



**HAL**  
open science

# Hermite type Spline spaces over rectangular meshes with complex topological structures

Meng Wu, Bernard Mourrain, André Galligo, Boniface Nkonga

► **To cite this version:**

Meng Wu, Bernard Mourrain, André Galligo, Boniface Nkonga. Hermite type Spline spaces over rectangular meshes with complex topological structures. *Communications in Computational Physics*, 2017, 21 (3), pp.835-866. 10.4208/cicp.OA-2016-0030 . hal-01196435v2

**HAL Id: hal-01196435**

**<https://inria.hal.science/hal-01196435v2>**

Submitted on 30 Nov 2015 (v2), last revised 26 Apr 2017 (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.

# $H^1$ -parameterizations of plane physical domains with complex topology in Isogeometric analysis

Meng Wu\* Bernard Mourrain André Galligo  
Boniface Nkonga

School of Mathematics, Hefei University of Technology, P. R. China.  
Galaad, Inria, Sophia Antipolis, France  
Lab. J. A. Dieudonné, University of Nice, Nice, France

## Abstract

Isogeometric analysis (IGA) is a method for solving geometric partial differential equations(PDEs). Generating parameterizations of a PDE's physical domain is the basic and important issues within IGA framework. In this paper, we present a global  $H^1$ -parameterization method for a planar physical domain with complex topology.

## 1 Introduction

IGA method was proposed by Hughes et al. [1] to bridge the gap between the geometry description of the physical domain and numerical analysis. Within IGA framework, the same basis functions are used to describe physical domains and IGA solutions obtained by numerical analysis.

A map to describe a physical domain is called a parameterization of this physical domain in IGA framework. The parameterization of the physical domain has an impact on the simulation result, the efficiency of the computation and the stability of numerical system[2],[3],[4]. Until now, there is no simple standard for measuring the quality of parameterization. In this paper, we develop a method for modifying the initial parameterization focusing on three basic requirements.

Firstly, for describing the geometry of physical domains, parameterizations should have no self-intersection such as [5],[6], [7]. In other words, it should be an injective map.

---

\*M. Wu (Corresponding author, email: meng.wu@hfut.edu.cn), B. Mourrain (email: Bernard.Mourrain@inria.fr), A. Galligo(email: Andre.Galligo@unice.fr), B. Nkonga (email: Boniface.Nkonga@unice.fr)

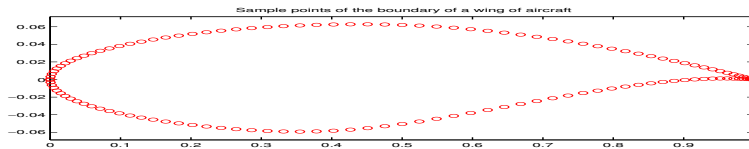
Secondly, for numerical analysis, the condition number of the linear system obtained within IGA framework should be as small as possible [8, 9]. Smaller condition numbers bring the efficiency and numerical stability. In applications, this linear system is always a large size. It is better to use iteration solvers in order to reduce the computation costs. At that moment, smaller condition number means rapid convergence and more accurate linear system solution. However, the condition number of the linear system in IGA depends on the underlying parameterization. Thus, we expect to obtain a parameterization which will bring a small condition number of the linear system within IGA framework.

Thirdly, the regularity property of test functions on physical domains is considered. The test functions used in Galerkin-based IGA are obtained by composing the inverse of the parameterization with basis functions used in IGA. Thus, parameterizations impact on the regularity property of the test functions on the physical domains. The singularity of parameterizations should satisfy some additional conditions which guarantee the regularity properties of the test functions on the physical domains, see [19, 20].

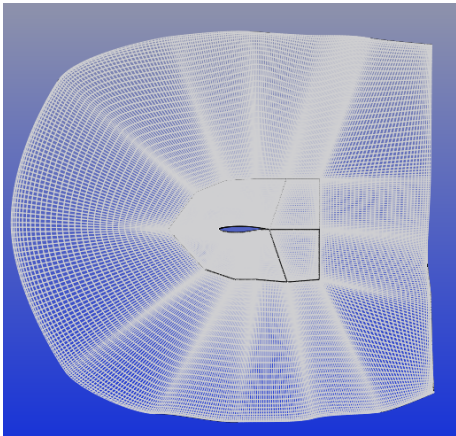
In this paper, based on these three requirements, we develop an algorithm for modifying the quality of initial parameterizations with a fix boundary representation. The resulting parameterization, called  $H^1$ -parameterization, satisfies  $H^1$  integrability assumption in [19] for solving second order PDEs, such as the Grad-Shafranov equation which is the equilibrium in the resistive magneto hydrodynamic model. This algorithm can be applied to several situations. For example, in order to simulate the two-dimensionality of the flow over the RAE2822 airfoil, the physical domain is a bounded domain around it. The initial parameterization (Figure 1(b)) is produced by fitting the sample points shown in Figure 1(a) of its boundary. Thus, during modification of the initial parameterization, fixed boundaries are expected. In this paper, a modification algorithm is developed in Sections 3.1 and 3.2. Applying this algorithm, the resulting  $H^1$ -parameterization is shown in Figure 1(c).

Moreover, usually, Non-Uniform Rational B-Splines (NURBS) are treated as basis functions of the classical IGA, i.e. NURBS-based IGA. Then when we consider a planar physical domain with a complex topology, patching multiple NURBS parameterizations or trimming techniques, such as [10], need to be considered. In this paper, to obtain a global representation of a physical domain with complex geometry features and complex topology features, the bicubic splines with rigid transformations as in [11] are used to represent the global parameterizations.

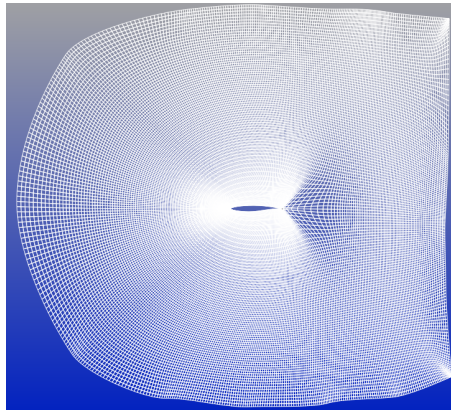
This paper is structured as follows. In Section 2, the global parameterization is introduced and some properties of parameterizations are presented. Then, in Section 3, we develop the modification algorithm for initial parameterizations taking into account these properties. Using this algorithm, in Section 4, several experiments are presented to modify the initial parameterizations of physical domains. These physical domains are with complex geometric features (such as sharp corners and



(a) The sample points from the RAE2822 airfoil



(b) The initial parameterization by fitting the sample points



(c) The resulting  $H^1$ -parameterization

Figure 1:

non-convex shapes) or complex topology features (such as many holes inside). In Section 5, we conclude this paper with future work directions. At last, all the proofs of lemmas or theorems are collected in Section 6 as the appendix of this paper.

## 2 Parameterization with splines defined over meshes with complex topologies

In this paper,  $C^1$  splines with complex topology are used to represent parameterizations. In this section, we give an intuitive explanation of this type of splines. One can refer to [11] for their precise mathematical definitions. Based on this intuitive explanation, the parameterization represented by these splines is decomposed into two parts. The different modification algorithms for these two parts will be presented in Section 3 later.

### 2.1 An intuitive introduction to splines

Before introducing the splines, parametric meshes are presented since the splines are defined over this type of mesh.

The concept of parametric meshes in this paper is a generalization of the classical tensor product mesh. Figure 2 shows a 2D tensor product mesh which contains anisotropic rectangles with arbitrary aspect ratio. These rectangles are known as the cells of this 2D tensor product mesh and these grid points are called its vertices. In our generalization, we relax the restrictions on the degree of each vertex in order to obtain a global parameterization without trimming of surfaces with more complex topology such as surfaces with holes inside or sphere.

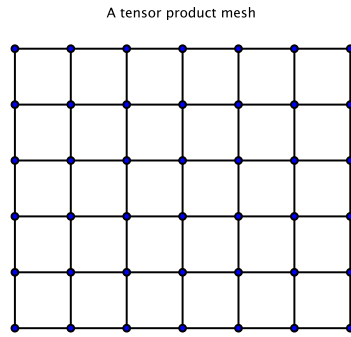


Figure 2: A 2D tensor product mesh

For example, in Figure 3, there are some undirected graphs  $\mathcal{M}_t$  which share the same topology with corresponding parametric meshes  $\mathcal{M}$ . From topological point of view, we make no difference between  $\mathcal{M}$  and  $\mathcal{M}_t$  (denoted as  $\mathcal{M}$  as well).

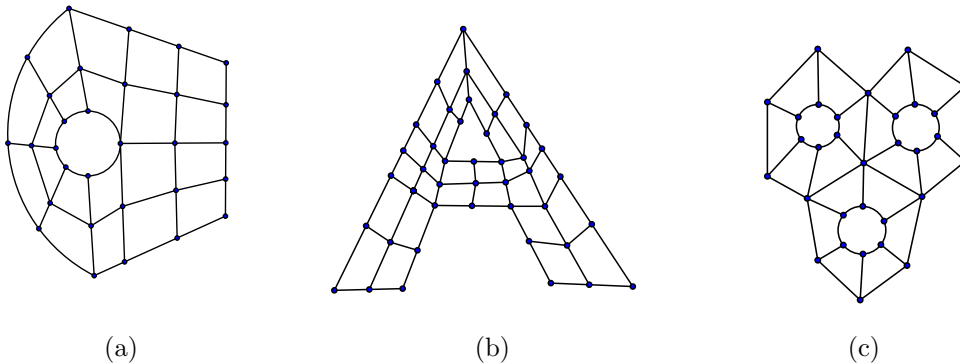


Figure 3: Examples of the topology of parametric meshes ( $\mathcal{M}$ s)

In the following, we will rely on illustrations to introduce some concepts. In Figure 4(a),  $\{v^i : i = 1, 2, \dots, 10\}$  and  $\{v_b^j : j = 1, 2, \dots, 20\}$  are vertices of  $\mathcal{M}$ .  $v_b^j$  ( $j = 1, 2, \dots, 20$ ) are **boundary vertices** and  $v^i$  ( $i = 1, 2, \dots, 10$ ) are **interior vertices**. The degree of a vertex of the undirected graph  $\mathcal{M}$  is also called the valence of this vertex. If the degree of an interior vertex is not 4, this vertex is **irregular**.

Irregular vertices are also called extraordinary vertices. Other vertices are **regular**. An edge of  $\mathcal{M}$  connects adjacent vertices. In Figure 4(a),  $v_b^{15}v_b^{20}$ ,  $v^1v^2$ ,  $v^1v_b^1$  are edges of  $\mathcal{M}$ . Here,  $v_b^{15}v_b^{20}$  is a **boundary edge** of  $\mathcal{M}$ , while  $v^1v^2$ ,  $v^1v_b^1$  are interior edges. Quadrilaterals with curved edges, such as  $v_b^{15}v_b^{20}v^{10}v^1$ ,  $v^3v^4v_b^{18}v_b^{17}$  and  $v_b^5v_b^6v^6v^5$ , are treated as cells of  $\mathcal{M}$ .

**Local frame of a cell  $C$ :** For each cell of the parametric mesh, a local frame is introduced. In Figure 4(a),  $\{\mathbf{s}_C, \mathbf{t}_C\}$  is a local frame of the cell  $C$ .

**Local parameters of a cell  $C$ :** A global spline on  $\mathcal{M}$  can be described with local parameters. For example, based on its local frames, in Figure 4(a), a spline with  $C^1$  continuity globally is a bicubic polynomial over a cell  $C$  in the local parameters  $(s_C, t_C)$ .

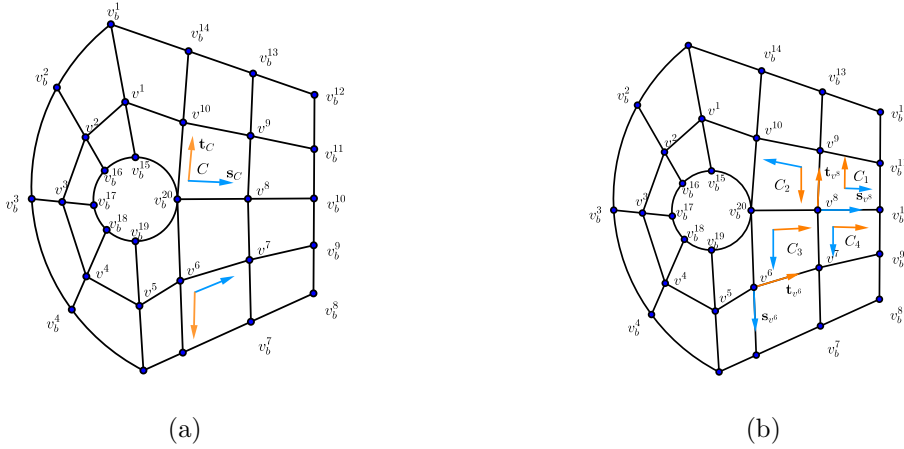


Figure 4: An example of  $\mathcal{M}$  for introducing the concepts of parametric meshes.

Associated with each vertex of  $\mathcal{M}$ , splines will be introduced to represent parameterizations in the following context. Based on the different types of vertices, splines will be classified into two classes.

1. For an extraordinary vertex  $v_{ir}$ , there is a spline  $B_{v_{ir}}^p$  defined at  $v_{ir}$ . Its coefficient  $\mathbf{P}_{v_{ir}} \in \mathbb{R}^2$  stands for the position of  $v_{ir}$ . This spline is called the **position spline** of  $v_{ir}$ .
2. For a regular vertex  $v_r$ , three splines  $B_{v_r}^p$ ,  $B_{v_r}^s$  and  $B_{v_r}^t$  will be defined over  $\mathcal{M}$ . For  $B_{v_r}^p$ , its coefficient  $\mathbf{P}_{v_r} \in \mathbb{R}^3$  stands for the position of  $v_r$ .  $B_{v_r}^p$  is called the position spline of  $v_r$ .  $B_{v_r}^s$  and  $B_{v_r}^t$  are called **tangent splines**.

In order to explain the coefficients of  $B_{v_r}^s$  and  $B_{v_r}^t$ , the local frame attached to  $v_r$  is introduced. This local frame  $(\mathbf{s}_{v_r}, \mathbf{t}_{v_r})$  is the same as one of the local frames of the cells around  $v_r$ . For example, in Figure 4(b), at  $v^8$ , there is a local frame  $(\mathbf{s}_{v^8}, \mathbf{t}_{v^8})$  which is the same as the local frame of  $C_1$ .  $\mathbf{T}_{v_r}^s$  (or  $\mathbf{T}_{v_r}^t$ ) is the tangent vector of the  $s$ -curve (or  $t$ -curve) at  $v_r$ , where the  $s$ -curve (or  $t$ -curve) means the image of mesh grid line of parameterization along  $\mathbf{s}_{v_r}$  (or  $\mathbf{t}_{v_r}$ ) direction. For example, in

Figure 5(b), the domain is an image of one of the parameterizations defined over  $\mathcal{M}$  shown in Figure 5(a). By this parameterization, the  $s$ -curve consists of the image of the edges  $v_b^{20}v^8$ ,  $v^8v_b^{10}$  of the direction which is matching with the local frame  $\mathbf{s}_{v^8}$  attached to  $v^8$ . Similarly, the  $t$ -curve can be defined.

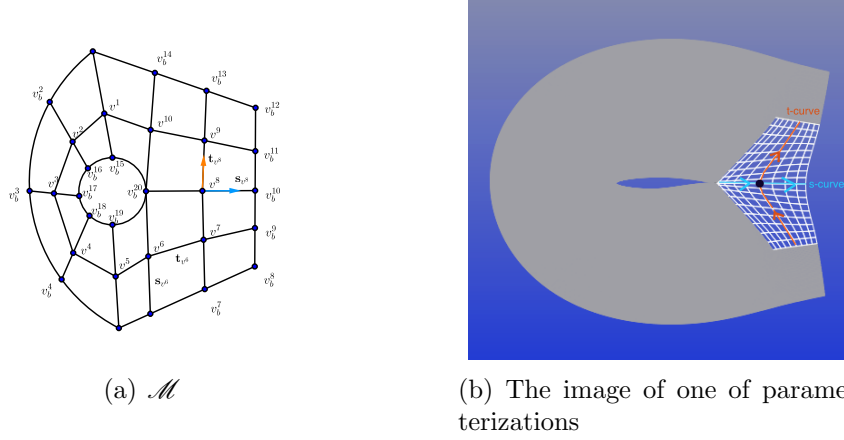


Figure 5: The  $s$ -curve and the  $t$ -curve at  $v^8$

Based on these splines, the parameterization is constructed as a linear combination of splines,

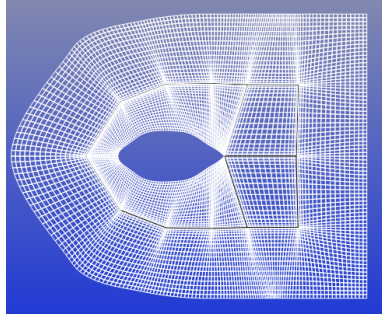
$$\mathcal{P} = \sum_{v \in V} \mathbf{P}_v B_v^p + \sum_{v_r \in V_r} (\mathbf{T}_{v_r}^s B_{v_r}^s + \mathbf{T}_{v_r}^t B_{v_r}^t) \quad (1)$$

where  $V$  and  $V_r$  are the set of vertices and the set of regular vertices of  $\mathcal{M}$ . In this paper, parameterizations of a planar physical domain are considered, i.e.  $\mathbf{P}_v, \mathbf{T}_{v_r}^s, \mathbf{T}_{v_r}^t \in \mathbb{R}^2$ . For example, Figure 6(a) is the original parameterization. In Figure 6(b), by changing  $\mathbf{P}_v$  of the given vertex, the position of this vertex is changed. In Figure 6(c) and Figure 6(d), by setting  $\mathbf{T}_{v_r}^s$  and  $\mathbf{T}_{v_r}^t$ , the parameterization has been changed around the given vertex.

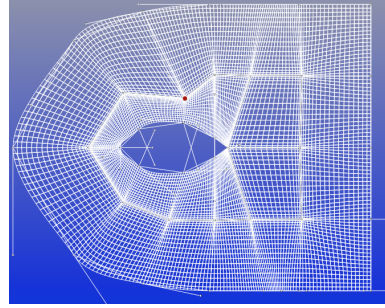
**Remark 2.1.** In [11], there is another spline  $B_{v_r}^{st}$  at the regular vertex  $v^r$ . Here, its coefficient is set as zero. In this paper, based on the geometrical meaning of the coefficients of  $B_v^p$ ,  $B_{v_r}^s$  and  $B_{v_r}^t$ , we develop an algorithm for generating  $H^1$ -parameterizations.

**Jacobian of  $\mathcal{P}$ :** Let  $p$  be a point of  $\mathcal{M}$ , then there is a cell  $C$  of  $\mathcal{M}$  such that  $p \in C$ . Consider  $\mathcal{P}|_C$ , by Equation (1),  $\mathcal{P}|_C = (x(s_C, t_C), y(s_C, t_C))$ , where  $x(s_C, t_C), y(s_C, t_C)$  are bicubic polynomials. Thus the Jacobian of  $\mathcal{P}$  at  $p$  is

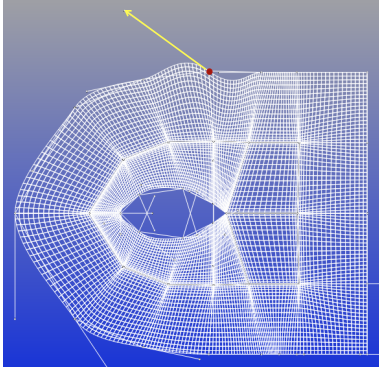
$$\begin{vmatrix} \frac{\partial x(s_C, t_C)}{\partial s_C} & \frac{\partial x(s_C, t_C)}{\partial t_C} \\ \frac{\partial y(s_C, t_C)}{\partial s_C} & \frac{\partial y(s_C, t_C)}{\partial t_C} \end{vmatrix}_p.$$



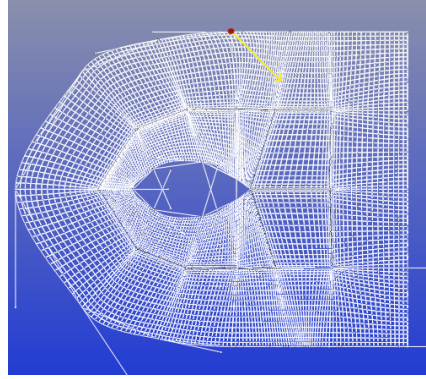
(a) The original parameterization



(b) Changing  $\mathbf{P}_v$  of the original parameterization in Figure 6(a) at the given vertex



(c) Changing  $\mathbf{T}_v^s$  or  $\mathbf{T}_v^t$  of the original parameterization in Figure 6(a) at the given vertex



(d) Changing  $\mathbf{T}_v^s$  or  $\mathbf{T}_v^t$  of the original parameterization in Figure 6(a) at the given vertex

Figure 6: The meaning of coefficients of splines defined over the parametric mesh in Figure 4(a)

## 2.2 Parameterizations of planar physical domains

In this section, we will analyze the parameterization represented by the splines defined in Section 2.1. Based on the meaning of coefficients of each spline, the parameterization  $\mathcal{P}$  can be decomposed into two parts  $\mathcal{P}_p$  and  $\mathcal{P}_t$ . i.e.,

$$\mathcal{P} = \mathcal{P}_p + \mathcal{P}_t,$$

where  $\mathcal{P}_p = \sum_{v \in V} \mathbf{P}_v B_v^p$ ,  $\mathcal{P}_t = \sum_{v_r \in V_r} (\mathbf{T}_{v_r}^s B_{v_r}^s + \mathbf{T}_{v_r}^t B_{v_r}^t)$ .  $\mathbf{P}_v$  is the position coordinates of the vertex  $v$ .  $\mathbf{T}_{v_r}^s$  and  $\mathbf{T}_{v_r}^t$  are the tangent vectors of the  $s$ -curve and  $t$ -curve at  $v_r$ .

$\mathcal{P}_p$ : For this part of the parameterization, the coefficients of the position splines are the positions of vertices in the physical domain. We are going to modify  $\mathcal{P}_p$ , i.e., distribute the layout of an undirected graph [13, 14]. This modification algorithm will be dealt with in Section 3.1. Take the parameterization presented in Figure 6(a)



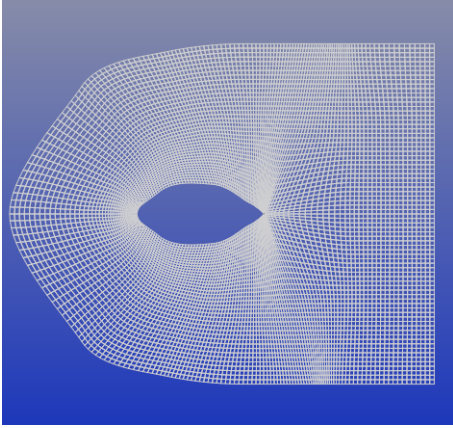
for example. This parameterization is only represented by the position splines at each vertex, i.e. the coefficients of  $\mathbf{T}_{v_r}^s, \mathbf{T}_{v_r}^t$  are zeros.

The following lemma will be proved in Section 6.

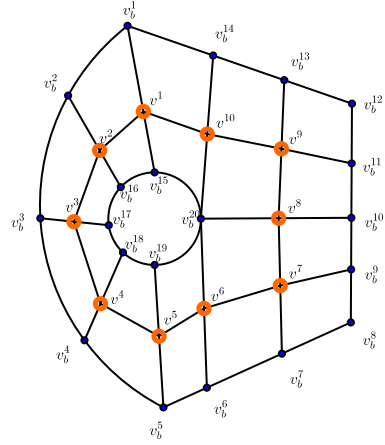
**Lemma 2.1.** *The Jacobian of  $\mathcal{P}_p$  is zero along  $s$ -curves and  $t$ -curves at any vertex of  $\mathcal{M}$ .*

Therefore, it is necessary to add a non-trivial  $\mathcal{P}_t$  in order to get a parameterization with a better Jacobian.

$\mathcal{P}_t$ : Adding a non-trivial  $\mathcal{P}_t$ , the Jacobian of  $\mathcal{P}$  will be modified from the original one. For example, Figure 7(a) and Figure 7(b) illustrates this process: by adding a non-trivial  $\mathcal{P}_t$  at the interior vertices of  $\mathcal{M}$ , we notice that the parameterization becomes smoother than the original parameterization in Figure 6(a). We will develop an algorithm in Section 3.2 called the smoothing algorithm for choosing a suitable  $\mathcal{P}_t$  at the regular interior vertices of the parametric mesh.



(a) The parameterization after adding a non-trivial  $\mathcal{P}_t$  at the regular interior vertices of the parameterization shown in Figure 6(a)



(b) The parametric mesh and its regular interior vertices marked in red

Figure 7: Change of coefficients of tangent splines at the regular interior vertices

### 3 The parameterization Algorithm

In this section, we suppose that an initial parameterization  $\mathcal{P}_0$  is given. We will develop an algorithm to modify  $\mathcal{P}_0$  with fixed boundaries such that the resulting parameterization satisfies the three requirements stated in Introduction, i.e. an injective map, small condition numbers and  $H^1$ -parameterization. Based on the

analyzing in Section 2.2,  $\mathcal{P}_0$  can be decomposed into two parts. We modify  $\mathcal{P}_0$  using two steps matching with these two parts. The first step amounts to distribute the layout of an undirected graph with fixed boundaries, in other words, determining the position splines of the coefficients at interior vertices. A force-directed layout algorithm is presented in Section 3.1; The second step is the smoothing algorithm used to modifying the tangent splines' coefficients at each interior regular vertex in Section 3.2.

### 3.1 The force-directed layout algorithm

Force-directed algorithms are among the most flexible methods for calculating layouts of simple undirected graphs. These force-directed algorithms tend to be aesthetically pleasant, exhibit symmetries and tend to produce crossing-free layouts for planar graphs. For example, the first paper about layouts for planar graphs, [16], is a force-directed method. In this section, we will deal with layout of the undirected graph with fixed boundaries based on the spring model and the point-charge model. The undirected graph is given by the image of the vertices and edges of the parameter mesh. Then, the position splines' coefficients of interior vertices are re-determined.

To re-display the vertices, the following energy is presented. Let  $P_i(t)$  be the position according to  $v_i \in V^o$  ( $i = 1, 2, \dots, n$ ) and  $P_i(t)$  evolves with time.

$$E(P_1(t), P_2(t), \dots, P_n(t)) = \sum_{v_i \in V^o} \|k\mathbf{F}_i^{sp}(t) + \mu\mathbf{F}_i^{ch}(t)\|_{L_2}^2, \quad (2)$$

where  $k$  and  $\mu$  are parameters of this model,  $\mathbf{F}_i^{sp}(t) = \sum_{P_j(t) \in N^1(P_i(t))} \overrightarrow{P_i(t)P_j(t)}$  and  $\mathbf{F}_i^{ch}(t) = \int_{P \in \partial\Omega} \overrightarrow{P_i(t)P} / (\|\overrightarrow{P_i(t)P}\|_{L_2})^2 ds$ .  $V^o$  is the set of interior vertices of  $\mathcal{M}$ .  $\Omega$  is the physical domain.  $\mathbf{F}_i^{sp}$  corresponds to the spring force at  $P_i(t)$ , while  $\mathbf{F}_i^{ch}(t)$  describes the electric force from  $\partial\Omega$  to  $P_i(t)$ . We suppose that  $\partial\Omega$  and  $P_i(t)$  carry the same type of charges. By the definition of  $\mathbf{F}_i^{ch}(t)$ , when  $P_i(t)$  is close to the boundary of  $\Omega$ ,  $\mathbf{F}_i^{ch}(t)$  becomes large. Thus,  $P_i(t)$  is confined within  $\Omega$  if the initial position of  $P_i(t)$  is within  $\Omega$  and  $\mu > 0$ .

When all the positions  $P_i(t)$  are in balance, the energy in (2) reaches its minimal value. Thus, at  $t$ , the following lemma proved in Section 6 is used to predict the new position of  $P_i(t)$  at  $t_0 + \Delta t$ .

**Lemma 3.1.** *Suppose that the velocity of  $P_i(t)$  at  $t_0$  is  $\mathbf{0}$ . Then*

$$P_i(t_0 + \Delta t) - P_i(t_0) = \frac{\mathbf{F}_i(t_0)\Delta t^2}{2m_i} + o(\Delta t^2).$$

where  $\mathbf{F}_i(t_0) = \mathbf{F}_i^{sp}(t_0) + \mathbf{F}_i^{ch}(t_0)$ .

Thus,

$$P_i(t_0 + \Delta t) - P_i(t_0) = \Delta P_i(t_0) \approx \frac{\mathbf{F}_i(t_0)\Delta t^2}{2m_i}. \quad (3)$$

Thus we can choose a suitable time step  $\Delta t$  and parameters  $m_i, k, \mu$ . The new position of  $P_i(t_0 + \Delta t)$  after one iteration can be approximated by  $P_i(t_0) + \Delta P_i(t_0)$ .

Now the algorithm goes as follows. Set the parameters  $\epsilon_0, \Delta t, k, \mu$  and  $MaxIterNumber$ , where  $\epsilon_0$  is the parameter employed to judge if this algorithm converges or not,  $\Delta t$  is the time step,  $k$  and  $\mu$  are the parameters shown in Equation (2),  $MaxIterNumber$  is the maximum number of iteration of this algorithm.

---

**Algorithm 1** Local Reducing Energy Algorithm

---

**Require:** Input the original positions of all the points  $P_i$ , the parametric mesh  $\mathcal{M}$ ,  $\epsilon_0, \Delta t, k$  and  $\mu$ . Set  $m_i = 1, \epsilon = \epsilon_0 + 1$  and  $IterationNum = 0$ .

**Ensure:**

```

while ( $\epsilon > \epsilon_0$ ) && ( $IterationNum < MaxIterNumber$ ) do
   $\epsilon \leftarrow 0$ 
   $IterationNum \leftarrow IterationNum + 1$ 
  for all  $P_i$  which is the image of an interior vertex of  $\mathcal{M}$  do
    Compute  $\Delta P_i$  based on Equation (3)
     $\epsilon \leftarrow \epsilon + \|\Delta P_i\|_{L_2}$ 
     $P_i \leftarrow P_i + \Delta P_i$ 
  end for
   $\epsilon \leftarrow \epsilon / \#\{P_i\}$ 
end while

```

In this algorithm,  $\epsilon$  is a parameter used to store the average of the relative distance of all of interior  $P_i$  for each iteration. If  $\epsilon$  is small enough,  $\{P_i\}$  should be in a balance state, i.e., if  $\epsilon < \epsilon_0$ , this algorithm converges.  $\#\{P_i\}$  is the number of the interior vertices of  $\mathcal{M}$ .

---

## 3.2 The Smoothing Algorithm

Thanks to Algorithm 1, the position splines' coefficients at the interior vertices have been obtained. In this section, a method for determining the coefficients of tangents splines at the interior regular vertices is presented. To simplify the process, we assume that all the extraordinary vertices are surrounded by regular ones. This can be achieved by subdivision. The following theorem provides an important observation.

**Theorem 3.2.** *Let  $\mathcal{M}$  be a parametric mesh and  $C$  is one of its cells. Suppose that  $C = [0, 1] \times [0, 1]$  and its local parameters are  $(s, t)$ . Denote its corners  $(0, 0), (1, 0), (1, 1), (0, 1)$  as  $v_1, v_2, v_3, v_4$  respectively. If the images of  $v_1, v_2, v_3, v_4$  of a parameterization  $P_1, P_2, P_3, P_4$  form a convex quadrilateral. Then,*

- if  $v_1, v_2, v_3, v_4$  are regular vertices of  $\mathcal{M}$ , then  $\mathbf{T}_{v_1}^s = \mathbf{T}_{v_2}^s = \overrightarrow{P_1 P_2}$ ,  $\mathbf{T}_{v_1}^t = \mathbf{T}_{v_4}^t = \overrightarrow{P_1 P_4}$ ,  $\mathbf{T}_{v_3}^t = \mathbf{T}_{v_2}^t = \overrightarrow{P_2 P_3}$ ,  $\mathbf{T}_{v_3}^s = \mathbf{T}_{v_4}^s = \overrightarrow{P_4 P_3}$ ;
- if a vertex  $v_1$  is an extraordinary vertex of  $\mathcal{M}$ , then  $\mathbf{T}_{v_2}^s = \overrightarrow{P_1 P_2}$ ,  $\mathbf{T}_{v_4}^t = \overrightarrow{P_1 P_4}$ ,  $\mathbf{T}_{v_3}^t = \mathbf{T}_{v_2}^t = \overrightarrow{P_2 P_3}$ ,  $\mathbf{T}_{v_3}^s = \mathbf{T}_{v_4}^s = \overrightarrow{P_4 P_3}$

therefore  $\mathcal{P}$  is injective over  $C$  and it satisfies  $H^1$  integral assumption.

This theorem will be proved in Appendix. In the following, we start to set the coefficients of the tangent splines in the general case. Let  $v_i$  be a regular interior vertex of  $\mathcal{M}$  and assume that its position has been determined by Algorithm 1. There are four cells which take  $v_i$  as their corner vertex since  $v_i$  is regular. Based on Theorem 3.2 over a cell, the tangents at this regular vertex  $v_i$  are estimated by  $P_i, P_i^1, P_i^2, P_i^3, P_i^4$ , where  $P_i = \mathcal{P}(v_i)$  and  $N^1(P_i) = \{P_i^1, P_i^2, P_i^3, P_i^4\}$  shown in Figure 8, where  $C_1, C_2, C_3, C_4$  are the cells around  $v_i$ .

If  $C_1, C_2, C_3, C_4$  have the same local frame and  $C_i = [0, 1] \times [0, 1]$ , the tangent vectors at  $v_i$  can be taken as an average of the coefficients given by Theorem 3.2 when we consider these coefficients cell by cell, i.e.,

$$\mathbf{T}_{v_i}^s = 1/2 \overrightarrow{P_i P_i^2} + 1/2 \overrightarrow{P_i^4 P_i}; \quad \mathbf{T}_{v_i}^t = 1/2 \overrightarrow{P_i P_i^1} + 1/2 \overrightarrow{P_i^3 P_i}. \quad (4)$$

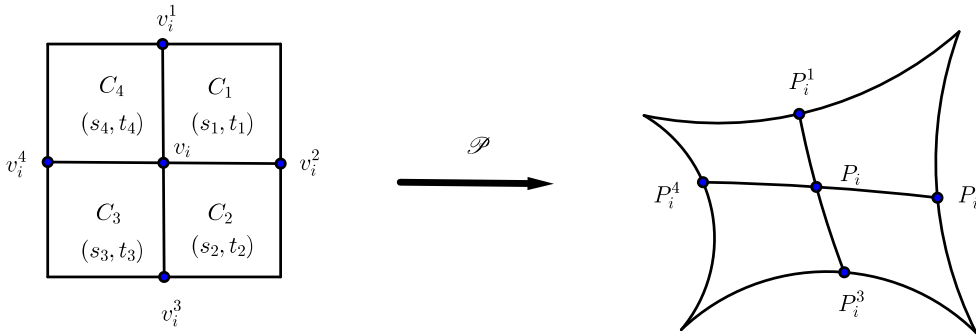


Figure 8: The interior regular vertex  $P_i$  and  $N^1(P_i) = \{P_i^1, P_i^2, P_i^3, P_i^4\}$

In general,  $C_1, C_2, C_3, C_4$  can have different local frames, if the direction of the  $s$ -curve of  $v_i$  is along  $P_i P_i^2$  and the direction of the  $t$ -curve at  $v$  is along  $P_i P_i^1$ , then

$$\mathbf{T}_{v_i}^s = \frac{1}{2\ell_1} \overrightarrow{P_i^4 P_i} + \frac{1}{2\ell_2} \overrightarrow{P_i P_i^2}; \quad \mathbf{T}_{v_i}^t = \frac{1}{2\ell_3} \overrightarrow{P_i P_i^1} + \frac{1}{2\ell_4} \overrightarrow{P_i P_i^3},$$

where  $\ell_1$  is the distance between  $v_i^4$  and  $v_i$  considered in  $C_4$ ,  $\ell_2$  is the distance between  $v_i$  and  $v_i^2$  considered in  $C_1$ ,  $\ell_3$  is the distance between  $v_i^3$  and  $v_i$  considered in  $C_2$ ,  $\ell_4$  is the distance between  $v_i$  and  $v_i^1$  considered in  $C_1$ . Select similarly  $\mathbf{T}_{v_i}^s, \mathbf{T}_{v_i}^t$  in the other cases of the local frame at  $v_i$ . Now the smoothing algorithm can be presented.

---

**Algorithm 2** The Smoothing Algorithm

---

**Require:**

Input  $\mathcal{M}$ ,  $\mathcal{P}_1$  generated from the initial parameterization  $\mathcal{P}_0$  by Algorithm 1 and  $\{P_i\}$  which are the resulting positions by Algorithm 1.

**Ensure:**

**for**  $\{P_i\}$  **do**

**if**  $v_i$  is an interior regular vertex **then**

    Compute  $P_i$ 's 1-neighbourhood  $N^1(P_i)$  based on the topology of  $\mathcal{M}$

    Determine  $\mathbf{T}_{v_i}^s$  and  $\mathbf{T}_{v_i}^t$  by the method presented in this section, where  $v_i$  is the vertex corresponding to  $P_i$ .

**else**

$\mathbf{T}_{v_i}^s, \mathbf{T}_{v_i}^t$  inherit from  $\mathcal{P}_0$ .

**end if**

**end for**

---

Besides fitting a physical domain's boundary application presented in Introduction, another application of the algorithm developed in this paper appears in solving PDEs within IGA framework. After h-refinement, updating the initial parameterization is expected to obtain a better approximation of the physical domain and a small condition number of the linear system within IGA framework. Because more degree of freedoms are generated after h-refinement of the parametric mesh. A better boundary approximation yields a better approximation of the physical domain. A smaller condition number of the linear system, which brings rapid convergence of iteration solvers and more accurately linear system solution, is determined by the quality of the parameterization within the interior of the physical domain. Logically, after updating the boundary approximation, the quality of parameterization within the inner of the physical domain is considered. Thus, it is necessary to fix the boundary representation when improving the quality of the parameterization. In the next section, we will analyse this  $H^1$ -parameterization algorithm with more general physical domains.

## 4 Algorithm Analysis, Numerical Experiments

In this section, experiments are presented. The first one is a four leaf clover-shape physical domain shown in Figure 9(a). This physical domain is of complex geometric

features with sharp corners and non-convex shape. The second one is a perforated model structure illustrated in Figure 9(b) with complex topology features.



(a) A four leaf clover-like physical domain



(b) A perforated model structure physical domain

Figure 9: The physical domains considered in Section 4

For each experiment, in order to test the qualities of the resulting  $H^1$ - parameterization within IGA Framework, we will study the Jacobian of a modified parameterization, the relationship between the condition number of the numerical linear system in IGA and the iteration times of Algorithm 1. The considered second order PDE is the following elliptic boundary problem.

$$\begin{aligned} -\Delta u &= f \\ u|_{\partial\Omega} &= 0 \end{aligned}$$

By IGA, in the weak form, the isogeometric solution is given by the solution of a numerical linear system  $Ax = b$ , where

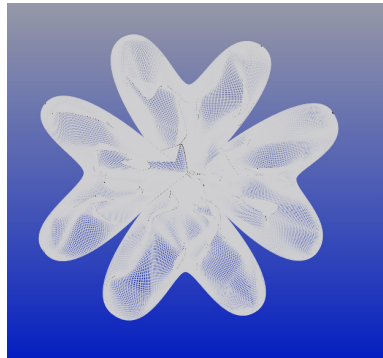
$$A = [a_{i,j}]_{n \times n}, \tag{5}$$

here  $A$  is called the stiffness matrix and  $a_{i,j} = \int \int_{\Omega} \nabla(B_i \circ \mathcal{P}^{-1}) \nabla(B_j \circ \mathcal{P}^{-1}) dx dy$ ,  $\{B_i\}$  are Hermite basis functions of spline space over  $\mathcal{M}$  defined in [11],  $\mathcal{P}$  is a parameterization of  $\Omega$ .

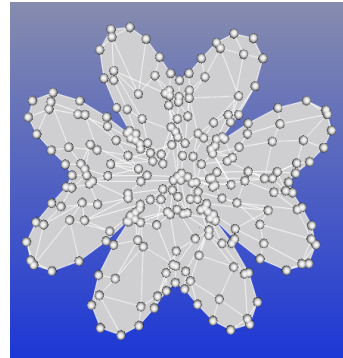
#### 4.1 A four leaf clover-like physical domain

The initial parameterization is given in Figure 10(a). In Figure 10(b), the layout of the position of vertices of this initial parameterization is presented.

For Algorithm 1, take  $\epsilon_0 = 0.001$ ,  $\Delta t = 0.1$ ,  $k = 1$  and  $\mu = 0.001$ . For this experiment, Algorithm 1 converges. With the help of Algorithm 2, the resulting

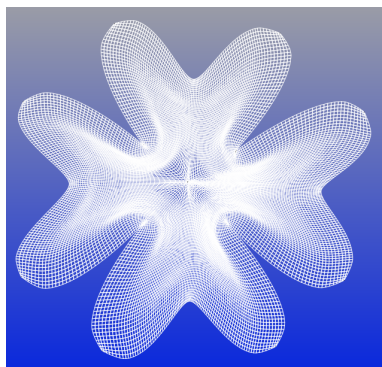


(a) The initial parameterization of the four leaf clover-like shape

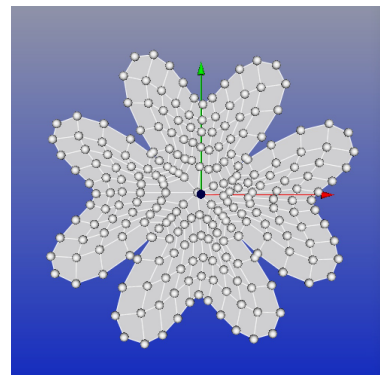


(b) The layout of the position of vertices

Figure 10: The initial parameterization and its layout of the vertex positions



(a) The resulting  $H^1$ - parameterization of the four leaf clover-like shape

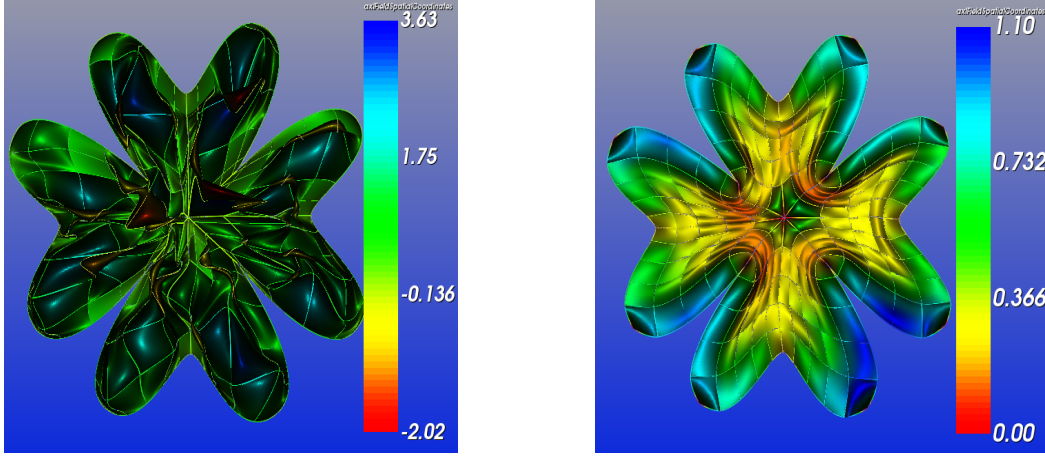


(b) The layout of the position of vertices

Figure 11: The resulting parameterization and its layout of the position of vertices

parameterization is given in Figure 11(a). And in Figure 11(b), the layout of new positions of vertices is shown.

In Figures 12(a) and 12(b), the Jacobians of the initial parameterization and the resulting parameterization are presented.



(a) The Jacobian distribution of the initial parameterization

(b) The Jacobian distribution of the resulting parameterization

Figure 12: Comparing the Jacobians of the two parameterizations

In order to study the condition number of the stiffness matrix, under the same parameters, Figure 13 presents the relationship between the iteration number of Algorithm 1 and the natural logarithm of the condition number of the stiffness matrix under the resulting parameterization with the given iteration number.

## 4.2 A perforated model structure physical domain

With the same structure as in Section 4.1, in this section, we will consider a perforated model structure. The initial parameterization is given in Figure 14(a). In Figure 14(b), the layout of the position of vertices under this initial parameterization is presented.

For Algorithm 1, take  $\epsilon_0 = 0.001$ ,  $\Delta t = 0.1$ ,  $k = 1$  and  $\mu = 0.001$ . Algorithm 1 converges. With the help of Algorithm 2, the resulting parameterization is shown in Figure 15(a). And in Figure 15(b), the layout of new positions of vertices is shown. Compare Jacobians of the initial parameterization and the resulting parameterization in Figures 16(a) and 16(b).

In order to study the condition number of the stiffness matrix, under the same parameters, Figure 17 presents the relationship between the iteration number of Algorithm 1 and the natural logarithm of the condition number of the stiffness matrix under the resulting parameterization with the corresponding iteration number.



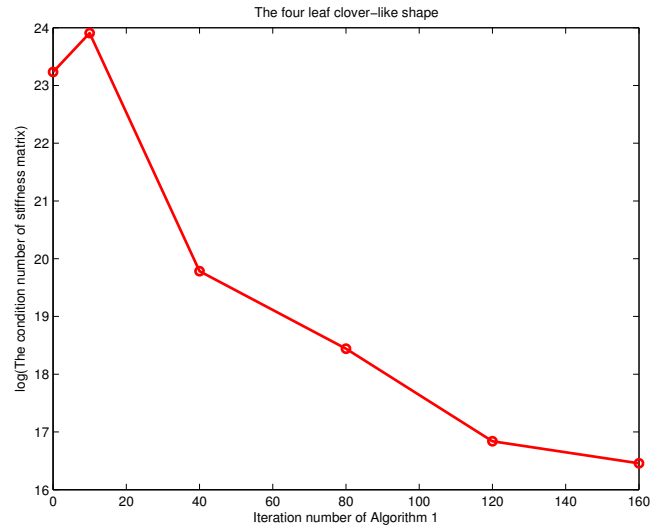


Figure 13: The relationship between the iteration number of Algorithm 1 and the condition number of the stiffness matrix

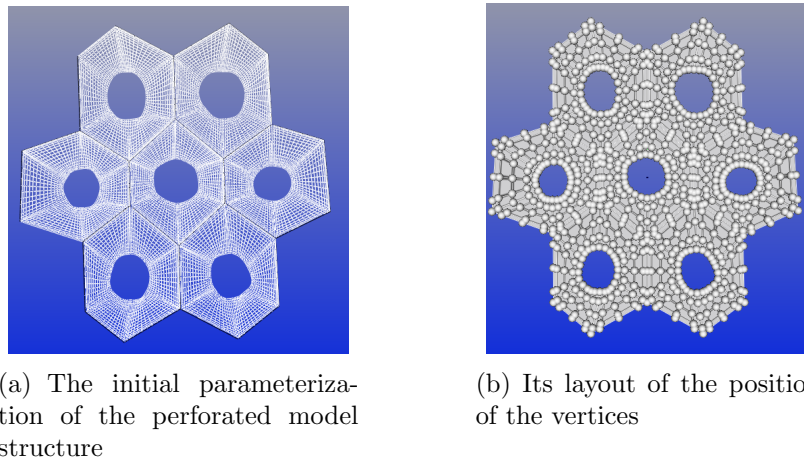
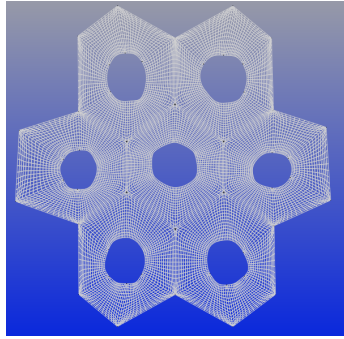
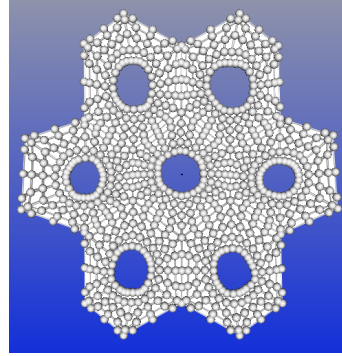


Figure 14: The initial parameterization and its layout of the position of vertices

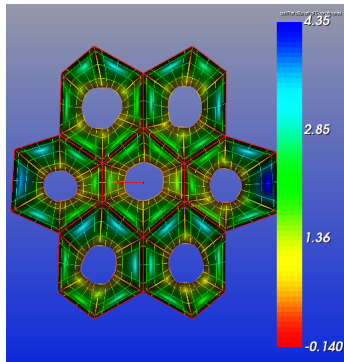


(a) The resulting  $H^1$ - parameterization of the perforated model structure

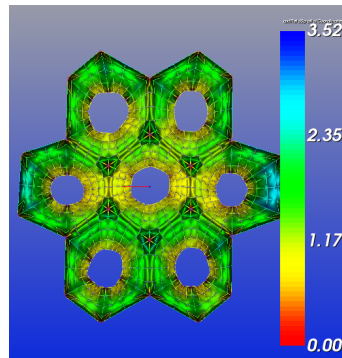


(b) Its layout of the position of vertices

Figure 15: The resulting parameterization and its layout of the position of vertices



(a) The Jacobian distribution of the initial parameterization



(b) The Jacobian distribution of the resulting parameterization

Figure 16: Comparing the Jacobians of the two parameterizations

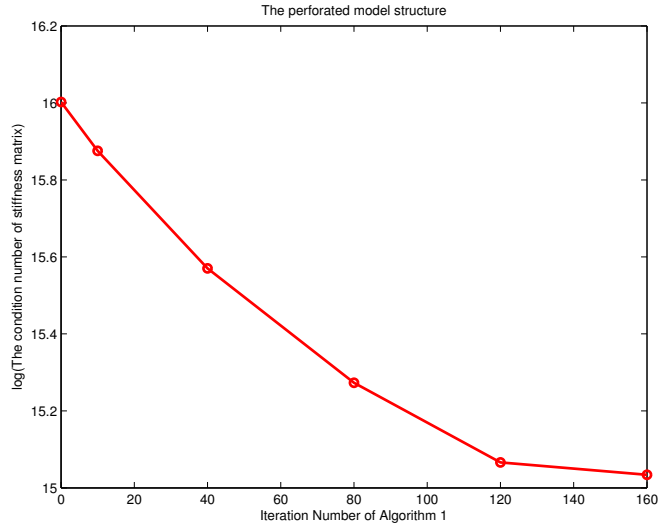


Figure 17: The relationship between the iteration number of Algorithm 1 and the natural logarithm of the condition number of the stiffness matrix

## 5 Conclusion and future work

This paper presented a method for modifying an initial parameterization with fixed boundaries over complex plane physical domains. Bi-cubic splines with rigid transition maps in [11] are used to represent globally the complex physical domains. The design of generating parameterization algorithm is based on the three basic requirements, i.e., injective parameterization (ref. Theorem 3.2), the efficiency and numerical stability (ref. Figures 13 and 17) and the regularity property of test functions (ref. Theorem 3.2).

Within IGA, for a better future solution of a second order PDE over a complex physical domain, besides the parameterization algorithm shown in this paper, an algorithm should be developed to obtain better boundaries representation for a better approximation of original physical domains.

## 6 Appendix

In this section, the proofs of lemmas and theorem are presented.

## 6.1 Splines in [11] and proofs of Lemma 2.1 and Theorem 3.2

Let  $\mathcal{P}$  be a parameterization defined over a parametric mesh  $\mathcal{M}$  and  $\mathcal{P} = \mathcal{P}_p + \mathcal{P}_t$ , where  $\mathcal{P}_p$  is the position part of  $\mathcal{P}$  and  $\mathcal{P}_t$  is the tangent part of  $\mathcal{P}$ .

Suppose that  $C = [0, 1] \times [0, 1]$  be a cell of  $\mathcal{M}$  with  $v_1(0, 0), v_2(1, 0), v_3(1, 1), v_4(0, 1)$  as its vertices. Let  $P_1, P_2, P_3, P_4$  be the position of these vertices with the help of  $\mathcal{P}_p$ , then, over  $C$ ,  $\mathcal{P}_p$  is a bicubic polynomial with vector coefficients, i. e.,

$$\mathcal{P}_p|_C(s, t) = \overrightarrow{OP_1}h_0(s)h_0(t) + \overrightarrow{OP_2}h_1(s)h_0(t) + \overrightarrow{OP_3}h_1(s)h_1(t) + \overrightarrow{OP_4}h_0(s)h_1(t), \quad (6)$$

where,  $h_0(s) = (1 + 2s)(1 - s)^2, h_1(s) = s^2(3 - 2s)$ .

Now we can prove Lemma 2.1:

**Proof.** Suppose  $\overrightarrow{OP_i} = (x_i, y_i), i = 1, 2, 3, 4$ . Then,

$$\mathcal{P}_p|_C(s, t) = (x(s, t), y(s, t));$$

where,  $x(s, t) = \sum_{i=1}^4 x_i B_i(s, t), y(s, t) = \sum_{i=1}^4 y_i B_i(s, t)$ . Thus,

$$\frac{\partial x}{\partial s}(0, t) = 0; \frac{\partial x}{\partial s}(1, t) = 0; \frac{\partial x}{\partial t}(s, 0) = 0; \frac{\partial x}{\partial t}(s, 1) = 0;$$

$$\frac{\partial y}{\partial s}(0, t) = 0; \frac{\partial y}{\partial s}(1, t) = 0; \frac{\partial y}{\partial t}(s, 0) = 0; \frac{\partial y}{\partial t}(s, 1) = 0;$$

Then Jacobians along the mesh lines,

$$\mathcal{J}(0, t) = \begin{vmatrix} \frac{\partial x}{\partial s}(0, t) & \frac{\partial x}{\partial t}(0, t) \\ \frac{\partial y}{\partial s}(0, t) & \frac{\partial y}{\partial t}(0, t) \end{vmatrix} = 0; \quad \mathcal{J}(1, t) = \begin{vmatrix} \frac{\partial x}{\partial s}(1, t) & \frac{\partial x}{\partial t}(1, t) \\ \frac{\partial y}{\partial s}(1, t) & \frac{\partial y}{\partial t}(1, t) \end{vmatrix} = 0;$$

$$\mathcal{J}(s, 0) = \begin{vmatrix} \frac{\partial x}{\partial s}(s, 0) & \frac{\partial x}{\partial t}(s, 0) \\ \frac{\partial y}{\partial s}(s, 0) & \frac{\partial y}{\partial t}(s, 0) \end{vmatrix} = 0; \quad \mathcal{J}(s, 1) = \begin{vmatrix} \frac{\partial x}{\partial s}(s, 1) & \frac{\partial x}{\partial t}(s, 1) \\ \frac{\partial y}{\partial s}(s, 1) & \frac{\partial y}{\partial t}(s, 1) \end{vmatrix} = 0.$$

□

Now we can prove Theorem 3.2.

**Proof.** Let  $\mathbf{P}_i = \overrightarrow{OP_i}$ , where  $O$  is the origin of the plane of the physical domain.

**Case 1:** If  $v_1, v_2, v_3, v_4$  are regular vertices of  $\mathcal{M}$ ,

$$\mathcal{P}|_C(s, t) = \mathbf{T}(s, t) = \sum_{i,j=0}^3 C_{i,j} B_i^{(4)}(s) B_j^{(4)}(t),$$

where  $C = (C_{i,j})_{i,j=0}^3 =$

$$\begin{pmatrix} \mathbf{P}_1 & \frac{1}{3}\mathbf{P}_2 + \frac{2}{3}\mathbf{P}_1 & \frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_1 & \mathbf{P}_2 \\ \frac{2}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_4 & \frac{1}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_4 & \frac{1}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3 & \frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3 \\ \frac{1}{3}\mathbf{P}_1 + \frac{2}{3}\mathbf{P}_4 & \frac{1}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_4 & \frac{1}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_4 & \frac{2}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_2 \\ \mathbf{P}_4 & \frac{1}{3}\mathbf{P}_3 + \frac{2}{3}\mathbf{P}_4 & \frac{2}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_4 & \mathbf{P}_3 \end{pmatrix}$$

And,

$$\frac{\partial \mathbf{T}}{\partial s} = \frac{1}{3}\lambda_{1,1}(s,t)(\mathbf{P}_3 - \mathbf{P}_4) + \frac{1}{3}\lambda_{1,2}(s,t)(\mathbf{P}_2 - \mathbf{P}_1), \quad (7)$$

where

$$\begin{aligned} \lambda_{1,1}(s,t) &= B_3^{(4)}(t) + B_2^{(4)}(t)(B_0^{(3)}(s) + B_2^{(3)}(s)) + B_1^{(4)}(t)B_1^{(3)}(s) \geq 0; \\ \lambda_{1,2}(s,t) &= B_0^{(4)}(t) + B_1^{(4)}(t)(B_0^{(3)}(s) + B_2^{(3)}(s)) + B_2^{(4)}(t)B_1^{(3)}(s) \geq 0. \end{aligned}$$

$$\frac{\partial \mathbf{T}}{\partial t} = \frac{1}{3}\lambda_{2,1}(s,t)(\mathbf{P}_4 - \mathbf{P}_1) + \frac{1}{3}\lambda_{2,2}(s,t)(\mathbf{P}_3 - \mathbf{P}_2), \quad (8)$$

where

$$\begin{aligned} \lambda_{2,1}(s,t) &= B_0^{(4)}(s) + B_1^{(4)}(s)(B_0^{(3)}(t) + B_2^{(3)}(t)) + B_2^{(4)}(s)B_1^{(3)}(t) \geq 0; \\ \lambda_{2,2}(s,t) &= B_3^{(4)}(s) + B_2^{(4)}(s)(B_0^{(3)}(t) + B_2^{(3)}(t)) + B_1^{(4)}(s)B_1^{(3)}(t) \geq 0. \end{aligned}$$

$(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_4 - \mathbf{P}_1)$ ,  $(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_2)$ ,  $(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1)$  and  $(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2)$  have the same direction. Because the quadrilateral given by  $P_1, P_2, P_3, P_4$  is convex. Thus, the Jacobian of  $\mathbf{T}(s, t)$ , denoted as  $\mathcal{J}(s, t)$ , is positive or negative, if and only if

$$\left\| \frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} \right\| \neq 0.$$

Moreover,

$$\left\| \frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} \right\| = 0,$$

if and only if

$$\begin{cases} \lambda_{1,1}(s,t)\lambda_{2,1}(s,t) = 0; \\ \lambda_{1,1}(s,t)\lambda_{2,2}(s,t) = 0; \\ \lambda_{1,2}(s,t)\lambda_{2,1}(s,t) = 0; \\ \lambda_{1,2}(s,t)\lambda_{2,2}(s,t) = 0. \end{cases} \quad (9)$$

However, for all  $(s, t) \in [0, 1] \times [0, 1]$ , it does not satisfy (9). Thus,  $\mathcal{J}(s, t) > 0$  (or  $< 0$ ) for all  $(s, t) \in C$ , i.e.,  $\mathcal{P}|_C(s, t)$  is injective.

Based on  $\mathcal{J}(s, t) > 0$  (or  $< 0$ ) and the continuity of  $\mathcal{P}|_C(s, t)$ , locally, there exists a  $C^1$  inverse map of  $\mathcal{P}|_C(s, t)$ . Thus, the image of splines on the physical domain is  $C^1$  ( $\subset H^1$ ).

**Case 2:** If  $v_1$  is an extraordinary vertex and  $v_2, v_3, v_4$  are regular vertices of  $\mathcal{M}$ ,

$$\mathcal{P}|_C(s, t) = \mathbf{T}(s, t) = \sum_{i,j=0}^3 C_{i,j} B_i^{(4)}(s) B_j^{(4)}(t),$$

where  $C = (C_{i,j})_{i,j=0}^3 =$

$$\begin{pmatrix} \mathbf{P}_1 & \mathbf{P}_1 & \frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_1 & \mathbf{P}_2 \\ \mathbf{P}_1 & \mathbf{P}_1 & \frac{1}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3 & \frac{2}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3 \\ \frac{1}{3}\mathbf{P}_1 + \frac{2}{3}\mathbf{P}_4 & \frac{1}{3}\mathbf{P}_1 + \frac{1}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_4 & \frac{1}{3}\mathbf{P}_2 + \frac{1}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_4 & \frac{2}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_2 \\ \mathbf{P}_4 & \frac{1}{3}\mathbf{P}_3 + \frac{2}{3}\mathbf{P}_4 & \frac{2}{3}\mathbf{P}_3 + \frac{1}{3}\mathbf{P}_4 & \mathbf{P}_3 \end{pmatrix}.$$

And,

$$\frac{\partial \mathbf{T}}{\partial s} = \frac{1}{3}\mu_{1,1}(s, t)(\mathbf{P}_3 - \mathbf{P}_4) + \frac{1}{3}\mu_{1,2}(s, t)(\mathbf{P}_2 - \mathbf{P}_1) + \frac{1}{3}\mu_{1,3}(s, t)(\mathbf{P}_3 - \mathbf{P}_1), \quad (10)$$

where,

$$\begin{aligned} \mu_{1,1}(s, t) &= B_3^{(4)}(t) + B_2^{(4)}(t)(B_0^{(3)}(s) + B_2^{(3)}(s)) \geq 0; \\ \mu_{1,2}(s, t) &= B_1^{(3)}(s)(2B_0^{(4)}(t) + B_1^{(4)}(t) + B_2^{(4)}(t)) + B_2^{(3)}(s)(B_0^{(4)}(t) + B_1^{(4)}(t)) \geq 0; \\ \mu_{1,3}(s, t) &= B_2^{(3)}(s)B_1^{(4)}(t) \geq 0. \end{aligned}$$

$$\frac{\partial \mathbf{T}}{\partial t} = \frac{1}{3}\mu_{2,1}(s, t)(\mathbf{P}_4 - \mathbf{P}_1) + \frac{1}{3}\mu_{2,2}(s, t)(\mathbf{P}_3 - \mathbf{P}_2) + \frac{1}{3}\mu_{2,3}(s, t)(\mathbf{P}_3 - \mathbf{P}_1), \quad (11)$$

where,

$$\begin{aligned} \mu_{2,1}(s, t) &= B_0^{(4)}(s)(2B_1^{(3)}(t) + B_2^{(3)}(t)) + B_1^{(4)}(s)(B_1^{(3)}(t) + B_2^{(3)}(t)) + B_2^{(4)}(s)B_1^{(3)}(t) \geq 0; \\ \mu_{2,2}(s, t) &= B_3^{(4)}(s) + B_2^{(4)}(s)(B_0^{(3)}(t) + B_2^{(3)}(t)) \geq 0; \\ \mu_{2,3}(s, t) &= B_1^{(4)}(s)B_1^{(3)}(t) \geq 0. \end{aligned}$$

Moreover, the following vectors share the same direction:

$$\begin{aligned} &(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_4 - \mathbf{P}_1), (\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_2), (\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_1), \\ &(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1), (\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2), (\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_1), \\ &(\mathbf{P}_3 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1), (\mathbf{P}_3 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2) \end{aligned}$$

because the quadrilateral given by  $P_1, P_2, P_3, P_4$  is convex. Thus,  $\mathcal{J}(s, t)$  is positive or negative, if and only if

$$\left\| \frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} \right\| \neq 0.$$

Moreover,

$$\left\| \frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} \right\| = 0,$$

if and only if

$$\begin{cases} \mu_{1,1}(s,t)\mu_{2,1}(s,t) = 0; \\ \mu_{1,1}(s,t)\mu_{2,2}(s,t) = 0; \\ \mu_{1,1}(s,t)\mu_{2,3}(s,t) = 0; \\ \mu_{1,2}(s,t)\mu_{2,1}(s,t) = 0; \\ \mu_{1,2}(s,t)\mu_{2,2}(s,t) = 0; \\ \mu_{1,2}(s,t)\mu_{2,3}(s,t) = 0; \\ \mu_{1,3}(s,t)\mu_{2,1}(s,t) = 0; \\ \mu_{1,3}(s,t)\mu_{2,2}(s,t) = 0. \end{cases} \quad (12)$$

There is only one solution of (12), i.e.  $(0,0) \in [0,1] \times [0,1]$ . Thus,

$$\mathcal{J}(s,t) = \begin{cases} = 0, & (s,t) = (0,0); \\ > 0(\text{or } < 0), & (s,t) \neq (0,0) \end{cases}$$

i.e.,  $\mathcal{P}|_C(s,t)$  is injective, because  $(0,0)$  is isolated.

Now, the parameterization  $\mathcal{P}|_C(s,t)$  is singular at  $(0,0)$ . However, the coefficients satisfy the  $H^1$ -integrability assumption (Assumption 5.1 in [19]) when the quadrilateral given by  $P_1, P_2, P_3, P_4$  is convex. Thus the images of splines on the physical domain are  $H^1$ .  $\square$

## 6.2 Proof of Lemma 3.1

**Proof.** Because the velocity of  $P_i(t)$  at  $t_0$  is  $\mathbf{0}$ ,

$$\begin{aligned} P_i(t_0 + \Delta t) - P_i(t_0) &= \int_{t_0}^{t_0 + \Delta t} dt \int_{t_0}^t \frac{\mathbf{F}_i(s)}{m_i} ds \\ &= \int_{t_0}^{t_0 + \Delta t} \frac{\mathbf{F}_i(s)}{m_i} (t_0 + \Delta t - s) ds \\ &= \frac{\mathbf{F}_i(t_0 + \theta \Delta t)}{m_i} \int_{t_0}^{t_0 + \Delta t} (t_0 + \Delta t - s) ds \\ &= \frac{\mathbf{F}_i(t_0 + \theta \Delta t)}{2m_i} \Delta t^2 \end{aligned}$$

where  $\theta \in [0,1]$ . Thus,

$$P_i(t_0 + \Delta t) - P_i(t_0) = \frac{\mathbf{F}_i(t_0) \Delta t^2}{2m_i} + o(\Delta t^2).$$

□

## References

- [1] 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*, vol. 194, Issues. 39-41, p. 4135–4195, 2005.
- [2] E. Cohen, T. Martin, R.M. Kirby, T. Lyche, R.F. Riesenfeld. “Analysis-aware Modeling: Understanding Quality Considerations in Modeling for Isogeometric Analysis”. *Comput. Methods Appl. Mech. Engrg.* vol. 199, p. 334-356, 2010.
- [3] T. Martin, E. Cohen, M. Kirby, T. Lyche, R. Riesenfeld. “Analysis-aware modeling: quality considerations in modeling for isogeometric analysis”. *Comput. Method Appl. Mech. Engrg.* vol. 199, p. 334356, 2010.
- [4] W. Wang, Y. Zhang. “Wavelets-based NURBS simplification and fairing”. *Comput. Method Appl. Mech. Engrg.* vol. 199, p. 290300, 2010.
- [5] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. “parameterization of computational domain in isogeometric analysis: methods and comparison”. *Comput. Methods Appl. Mech. Engrg.* vol. 200, p. 2021-2031, 2011.
- [6] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. “Optimal analysis-aware parameterization of computational domain in 3D isogeometric analysis”. *Computer-Aided Design.* vol. 45, p. 812-821, 2013.
- [7] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo. “Constructing analysis-suitable parameterization of computational domain from CADboundary by variational harmonic method”. *Journal of Computational Physics.* vol. 252, p. 275-289, 2013.
- [8] E. Pilgerstorfer, B. Jüttler . “Bounding the influence of domain parameterization and knot spacing on numerical stability in Iso-geometric Analysis”. *Computer Methods in Applied Mechanics and Engineering*, vol. 268, p. 589-613, 2014.
- [9] K. Gahalaut, S. Tomar. “Condition number estimates for matrices arising in the isogeometric discretizations”. *RICAM report 23*, p. 1-38, 2012.
- [10] Hyun-Jung Kim, Yu-Deok Seo, and Sung-Kie Youn. “Isogeometric analysis with trimming technique for problems of arbitrary complex topology”. *Computer Methods in Applied Mechanics and Engineering.* vol. 199, p. 2796-2812, 2010.



- [11] M. Wu, B. Mourrain, A. Galligo, B. Nkonga. “Spline spaces over rectangular meshes with arbitrary topologies and its application to the Grad-Shafranov equation”. *submitted to Comput. Methods Appl. Mech. Engrg.* <https://hal.inria.fr/hal-01196428/>, 2015
- [12] S. Lipton, J.A. Evans, Y. Bazilevs, T. Elguedj, T.J.R. Hughes. “Robustness of isogeometric structural discretizations under severe mesh distortion”. *Comput. Methods Appl. Mech. Engrg.* vol. 199, p. 357-373, 2010.
- [13] T. Kamada, S. Kawai. “An algorithm for drawing general undirected graphs”. *Information Processing Letters.* vol. 31, p. 7-15, 1989.
- [14] B. Becker, G. Hotz. “On the optimal layout of planar graphs with fixed boundary”. *SIAM J. Comput.* vol. 16, p. 946-972, 1987.
- [15] V.M. Calo, H. Gómez, Y. Bazilevs, G.P. Johnson, T.J.R. Hughes, “Simulation of Engineering Applications Using Isogeometric Analysis”. *Proceedings of Tera Grid.* 2008.
- [16] W. T. Tutte, “How to draw a graph”. *Proc. London Math. Society.* vol. 13, p. 743-768, 1963.
- [17] P. Eades, “A heuristic for graph drawing”. *Congressus Numerantium.* vol. 42, p. 149-160, 1984.
- [18] J. Joseph Fowler, Stephen G. Kobourov, “Planar Preprocessing for Spring Embedders.” *Graph Drawing.* p: 388-399, 2012.
- [19] T. Takacs, B. Jüttler. “Existence of Stiffness Matrix Integrals for Singularly Parameterized Domains in Isogeometric Analysis”. *Computer Methods in Applied Mechanics and Engineering,* vol. 200, p. 3568–3582, 2011.
- [20] T. Takacs, B. Jüttler. “ $H^2$  regularity properties of singular parameterizations in isogeometric analysis”. *Graphical Models,* vol. 74, p. 361–372, 2012.