Parameterization of Contractible Domains Using Sequences of Harmonic Maps

Thien Nguyen and Bert Jüttler

Johannes Kepler University, Linz, Austria

Abstract. In this paper, we propose a new method for parameterizing a contractible domain (called the computational domain) which is defined by its boundary. Using a sequence of harmonic maps, we first build a mapping from the computational domain to the parameter domain, i.e., the unit square or unit cube. Then we parameterize the original domain by spline approximation of the inverse mapping. Numerical simulations of our method were performed with several shapes in 2D and 3D to demonstrate that our method is suitable for various shapes. The method is particular useful for isogeometric analysis because it provides an extension from a boundary representation of a model to a volume representation.

1 Introduction

Parameterization is one of the classical topics and attracts a lot of research in computer graphics and geometric modeling because it plays a central role in various applications such as texture mapping or morphing, to name a few. Even though there are many powerful methods to deal with this problem for surfaces, only few results have been achieved for volumes. In fact, due to computational complexity, volume parameterization is considered a big challenge. In general, methods proposed for this problem exploit information about representation and topology of volumes such as tetrahedral mesh volumes or swept volumes. Our work investigates more general problems in which objects are given in boundary representation.

Assume that we are given a contractible domain defined by its boundary. We also assume that we are able to divide the boundary into several patches, i.e., four curves in plane or six surfaces in three-dimensional space. Our goal is to construct a spline representation for the domain. To achieve that, our method creates a mapping from the domain to the unit square or unit cube, see Figure 1. The mapping should be regular or injective. We attain this by solving several variational problems. Finally, the spline approximation for the inverse of the mapping is found by least squares fitting.

2 Related Work

Isogeometric Analysis (IGA for short) was invented by Hughes et al. [12] in 2005 as a promising approach to bridge the gap between numerical simulation and computer aided design (CAD). Immediately, it attracted a lot of

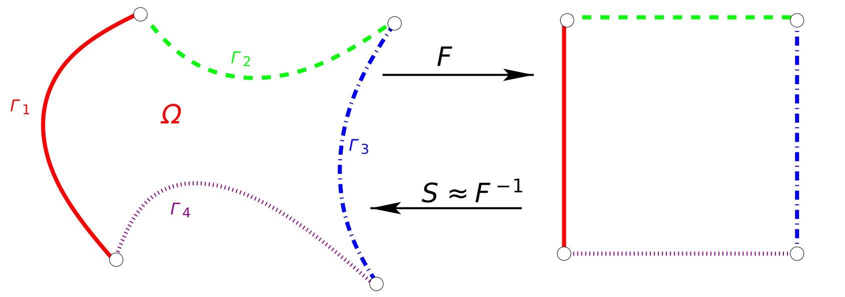


Fig. 1: Mapping from a planar domain to the unit square: the same style curves should be mapped to each other.

research interest from both numerical analysis and geometric modeling communities because of some important advantages. The first advantage of IGA is that the computational domains are solid objects which are represented by tensor product B-splines or NURBS. It is superior to classical finite element methods which work on approximate domains represented by discrete meshes. The second advantage of IGA is that the discretization space is spanned by B-splines or NURBS basis and therefore it provides refinement possibilities as proposed in [12]. So, refinement does not require interaction with the original CAD model as finite element methods, but only requires simple operators such as knot insertion or degree elevation. Due to these properties, the authors called the method isogeometric analysis, i.e., geometry is preserved at all levels of refinement.

Nevertheless, there are some difficulties arising when one wants to use IGA to solve a particular problem. One of the problems is that IGA requires solid models represented by tensor product NURBS, i.e, a NURBS mapping from a parametric domain to a physical domain should cover the entire domain. However, CAD systems usually only provide information on the boundary, i.e, mappings from the parametric domains to the boundary of the models [3]. This observation motivates our research to find a spline representation of the solid models which are only defined by their boundary representation.

Ad hoc parameterization. The problem of parameterizing a domain defined by boundary curves (or surfaces) can be traced back to the well known work by Gordon and Coons [11]. This method uses blending functions to define a Boolean sum operator interpolating boundary curves. It provides reasonable parameterization for quadrilateral patches with nice shapes. Inspired by this direction, there is a later work by Farin on discrete Coons patches [7]. However, because of its simplicity this method may not be suitable for singular cases and complicated shapes. Recently, Xu et al. have used the discrete Coons method combined with shape optimization method to find optimal parameterizations as a prerequisite of IGA for planar shapes [18].

In volume parameterization motivated by IGA, to the best of our knowledge, there are only two remarkable results. The first is the paper by Martin et al. [15] where the authors proposed a parameterization method for a generalized cylinder-type volume defined by a tetrahedral mesh. This method bases on discrete volumetric harmonic functions and solves several Laplace equations. After remeshing the tetrahedral mesh to the hexahedral mesh, a trivariate B-spline for the solid model is generated by an iterative fitting method. The second is the paper by Aigner et al. [1] where the authors proposed a parameterization method for swept volumes which cover many free-form shapes in CAD system like blades or propellers. In this method, a spline approximation for a solid model can be found by solving a minimization problem with several penalty terms corresponding to particular features of the shape.

In Voruganti et al. [17], a method is proposed to build a bijective map between a genus-zero domain in three-dimensional space and the unit sphere, which partially inspired our approach. The authors first choose a point inside the domain, called "shape center", and compute a potential function by solving the Laplace equation with Dirichlet boundary conditions which are zero at the shape center and one on the boundary. Then, streamlines, which are the curves joining each boundary point to the shape center, are build so that they intersect the level sets of the computed function orthogonally. This results in a parameterization of the unit sphere by means of spherical coordinates.

3 Harmonic Maps

Recently, there has been much research interest in harmonic maps and their applications in computer graphics [2, 5, 8]. Harmonic maps are used to find mappings between two manifolds due to their beautiful properties. It is well-known that any conformal mapping between 2-dimensional manifolds is harmonic. Moreover, if we consider the mappings between a surface in $\mathbb{R}^3$ and a fixed boundary domain in $\mathbb{R}^2$, harmonic and conformal mappings are the same [10]. A map u between Riemannian manifolds (M, g) and (N, h) is called *p-harmonic* if it is a critical point of the *p-harmonic* energy

$$E_p(u) = \int_M ||\nabla_M u||^p \mathrm{d}volM$$

where $||\nabla_M u||$ is the length of the differential in M. If $p = 2$, we call it simply harmonic map. Results concerning existence and regularity of the solution of the above problem are described by Jost [13]. In particular, we are interested in the maximum principle of the solution in the case that M has a boundary and $N = \mathbb{R}$. Then if u is a harmonic map, it attains maxima or minima values on the boundary. So, we see that the level sets of u on M are contractible hypersurfaces. In our method, we use the following particular cases of manifolds M and N:

1. $M = \bar{M}_0$ and M_0 is an open subset of $\mathbb{R}^n$ ($n = 2$ or 3) and $N \subseteq \mathbb{R}$ (both M and N with Euclidean metrics): In this case, the harmonic energy has the form
$$E_2(u) = \int_M ||\nabla u||^2 \mathrm{d}\mathbf{x}$$

where ∇ is the gradient operator and integration has the usual meaning in $\mathbb{R}^n$.

2. M is a Riemannian manifold in $\mathbb{R}^n$ ($n = 2$ or 3) which is given by the zeroAdditional integrals over the boundary are used to achieve a good approximation level set of some functions, and $N \subseteq \mathbb{R}$: In this case, the harmonic energy has the form

$$E_2(u) = \int_M ||\nabla_M u||^2 \mathrm{d}volM$$

Here, $\nabla_M u$ is the covariant derivative of u with respect to M which is simply the projection of gradient of u in $\mathbb{R}^n$ on the tangent space of M [2].

4 Our Framework

For the sake of simplicity, we describe our framework only for the planar case. It can be naturally extended to the three-dimensional case by similar ideas. Recall that we are given a contractible computational domain Ω defined by a closed piece-wise smooth curve. Assume that we are able to divide the curve into 4 curves. In fact, this can be done simply by picking 4 points on the curve. Our framework consists of 2 main steps.

4.1 Step 1: Finding a Mapping $F : \Omega \to [0,1]^2$

The mapping F should map the boundary of Ω to the boundary of the unit square and might provide constant parametric speed along boundary. In coordinates, F can be written as $F = (f, g)^T$. In order to use the mapping F in step 2, F should be regular or injective. In other words, the determinant of the Jacobian of F should not vanish in Ω or equivalently, ∇f and ∇g should be linearly independent on Ω. In order to achieve this goal, we propose a 2-step method, namely, we first find f then use f to find g.

Step 1.1 The scalar function $f : \Omega \to [0, 1]$ can be found by minimizing its Dirichlet energy

$$E_2(f) = \int_\Omega ||\nabla f||^2 \mathrm{d}\mathbf{x} \to \min_f \tag{1}$$

subject to the boundary conditions

$$\begin{aligned} f(x, y) &= 0 \text{ on } \Gamma_1 \\ f(x, y) &= 1 \text{ on } \Gamma_3 \\ f(x, y) &= s(x, y) \text{ on } \Gamma_2 \text{ and } \Gamma_4 \end{aligned} \tag{2}$$

where $s(x, y)$ is any function that maps points on the curves Γ_2 or Γ_4 to $[0, 1]$ and is strictly increasing in the direction from Γ_1 to Γ_3. In our computations,

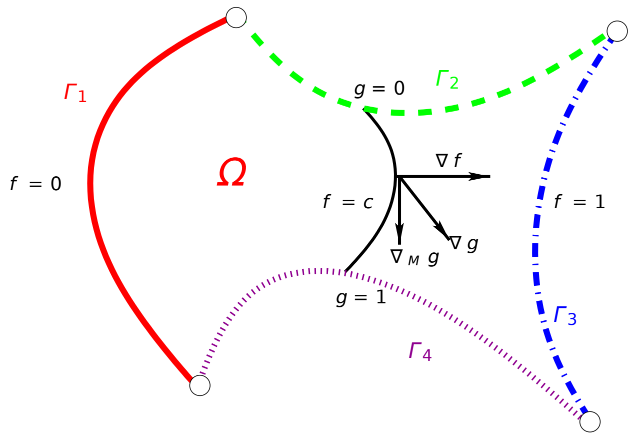


Fig. 2: Covariant gradient of g on a level set of f

$s(x, y)$ restricted to a curve is the arc-length function scaled to $[0, 1]$. On the other hand, we can choose Neumann boundary conditions, i.e., we can set $<\nabla f, n >= 0$ on $\mathit{\Gamma}_2$ or $\mathit{\Gamma}_4$ where n is the normal vector at corresponding point. However, we observed in our computations that this boundary conditions lead to unsatisfactory solutions.

Step 1.2 Once f is found, the scalar function g is generated such that g minimizes the harmonic energy on the level sets of f. More precisely, we define $M = f^{-1}(c) \cap \mathit{\Omega}$ for some constant c and we aim at solving the following minimization problem

$$E_2(g) = \int_M ||\nabla_M g||^2 \mathrm{d}volM \rightarrow \min_g$$

subject to the boundary conditions

$$g(x, y) = 0 \text{ if } (x, y) \in \mathit{\Gamma}_4$$
$$g(x, y) = 1 \text{ if } (x, y) \in \mathit{\Gamma}_2.$$

Note that there are only 2 points in the boundary conditions which are the intersections of M with 2 curves $\mathit{\Gamma}_2$ and $\mathit{\Gamma}_4$. The covariant gradient of g on M is the projection of the gradient of g onto the tangent line with the normal vector ∇f, see Figure 2. Its length can be calculated by

$$||\nabla_M g||^2 = ||\nabla g - \langle \nabla g, \nabla f \rangle \frac{\nabla f}{||\nabla f||^2}||^2.$$

Hence, we can rewrite the harmonic energy as

$$\begin{aligned} E_2(g) &= \int_M ||\nabla g - \langle \nabla g, \nabla f \rangle \frac{\nabla f}{||\nabla f||^2}||^2 \mathrm{d}volM \\ &= \int_{\mathit{\Omega}} ||\nabla g - \langle \nabla g, \nabla f \rangle \frac{\nabla f}{||\nabla f||^2}||^2 \delta(f - c) ||\nabla f|| \mathrm{d}\mathbf{x} \qquad (3) \end{aligned}$$

where $\delta(.)$ is the Dirac delta function in the sense of distributions. In the last equality, we use the formula

$$\int_M \mathcal{R} \mathrm{d}volM = \int_\Omega \mathcal{R}\delta(f-c)||\nabla f||\mathrm{d}\mathbf{x}$$

where $\mathcal{R}$ is some function. Next, by means of minimizing simultaneously the harmonic energy for all level sets of f contained in Ω, we can eliminate the Dirac delta function in the Eq. (3) and formulate our minimization problem as

$$\int_\Omega ||\nabla g - \langle \nabla g, \nabla f\rangle \frac{\nabla f}{||\nabla f||^2}||^2 ||\nabla f||\mathrm{d}\mathbf{x} \to \min_g \tag{4}$$

subject to the boundary conditions

$$\begin{aligned} g(x,y) &= 0 \text{ if } (x,y) \in \Gamma_4 \\ g(x,y) &= 1 \text{ if } (x,y) \in \Gamma_2. \end{aligned}$$

Note that we do not need to specify any boundary condition on the two curves Γ_1 and Γ_3.

4.2 Step 2: Finding a Spline Approximation S of F^{-1}

We use least squares fitting to find the mapping S that maps the unit square to $\mathbb{R}^2$. We write S in the tensor product B-spline form as

$$S(u,v) = \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} M_{i,\mathcal{U}}(u)M_{j,\mathcal{V}}(v)\mathbf{d}_{ij} = B(u,v)\cdot D,\ (u,v)\in[0,1]^2,$$

where $D = (\mathbf{d}_{ij})$ is the vector of control points and $B(u,v) = M_{i,\mathcal{U}}(u)M_{j,\mathcal{V}}(v)$ is the vector of tensor product B-splines. We denote by $\mathcal{U}$ and $\mathcal{V}$ two given knot vectors with degree-fold boundary knots 0 and 1. The index sets $\mathcal{I}$ and $\mathcal{J}$ are determined by the knot sequences and degrees of the B-splines. Then, we formulate the problem to find $S \approx F^{-1}$ as the following least squares problem

$$\int_\Omega ||S\circ F - \mathrm{id}||^2 + R_f \to \min_D \tag{5}$$

where R_f is the fairness term

$$R_f = \omega_1 \int_{[0,1]^2} (S_u^2 + S_v^2)\mathrm{d}u\mathrm{d}v + \omega_2 \int_{[0,1]^2} (S_{uu}^2 + 2S_{uv}^2 + S_{vv}^2)\mathrm{d}u\mathrm{d}v$$

and ω_1 and ω_2 are weights. The least squares problem is a quadratic minimization problem with respect to D. Therefore, it is relatively simple to find the solution by solving a linear system.

5 Injectivity

Before going to the details of the implementation, we discuss the injectivity of the mapping F. The argument for three dimensional case is similar. We will consider each coordinate function of F separately.

The first coordinate function f minimizes the harmonic energy over the whole domain Ω. The Euler-Lagrange equation associated with (1) is the Laplace equation $\Delta f = 0$. In fact, if f is a solution of (1), f is a weak harmonic function. By the maximum principle, we conclude that each level set of f is a simple curve segment in Ω.

For the second coordinate function g, the Euler-Lagrange equation corresponding to the problem (4) is

$$\frac{1}{||\nabla f||}\nabla \cdot \left(\left(\nabla g - \langle \nabla g, \nabla f\rangle \frac{\nabla f}{||\nabla f||^2}\right)||\nabla f||\right) = 0 \qquad (6)$$

So, if g is a solution of (4), then g is a weak solution of (6). Conversely, if g is a weak solution of (6), then the weak form of (6) also is satisfied on the manifold $M = f^{-1}(c) \cap \Omega$ where c is some constant. Therefore, g is a weak harmonic function on the manifold M and g satisfies the maximum principle on M.

Proposition 1 *If f is the solution of* (1) *and g is the solution of* (4)*, then the mapping $F = (f, g)^T$ from Ω to the unit square is injective.*

Proof. We assume that there are 2 points P_1 and P_2 on Ω such that $F(P_1) = F(P_2)$. So, P_1 and P_2 must lie on a level set of f. By the assumption, we have that $g(P_1) = g(P_2)$. Since each level set of f is a simple curve segment, and g is continuous on that level set and has no local maxima or minima, we conclude that P_1 and P_2 must lie on the boundary of that level set of f or more precisely on the intersection of that level set of f with boundary of Ω. However, the intersection set consists of only 2 points. Since g satisfies the boundary conditions, the values of g on those points are different. This is a contradiction. □

However, because we must solve the variational problems numerically, the discretization error may potentially destroy the injectivity. It depends on how well we can approximate the exact solutions. This will be described in the next section.

6 Implementation and Examples

In this section, we present how to solve the problems (1), (4) and (5). Then we present some experiments with our method.

6.1 Solving Variational Problems

To solve the variational problems (1) and (4) in the previous section, we employ a least squares meshless method using B-splines which is inspired by web-splines

[9]. We want to find approximate solutions $\hat{f} \approx f$ and $\hat{g} \approx g$ in the finite dimensional space spanned by B-splines. The reason to use this space is that B-spline functions are locally piecewise polynomial with rectangular supports and provide possibilities for refinement. Moreover, we can obtain a very good approximation while having only to use a few basis functions. So, once we obtain the solution $\hat{f}$, we can evaluate its values very fast. This is important because in the next variational problem for g, fast evaluation of $\hat{f}$ and its gradient speeds up the time for computing g.

For finding an approximate solution $\hat{f}$, we write

$$\hat{f} = \sum_{(i,j)\in\mathcal{I}'} N_{i,\mathcal{X}}(x) N_{j,\mathcal{Y}}(y) c_{ij}$$

where c_{ij} are real coefficients. The basis functions $N_{i,\mathcal{X}}$ and $N_{j,\mathcal{Y}}$ are B-splines of degree d_1 and d_2 with respect to knot sequences $\mathcal{X}$ and $\mathcal{Y}$ which are determined by the projections to the axes of the edges of the bounding box $\mathcal{B}$ of Ω. In our implementation, we use uniform knot vectors in x and y direction and consider only those tensor-product basis functions whose supports overlap the domain. By using the penalty method to enforce the boundary conditions, we arrive at the following least squares problem

$$\lambda \int_{\partial\Omega} (\hat{f} - f_{\mathcal{D}})^2 \mathrm{d}s + \int_{\Omega} ||\nabla \hat{f}||^2 \mathrm{d}\mathbf{x} \to \min_{c_{ij}} \tag{7}$$

where λ is a large weight, e.g, 10^3 and $f_{\mathcal{D}}$ is the function defined by Dirichlet boundary conditions of f on the boundary of Ω. The above minimization problem is quadratic with respect to c_{ij}, so it can be solved by solving a linear system. Note that we have to evaluate the second integral on the domain Ω which is unknown except for its boundary. At this point, an approximate implicitization algorithm, such as [14] or [6], comes to play its role. So, the boundary of Ω can be represented by the zero level set of an auxiliary function. The domain inside the boundary curve corresponds to the points on which the function possesses non-positive values. In other words, we can construct a characteristic function χ of the domain Ω. The second term in (7) can be rewritten as

$$\int_{\mathcal{B}} ||\nabla \hat{f}||^2 \chi(\Omega) \mathrm{d}\mathbf{x}$$

The linear system may be singular when the supports of some B-splines basis functions lie outside the domain Ω. Also, it may have a high condition number, if the supports have only a small intersections with the domain. In the web-spline method [9], this issue is dealt with by coupling some functions along the boundary. Instead of implementing the full web-spline method, we use a simpler approach. We simply set the value of χ to 1 on Ω and to a small value, e.g. 10^{-3}, outside Ω. We evaluate the integral by Gauss quadrature. The first integral term in (7) is evaluated by taking a weighted sum of finitely many points.

For finding the approximate solution $\hat{g}$, we use a similar method as above. The least squares problem is

$$\lambda \int_{\Gamma_2 \cup \Gamma_4} (\hat{g} - g_{\mathcal{D}})^2 \mathrm{d}s + \int_{\Omega} ||\nabla \hat{g} - \langle \nabla \hat{g}, \nabla \hat{f} \rangle \frac{\nabla \hat{f}}{||\nabla \hat{f}||^2}||^2 ||\nabla \hat{f}|| \mathrm{d}\mathbf{x} \to \min_{c'_{ij}} \tag{8}$$

where $g_{\mathcal{D}}$ is the function defined by Dirichlet boundary conditions of g on the two curves Γ_2 and Γ_4. We denote c'_{ij} the new variables to indicate that we can use a different discretization space but the domain is the same. Analogously, the integral over Ω can be transformed to the integral over $\mathcal{B}$ by using the characteristic function. Again, this is a quadratic minimization problem with respect to the coefficients of $\hat{g}$.

6.2 Spline Approximation of the Inverse Mapping

Here we want to describe in detail how to solve the problem (5). The first integral term in (5) can be written as

$$\iint_{\mathbb{R}^2} ||B(f(x,y), g(x,y)) \cdot D - (x,y)^T||^2 \chi_{\Omega}(x,y) \mathrm{d}x\mathrm{d}y$$

where the characteristic function $\chi_{\Omega}(x,y)$ is already known from approximate implicitization but now it is set to 0 outside Ω. We evaluate the integral by numerical quadrature, i.e, taking uniform grid of points on the bounding box $\mathcal{B}$ of Ω. In order to obtain a good approximation for the boundary of Ω, we add more terms that are integrals over the boundary of Ω as

$$\omega_0 \Bigg(\int_{\Gamma_1} ||B(0, g(\mathbf{p})) \cdot D - \mathbf{p}||^2 \mathrm{d}\mathbf{p} + \int_{\Gamma_3} ||B(1, g(\mathbf{p})) \cdot D - \mathbf{p}||^2 \mathrm{d}\mathbf{p}$$
$$+ \int_{\Gamma_2} ||B(f(\mathbf{p}), 1) \cdot D - \mathbf{p}||^2 \mathrm{d}\mathbf{p} + \int_{\Gamma_4} ||B(f(\mathbf{p}), 0) \cdot D - \mathbf{p}||^2 \mathrm{d}\mathbf{p} \Bigg)$$

where ω_0 is a large weight to indicate that we want to achieve a better approximation on the boundary. Once we have found the spline approximation, we use the L^∞ norm to estimate the error on the boundary of the domain. If the accuracy is not sufficient, then one may increase the numbers of degrees of freedom via knot insertion.

The obtained parametrization is an approximation on the inverse of the constructed coordinate functions, hence it does not reproduce the original boundary curves in general. It is possible to preserve the original boundaries, provided that the computed parameterization is compatible with the original parameterization of the boundaries. In this situation, the given control points of the boundary curves can be used as known coefficients in the fitting process.

The compatibility of the parametrizations can be achieved by composing the computed parameterization with two monotonic re-parameterizations, one for each coordinate function. Each of these monotonic re-parameterizations takes constant values 0 and 1 on two of the four boundaries, and it is determined by the original and the computed parametrizations on the other two boundaries. It can be extended to the entire domain by simple linear interpolation, which preserves the monotonicity along the parameter lines.

We did not (yet) implement this approach. We feel that the main advantage of IGA is that simulation and geometric design are based on the same geometry representation, and not necessarily in using an already existing CAD description. This will often be impossible, e.g., for domains possessing more or less than four B-spline boundary curves.

6.3 Putting Things Together

Now we summarize the implementation step by step. First, a domain defined by 4 curves is given. Next, we have to parameterize the curves Γ_2 and Γ_4 to $[0,1]$ in order to define the boundary conditions (2). This is performed by arc length parameterization scaled to $[0,1]$. Next, we find the bounding box of Ω. It is the domain for spline approximation of the functions f and g. Then, we generate the characteristic function for Ω by approximate implicitization. Finally, the two main steps, variational problems solving and spline approximation as described above, are performed.

6.4 Examples

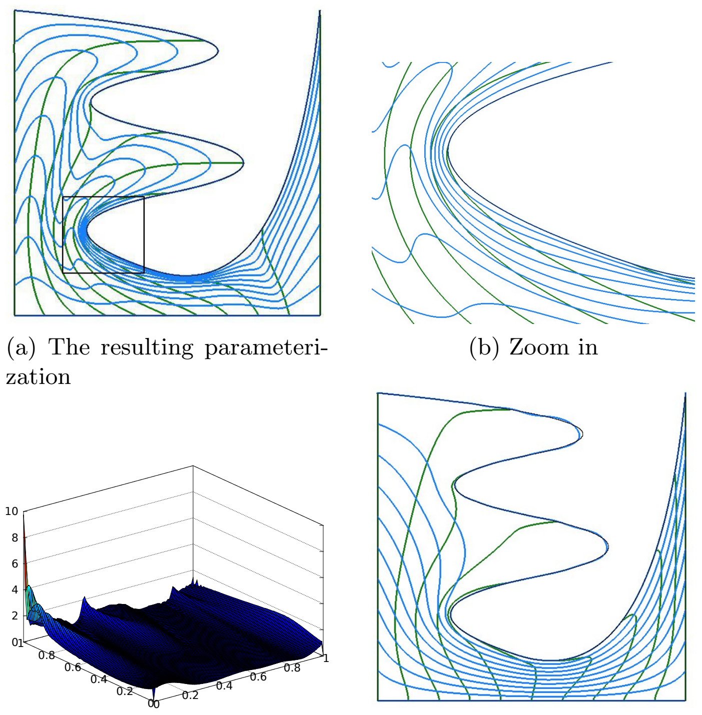
Our method is implemented in *C++* and uses *Gotools* [16] for B-splines manipulations. In addition, for solving the various arising sparse linear systems, we employ the *umfpack* routine [4]. The results in this paper are visualized using *OpenGL* and *Qt*. All examples in this paper are performed on a 2.8 GHz Intel Core 2 Duo CPU with 4 GByte Ram.

Example 1. Figure 3 demonstrates our method for a weird planar shape. Three boundary curves of that shape are line segments and the remain curve is modeled as quadratic B-splines curve with 7 control points. The picture (3a) shows the resulting spline approximation S and the level sets of f and g are approximated to the green curves and blue curves, respectively. The estimate errors in L^∞ norm are 0.002, 0.0004, 0.004 and 0.001 with respect to the boundary curve 1 to 4. We check the injectivity of F and S by evaluating its Jacobian at each sample point inside the domain. Then we plot the Jacobian of S in the picture (3c).

It is also possible to certify the injectivity of the parameterization. For example, sufficient conditions for injectivity are presented in the recent paper [18]. However, this method requires additional computational effort.

The construction of the parametrization map is not symmetric. If we swap the order for constructing f and g, we obtain the result plotted in the picture (3d). It is not a good parameterization, although F is still injective.

Example 2. The second example in Figure 4 is a shape representing the map of Austria. The four boundary curves of this shape are quadratic B-spline curves. The spline which parameterize the shape has 62×62 control points.



(a) The resulting parameterization

(b) Zoom in

(c) Jacobian of the resulting spline surface

(d) The resulting parameterization in the reverse order

Fig. 3: Parameterization for a weird planar shape.

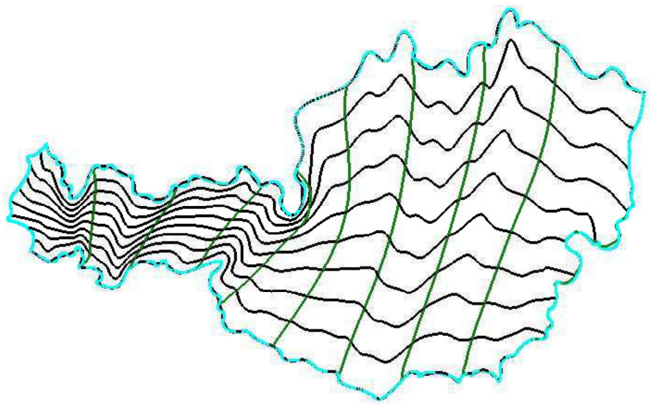
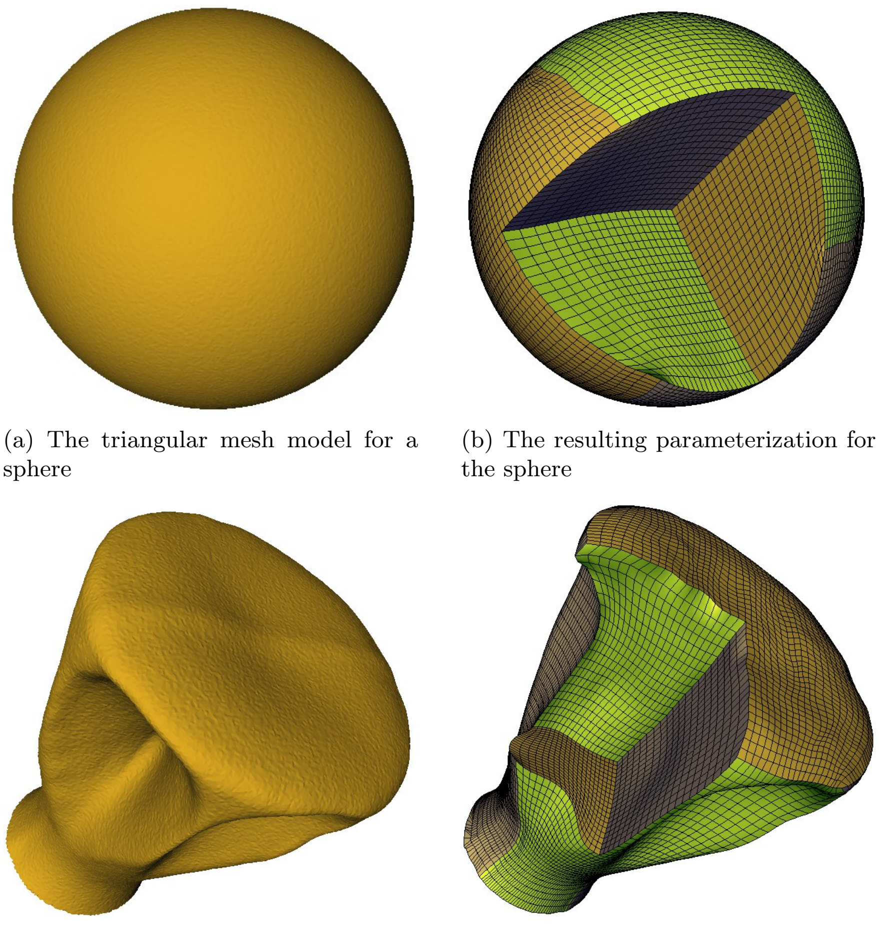


Fig. 4: The Austria map.

Example 3. We shows some results for 3D case in Figure 5. In this situation, the input data is a closed triangular meshes. To be able to perform our the method, two preprocessing steps must be done. First, the mesh is segmented into 6 faces such that each face has exactly 4 incident faces. In other words, we should have the same topology as the unit cube. Second, we parameterize 4 faces to obtain boundary data for the variational problems. In order to visualize the quality of the parametrization, the figures (5b) and (5d) show a cut-off view of the interior of the solid objects.


(a) The triangular mesh model for a sphere

(b) The resulting parameterization for the sphere

(c) The triangular mesh model for a base of a screwdriver

(d) The resulting parameterization for the base of the screwdriver

Fig. 5: Volumetric parameterization for two solids represented by closed triangular meshes.

7 Conclusion

In this paper, we proposed a framework to compute an injective mapping from a domain defined by its boundary curves (surfaces) to the unit square (unit cube). This mapping is defined via solutions of two (three in 3D) variational problems. The final spline representation for the domain was constructed as an approximation of the inverse of the computed mapping. We also demonstrated that our method works efficiently for some complicated domains.

References

1. M. Aigner, C. Heinrich, B. Jüttler, E. Pilgerstorfer, B. Simeon, and A. V. Vuong. Swept volume parameterization for isogeometric analysis. In Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII, pages 19–44, Berlin, Heidelberg, 2009. Springer-Verlag.
2. M. Bertalmío, LT. Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. J. Comput. Phys., 174(2):759–780, 2001.
3. E. Cohen, T. Martin, R.M. Martin, T.Lyche, and R.F. Riesenfeld. Analysis-aware modeling: Understanding quality considerations in modeling for isogeometric analysis. Comput.Methods Appl.Mech.Engrg, 2009.
4. T. Davis. http://www.cise.ufl.edu/research/sparse/umfpack/.
5. H.Q Dinh, A. Yezzi, and G. Turk. Texture transfer during shape transformation. ACM Trans. Graph., 24(2):289–310, 2005.
6. T. Dokken and J. B. Thomassen. Weak approximate implicitization. Shape Modeling and Applications, International Conference on Shape Modeling and Applications 2006 (SMI'06), 0:31, 2006.
7. G. Farin and D. Hansford. Discrete Coons patches. Comput. Aided Geom. Des., 16(7):691–700, 1999.
8. X. Gu and S.T. Yau. Global conformal surface parameterization. In Proceedings of the First Eurographics Symposium on Geometry Processing, pages 127–137. Eurographics Association, 2003.
9. K. Höllig. Finite element methods with B-splines. Frontiers in Applied Mathematics. 26. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.
10. K. Hormann, K. Polthier, and A. Sheffer. Mesh parameterization: Theory and practice. In SIGGRAPH Asia 2008 Course Notes, number 11, Singapore, December 2008. ACM Press.
11. J. Hoschek and D. Lasser. Fundamentals of computer aided geometric design. Wellesley, 1993.
12. T.J.R Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. Computer Methods in Applied Mechanics and Engineering, 194(39-41):4135–4195, 2005.
13. J. Jost. Riemannian Geometry and Geometric Analysis. Springer, 2005.
14. B. Jüttler and A. Felis. Least-squares fitting of algebraic surfaces. Advances in Computational Mathematics, 17:135-152, 2002.
15. 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.

16. SINTEF. http://www.sintef.no/Projectweb/Geometry-Toolkits/GoTools/.
17. H. K. Voruganti, V. Gupta, and B. Dasgupta. Domain mapping for spherical parametrization using harmonic function theory. In IISc Centenary-International Conference on Advances in Mechanical Engineering (IC-ICAME), 2008.
18. G. Xu, B. Mourrain, R. Duvigneau, and A. Galligo. Parametrization of computational domain in isogeometric analysis: methods and comparison. Computer Methods in Applied Mechanics and Engineering, In Press, Accepted Manuscript, 2011.