, 6]$ represented by cubic B-spline surfaces, where the knot vectors in u and v directions are both $\{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$. Fig. 2 (a) and Fig. 2(b) present two different placements of inner control points, Fig. 2 (c) and Fig. 2(d) show the isoparametric curves on the computational domain with respect to different placements of inner control points.

We test these two parameterizations based on the heat conduction problem in (1) with source term

$$\mathbf{f}(x, y) = -\frac{4}{9} \sin\left(\frac{\pi x}{3}\right) \sin\left(\frac{\pi y}{3}\right). \quad (3)$$

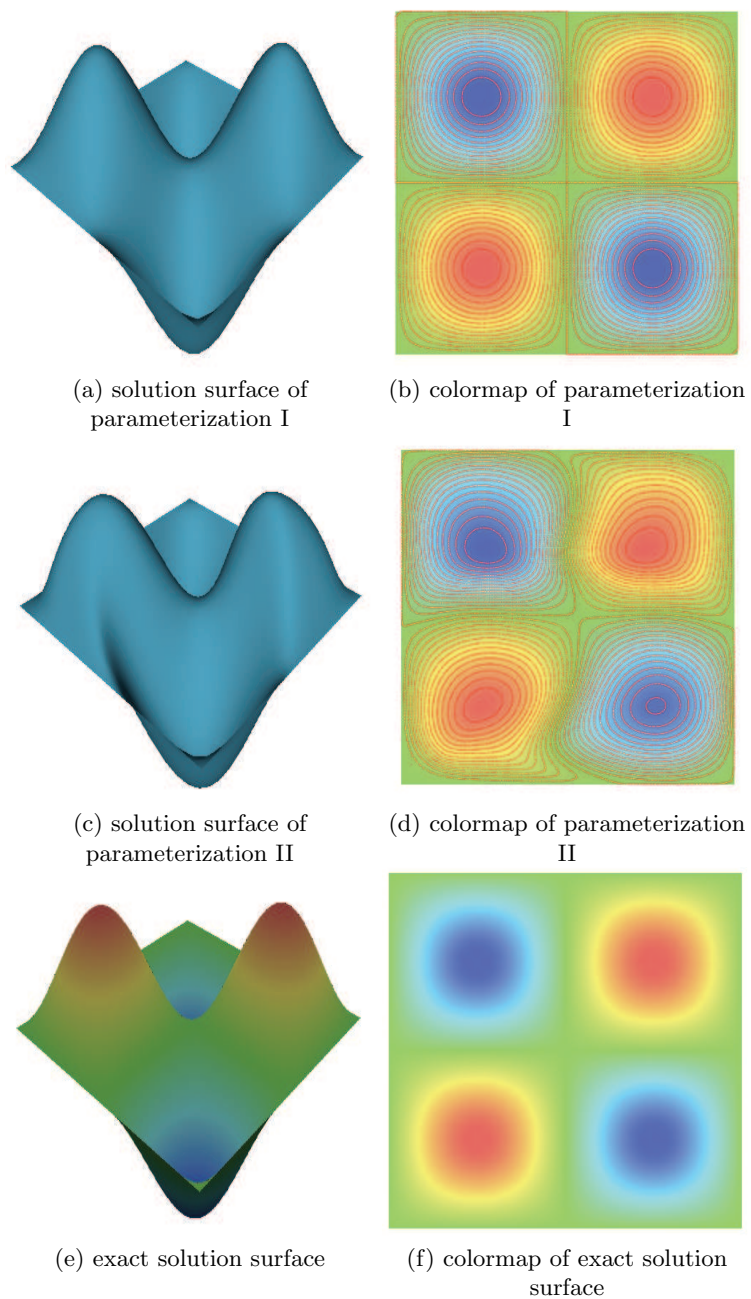


Fig. 3. Simulation result and exact solution.

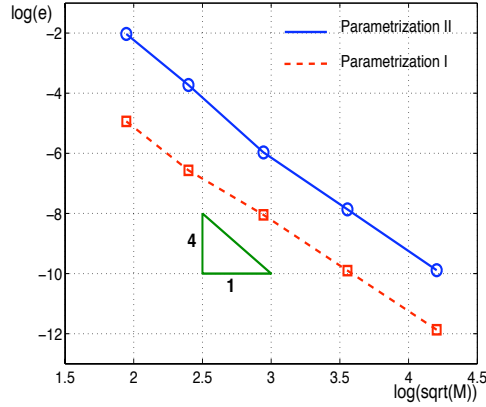


Fig. 4. Error analysis with the curve $(\log \sqrt{M}, \log e)$, where M is the number of control points in each refinement.

For this problem with boundary condition $\mathbf{T}_0(\mathbf{x}) = 0$ and $\Phi_0(\mathbf{x}) = 0$, the exact solution over the computational domain $[0, 6] \times [0, 6]$ is

$$\mathbf{T}(x, y) = 2 \sin\left(\frac{\pi x}{3}\right) \sin\left(\frac{\pi y}{3}\right). \quad (4)$$

Fig.3 (a) and Fig.3 (b) show the approximate solution surface, color map and iso-temperature lines with respect to parameterization I; Fig.3 (c) and Fig.3 (d) show the approximate solution surface, color map and iso-temperature lines with respect to parameterization II. In Fig.3 (e) and Fig.3(f), the exact solution surface and its colormap are presented. Obviously, parameterization I is better than parameterization II for this specified heat conduction problem.

Refinement via knot insertion is an efficient operation to improve the result of isogeometric analysis. We compare the error history during refinement operation for these two different parameterization in Fig.4. The error is computed in relative L_2 norm as follows [23]

$$e = \sqrt{\frac{\int_{\Omega} (\mathbf{T} - \tilde{\mathbf{T}})^T (\mathbf{T} - \tilde{\mathbf{T}}) d\Omega}{\int_{\Omega} \mathbf{T}^T \mathbf{T} d\Omega}},$$

where \mathbf{T} is the exact solution and $\tilde{\mathbf{T}}$ is the approximate solution. From Fig.4, we see that different parameterizations have different impact on the final result after refinement operation. Though the convergence rates of the two different parameterization are in good agreement with theoretical convergence (4 for cubic parameterization), for an error value about 5×10^{-5} , parameterization I requires 35×35 control points, and parameterization II requires 67×67 control points. One reasonable explanation is that with B-spline tensor product surfaces, we

can only perform the refinement operations along the parametric directions in IGA, hence it is more restricted than the refinement of a mesh in FEA.

The above example and its analysis show that good parameterization of computational domain is a key issue for IGA. In the next section, we will propose a shape optimization method to construct optimal parameterization of a computational domain.

5 Optimization method for parametrization of computational domain

5.1 Problem statement

The problem studied in this section can be stated as follows: given four coplanar boundary B-spline curves, find the inner control points such that the parameterization of a computational domain is optimal for an IGA problem with known exact solution.

5.2 Shape optimization method

The shape optimization problem consists in finding the shape which is optimal in that it minimizes a certain cost function while satisfying given constraints. The purpose of shape optimization in CAE is to optimize the CAD object for some physical problem, and the design variables are the control points of the CAD object. For 2D isogeometric shape optimization problem, the design variables are the control points of boundary B-spline curves.

Inspired from the idea of shape optimization, in order to obtain optimal parameterization of computational domain, we should let the inner control points, rather than boundary control points, be the design variables for the shape optimization, and find the best placement of inner control points to make the value of a cost function as small as possible.

Initial construction of inner control points. As the shape optimization problem, we need to construct an initial placement of inner control points as starting point in the iteration process. We rely on the discrete Coons method presented in [16] to generate inner control points as initial value from boundary control points.

Given the boundary control points $\mathbf{P}_{0,j}, \mathbf{P}_{n,j}, \mathbf{P}_{i,0}, \mathbf{P}_{i,m}$, $i = 0, \dots, n, j = 0, \dots, m$, the inner control points $\mathbf{P}_{i,j}$ ($i = 1, \dots, n-1, j = 1, \dots, m-1$) can be constructed by the discrete Coons method as follows:

$$\begin{aligned} \mathbf{P}_{i,j} = & (1 - \frac{i}{n})\mathbf{P}_{0,j} + \frac{i}{n}\mathbf{P}_{n,j} + (1 - \frac{j}{m})\mathbf{P}_{i,0} + \frac{j}{m}\mathbf{P}_{i,m} \\ & - [1 - \frac{i}{n} \quad \frac{i}{n}] \begin{pmatrix} \mathbf{P}_{0,0} & \mathbf{P}_{0,m} \\ \mathbf{P}_{n,0} & \mathbf{P}_{n,m} \end{pmatrix} \begin{pmatrix} 1 - \frac{j}{m} \\ \frac{j}{m} \end{pmatrix} \end{aligned}$$

Remarks 1. Since the sum of the coefficients equals 1, the resulting inner control points lie in the convex hull of the boundary control points.

Remarks 2. For some given boundary curves, this construction maybe cause some self-intersections, which is improper parameterization for IGA. We use the linear injectivity condition proposed in Section 3 to check the injectivity of initial parameterization. If it does not satisfy the condition, the linear programming method is used to produce initial parameterization.

Optimization method. In the proposed approach, we minimize the error computed from the IGA solution and the exact solution, by moving inner control points of the computational domain. Therefore, we consider as optimization variables the coordinates of the inner control points and as cost function the error of the IGA solution. The optimization algorithm used for this study is a classical steepest-descent method in conjunction with a back-tracking line-search. For this exercise, the gradient of the cost function is approximated using a centered finite-differencing scheme.

Each iteration k of the optimization algorithm can be summarized as follows, starting from a point x_k in the variable space:

1. Evaluation of perturbed points $x_k + \epsilon e_k$
2. Estimation of the gradient $\nabla f(x_k)$ by finite-difference
3. Define search direction $d_k = -\nabla f(x_k)$
4. Line search : find ρ such as $f(x_k + \rho d_k) < f(x_k)$

These steps are carried out until a stopping criterion is satisfied.

5.3 Examples and comparison

In this section, we will present some parameterization results and compare them with the initial solution based on the test model (1).

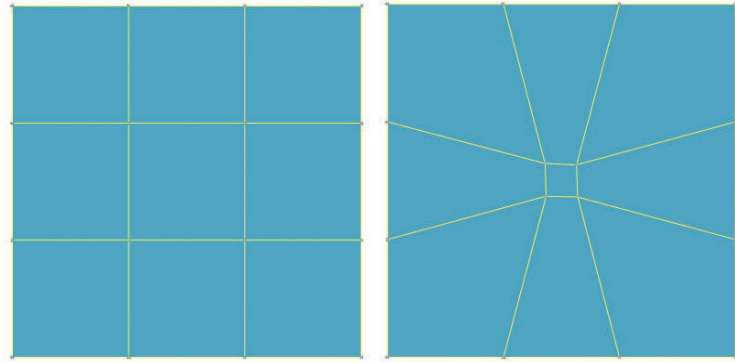
Example II . The second example is for the parameterization of domain $\Omega = [0, 3] \times [0, 3]$ by cubic Bézier surface. The corresponding source term and exact solution is presented in (3) and (4). The parameterization result and comparison with initial parameterization are shown in Fig.5. The initial error is reduced by 24.52% as shown in Fig.5 (e). The final parameterization is clearly better than the initial parameterization during refinement operations as presented in Fig.5 (f).

Example III . The next example is for the parameterization of the domain

$$\Omega(x, y) = \{(x, y) | -1 \leq y \leq x^2, 0 \leq x \leq 1\}$$

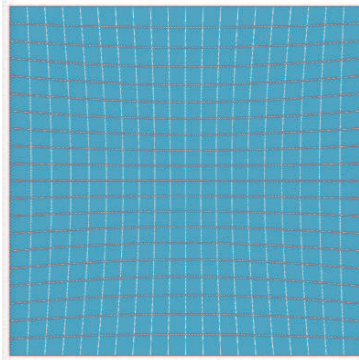
by Bézier surface with degree 3×6 . The parabola is represented by degenerate cubic Bézier curve. For the problem with boundary condition $\mathbf{T}_0(\mathbf{x}) = 0$ and $\Phi_0(\mathbf{x}) = 0$ in (1), we can construct an exact solution $\mathbf{T}(x, y)$ as follows

$$\mathbf{T}(x, y) = \sin(\pi(y - x^2)) \sin(\pi x) \sin(\pi y)$$

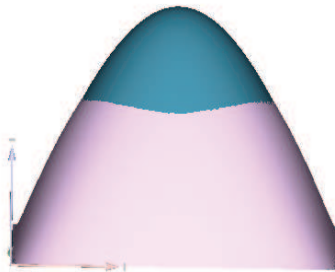


(a) initial control mesh and surface

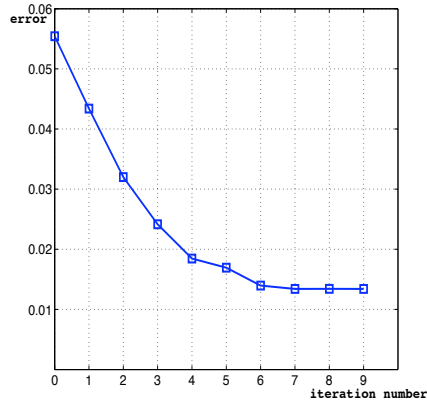
(b) final control mesh and surface



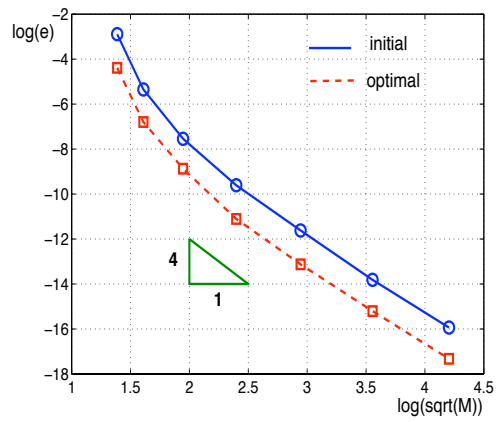
(c) final parametrization



(d) initial solution surface(red) and final solution surface(blue)



(e) error history during the optimization process



(f) error history during the refinement process

Fig. 5. Example II.

The initial placement of inner control points is produced by the discrete Coons method as shown in Fig.6 (a). The final parameterization results and some comparisons are also shown in Fig.6. We can find that there are some self-intersections on the control mesh in Fig.6 (b). However, there is no self-intersection on the final parameterization as shown in Fig.6 (c). During the optimization, the initial error is reduced by 14.65% as shown in Fig.6 (g). The error history during refinement operation is presented in Fig.6 (h).

Example IV . The final example is for the parametrization of the domain $\Omega = [0, 3] \times [0, 6]$ by cubic B-spline surface with knot vector $\{0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4\}$ in the u direction and knot vector $\{0, 0, 0, 0, 1, 1, 1, 1\}$ in the v direction. The corresponding source term and exact solution is presented in (3) and (4). The initial placement of inner control points is non-uniform as shown in Fig.7 (a), and the final parametrization result and some comparison are also shown in Fig.7. During the optimization, the initial error is reduced by 3.31% as shown in Fig.7 (g). The error history during refinement operation is presented in Fig.7 (h).

6 Conclusion and future work

Parameterization of computational domains is the first step in an IGA process. In this paper, we show that for different parameterizations of a computational domain, different simulation results can be obtained. Based on this observation and inspired by shape optimization, an approach for optimal parameterization of computational domain is proposed. We also proposed a linear and easy-to-check sufficient condition for injectivity of planar B-spline parameterization. Several examples are presented to illustrate the effectiveness of the proposed method. As shown in Fig.8, a user-friendly interface for isogeometric solver and optimizer is implemented as plugin in the AXEL platform.

The proposed method will be generalized and tested on 3D IGA problems with exact solutions in the future. The construction of a proper parameterization of computational domain for general problem, in which the exact solution is unknown, is also a part of our ongoing work. One possible way is to find an accurate posteriori error estimation method for IGA, and perform the optimization based on this estimation. We will discuss this topic in another paper.

Acknowledgments. The authors are supported by the 7th Framework Program of the European Union, project SCP8-218536 “EXCITING”.

References

1. M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon and A.-V. Vuong. Swept volume parametrization for isogeometric analysis. In E. Hancock and R. Martin (eds.), *The Mathematics of Surfaces (MoS XIII 2009)*, LNCS vol. 5654(2009), Springer, pp.19-44

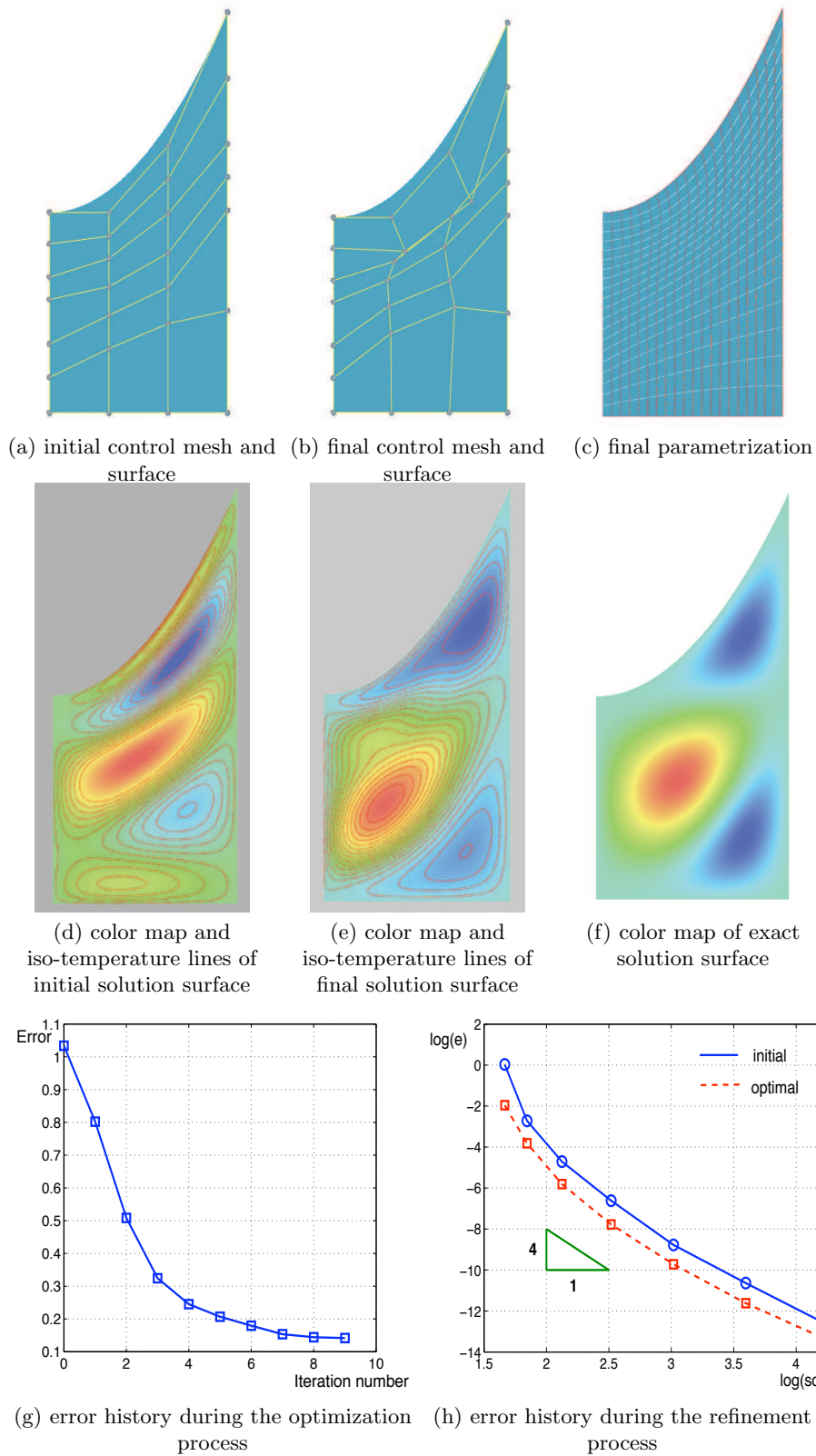


Fig. 6. Example III.

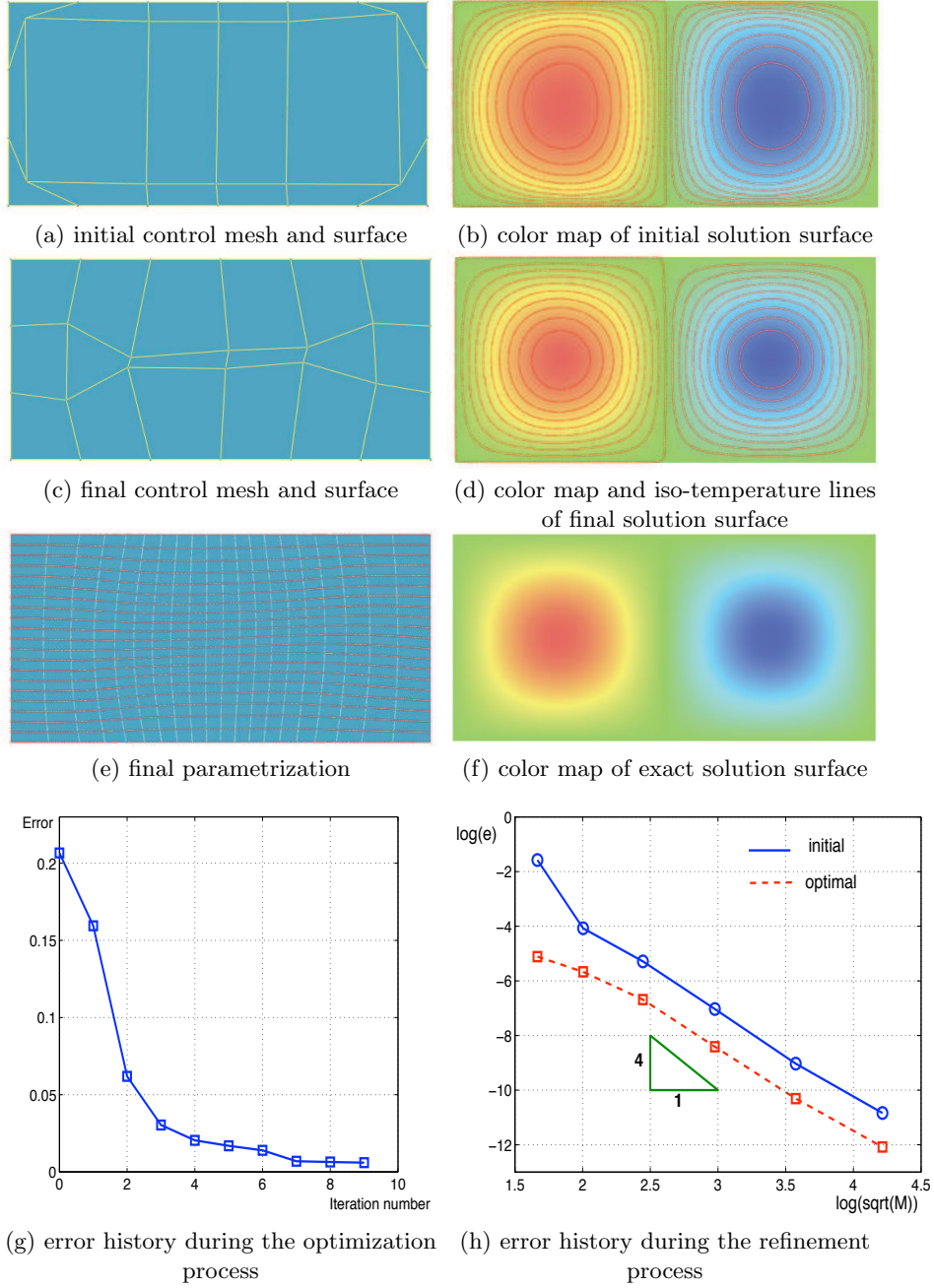


Fig. 7. Example IV.

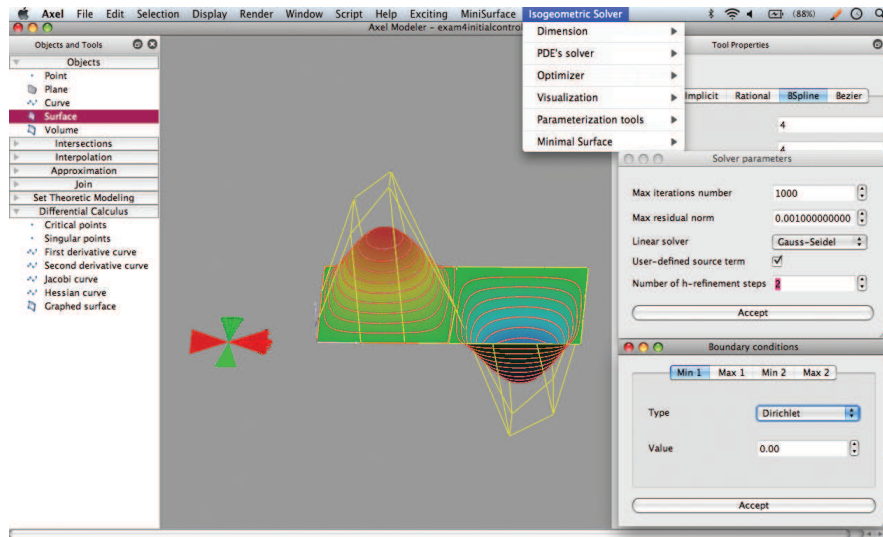


Fig. 8. Interface for isogeometric solver in AXEL.

2. F. Auricchio, L.B. da Veiga, A. Buffa, C. Lovadina, A. Reali, and G. Sangalli. A fully locking-free isogeometric approach for plane linear elasticity problems: A stream function formulation. *Computer Methods in Applied Mechanics and Engineering*, 197:160-172, 2007.
3. Y. Bazilevs, L. Beirão de Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for refined meshes. *Mathematical Models and Methods in Applied Sciences*, 6:1031-1090, 2006.
4. Y. Bazilevs, V.M. Calo, T.J.R. Hughes, and Y. Zhang. Isogeometric fluid structure interaction: Theory, algorithms, and computations. *Computational Mechanics*, 43:3-37, 2008.
5. Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes. Isogeometric fluid structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38:310-322, 2006.
6. Y. Bazilevs and T.J.R. Hughes. NURBS-based isogeometric analysis for the computation of flows about rotating components. *Computational Mechanics*, 43:143-150, 2008.
7. Y. Bazilevs, V.M. Calo, J.A. Cottrell, J. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-Splines. *Computer Methods in Applied Mechanics and Engineering*, to appear.
8. E. Cohen, T. Martin, R.M. Kirby, T. Lyche and R.F. Riesenfeld, Analysis-aware Modeling: Understanding Quality Considerations in Modeling for Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, to appear.
9. J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160-4183, 2007.
10. J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257-5296, 2006.

11. T. Dokken, V. Skytt, J. Haenisch, K. Bengtsson. Isogeometric representation and analysis—bridging the gap between CAD and analysis. 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition, 5-8 January 2009, Orlando, Florida.
12. M. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, to appear.
13. R. Duvigneau. An introduction to isogeometric analysis with application to thermal conduction. INRIA Research Report RR-6957, June 2009
14. T. Elguedj, Y. Bazilevs, V.M. Calo, and T.J.R. Hughes. \bar{B} and \bar{F} projection methods for nearly incompressible linear and non-linear elasticity and plasticity using higher-order NURBS elements. *Computer methods in applied mechanics and engineering*, 197:2732-2762,2008.
15. J.A. Evans, Y. Bazilevs, I. Babuka, T.J.R. Hughes. n -Widths, supinfs, and optimality ratios for the k-version of the isogeometric finite element method, *Computer Methods in Applied Mechanics and Engineering*, 198 (2009) 1726- 1741.
16. Gerald Farin, Dianne Hansford. Discrete coons patches[J]. *Computer Aided Geometric Design*, 1999, 16(7):691-700.
17. James E. Gain, Neil A. Dodgson. Preventing self-Intersection under free-form deformation. *IEEE Transactions on Visualization and Computer Graphics*, 7(4), pp. 289-298, Oct.-Dec. 2001.
18. H. Gomez, V.M. Calo, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of the Cahn-Hilliard phase-field model. *Computer Methods in Applied Mechanics and Engineering*, 197:4333-4352, 2008.
19. T.J.R. Hughes, J.A. Cottrell, Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 194, 39-41, pp 4135-4195, 2005.
20. T.J.R. Hughes, A. Realli, and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. *Computer methods in applied mechanics and engineering*, 197: 4104-4124, 2008.
21. T.J.R. Hughes, A. Realli, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, to appear.
22. H. Kestelman. Mappings with non-vanishing Jacobian. *Amer. Math. Monthly* 78, 662663, 1971.
23. H.J. Kim, Y.D Seo and S.K Youn. Isogeometric analysis for trimmed CAD surfaces . *Computer Methods in Applied Mechanics and Engineering*, to appear.
24. T. Martin, E. Cohen, and R.M. Kirby. Volumetric parameterization and trivariate B-spline fitting using harmonic functions. *Computer Aided Geometric Design*, 26(6):648-664, 2009.
25. R. Sevilla, S. Fernandes-Mendez, and A. Huerta. NURBS-enhanced finite element method for Euler equations. *International Journal for Numerical Methods in Fluids*, 57: 1051-1069, 2008.
26. W.A. Wall, M.A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197:2976-2988, 2008.
27. Y. Zhang, Y. Bazilevs, S. Goswami, C. Bajaj, and T.J.R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196:2943-2959, 2007.