H1-parametrizations of complex planar physical domains in Isogeometric analysis

Meng Wu, Bernard Mourrain, André Galligo, Boniface Nkonga

H^1-parametrizations of complex planar physical domains in isogeometric analysis

Meng Wu[a,*], Bernard Mourrain[b], André Galligo[c], Boniface Nkonga[c]

[a]*School of Mathematics, Hefei University of Technology, P. R. China.*
[b]*Aromath, Inria, Sophia Antipolis, France*
[c]*Lab. J. A. Dieudonné, University of Nice, Nice, France*

Abstract

Isogeometric analysis is a method for solving geometric partial differential equations (PDEs). Generating parametrizations of a PDE's physical domain is a basic and important issue within isogeometric analysis framework. In this paper, we present a global H^1-parametrization method for a complex planar physical domain.

Keywords: H^1-parametrization, complex planar physical domains, isogeometric analysis

1. Introduction

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

A map to describe a physical domain is called a parametrization of the physical domain. A parametrization has an impact on the simulation results, efficiency of the computation, and stability of numerical system [2, 3, 4]. So far, there is no simple criterion to measure the quality of a parametrization. In the following, we list three basic requirements in recent works on this topic.

First, parametrizations are injective. For describing the geometry of physical domains, parametrizations should have no self-intersection, such as in [5, 6, 7].

Second, a uniform parametrization is expected to obtain an efficient and stable solution process. For numerical analysis, the condition number of the linear system obtained within isogeometric analysis framework should be as small as possible [8, 9]. Smaller condition numbers bring efficiency and numerical stability. In applications, a

[*] Corresponding author at: School of Mathematics, Hefei University of Technology, No. 193, Tunxi Road, Hefei, Anhui Prov., 230009, P. R. China.

Email addresses: meng.wu@hfut.edu.cn, wumeng@mail.ustc.edu.cn (Meng Wu), Bernard.Mourrain@inria.fr (Bernard Mourrain), andre.galligo@unice.fr (André Galligo), boniface.nkonga@unice.fr (Boniface Nkonga)

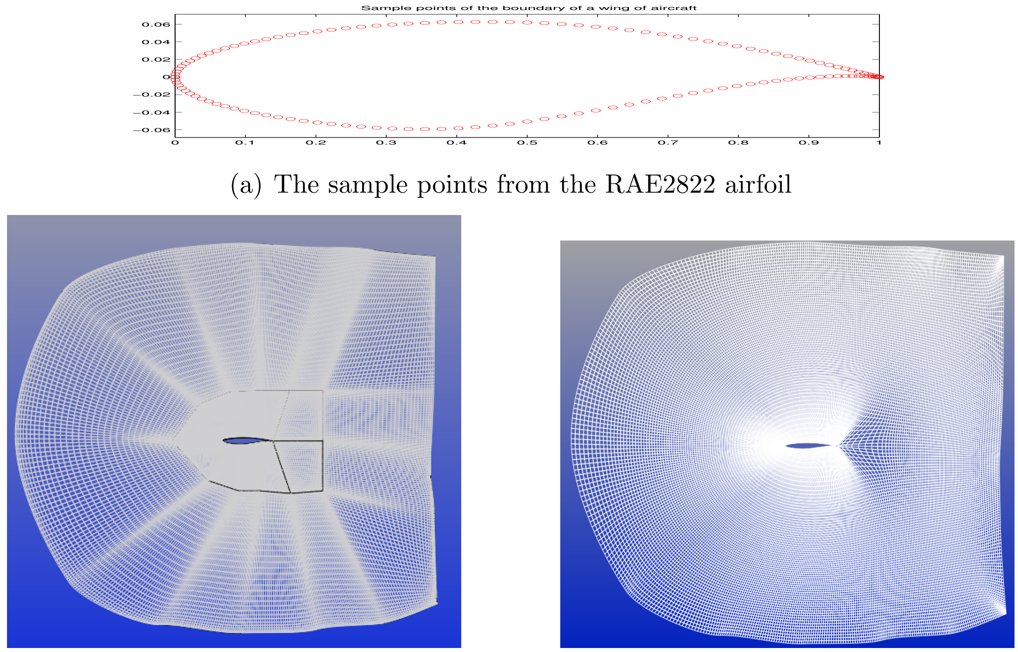
linear system is always large. It is better to use iterative solvers to reduce the computational costs. A smaller condition number means a rapid convergence and a more accurate linear system solution. However, this condition number depends on the underlying parametrization. To obtain a smaller condition number, [8] suggests that the size of elements on the physical domain should not vary too much, the parameters lines should be as orthogonal as possible, and the parameter directions should have approximately the same length. Moreover, in the classical FEM, there are many results regarding the stiffness matrix and bounds of its condition number. Normally, a condition number is affected by the size [10, 11] and geometry [12, 13] of elements on a given physical domain and employed basis functions [14, 15, 16]. From the results of the condition number bounds in isogeometric analysis and the classical FEM, a uniform parametrization is expected when the employed basis functions are given.

Third, parametrizations should satisfy some additional conditions that guarantee the regularity properties of the test functions on a physical domain if there are singularities, see [17, 18]. The test functions used in Galerkin-based isogeometric analysis are obtained by composing the inverse of a parametrization with basis functions. Thus, singularities of a parametrization impact the regularity property of the test functions, i.e., the regularity property of test functions should be considered if singularities cannot be avoided. For example, in this paper, a resulting parametrization satisfies the H^1 integrability assumption outlined in [17], i.e., test functions are H^1 at least on a planar physical domain. Thus, it is called a H^1-parametrization. According to H^1-parametrizations, isogeometric technique can be used to solve second order PDEs over a planar physical domain.

Based on these three basic requirements, a reparametrization algorithm is presented in this paper to obtain a global uniform injective H^1-parametrization of a complex planar physical domain. Usually Non-Uniform Rational B-Splines (NURBS) are treated as basis functions of classical isogeometric analysis, i.e., NURBS-based isogeometric analysis. When we consider a complex planar physical domain (especially with a complex topology), patching multiple NURBS parametrizations or trimming techniques, such as in[19], need to be considered. To obtain a global representation of a physical domain with complex (geometric or topological) features, bicubic splines with rigid transformations, as in [20], are used to represent the global parametrizations.

This algorithm can be applied to several situations. For example, to simulate the two-dimensionality of the flow over the RAE2822 airfoil, the physical domain is a bounded domain around it. The initial parametrization (Figure 1(b)) is produced by fitting the sample points shown in Figure 1(a) of its boundary. Thus, during the reparametrization process, fixed boundaries are expected. Applying the reparametrization algorithm in this paper, the H^1-parametrization is shown in Figure 1(c).

This paper is structured as follows. In Section 2, global parametrizations are introduced and some properties of parametrizations are presented as background. Then, in Section 3, we develop the reparametrization algorithm. Using this algorithm, in Section 4, several experiments are presented to reparametrize physical domains. These physical domains include a square domain, complex geometric features (such as sharp



(a) The sample points from the RAE2822 airfoil

(b) The initial parametrization by fitting the sample points

(c) The H^1-parametrization

Figure 1: parametrizations of the physical domain around RAE2822 airfoil

corners and non-convex shapes) and complex topological features (such as many holes inside). In Section 5, we conclude this paper by indicating directions for future work. Lastly, the proofs of lemmas are collected in Section 6 as an appendix.

2. Parametrizations with splines defined over meshes with complex topologies

In this paper, C^1 splines defined over a mesh with a complex topology are used to represent parametrizations, where C^1 means the parametrical smoothness order is 1. There are similar works in geometric modelling, such as [21]. In [21], the author defines a spline space that meets the parametrical smoothness conditions. This spline space consists of spline surfaces for applications in geometric modelling. Although there are singularities at extraordinary vertices, the regularity of spline surfaces on a physical domain can be obtained by restricting the higher order derivatives at extraordinary vertices, i.e., certain neighboring Bézier points are coplanar. Similar to the approach taken by [21], C^1 splines are singular at extraodinary vertices. For applications in isogeometric analysis, they are scalar functions used as shape functions, trial functions, test functions by parametrizations. The regularity of test/trial functions on a physical domain is obtained by designing a proper parametrization that satisfies an integrability assumption around the extraordinary vertex, such as an H^1-integrability assumption in this paper. Restriction of the discussion to planar domains leads to a smaller number of constraints with respect to [21], where non-planar domains were considered, and as a result C^1 splines have more degrees of freedoms available to approximate numerical solutions. In this section, we give an intuitive explanation of C^1 splines. One can refer to [20] for the precise mathematical definitions. Based on this intuitive explanation, the parametrization represented by these splines is decomposed into two parts. The different modification algorithms for these two parts are presented in Section 3 later.

2.1. An intuitive introduction to splines

Before introducing the splines, parametric meshes are presented, where 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 that contains anisotropic rectangles with an 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 valence of each vertex to obtain a global parametrization without trimming surfaces with more complex topology, such as surfaces with holes inside.

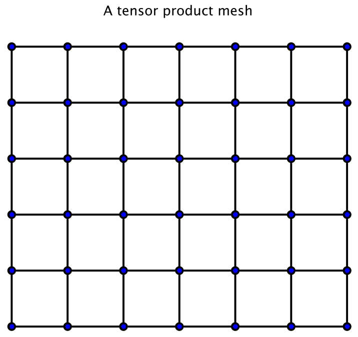


Figure 2: A 2D tensor product mesh

For example, in Figure 3, there are some undirected graphs that share the same topology with corresponding parametric meshes $\mathscr{M}$. They are denoted as $\mathscr{M}$ as well.

In the following, we will rely on illustrations to introduce some concepts. In Figure 4(a), $\{v^i : i = 1, 2 \cdots, 10\}$ and $\{v_b^j : j = 1, 2, \cdots, 20\}$ are vertices of $\mathscr{M}$. v_b^j $(j = 1, 2, \cdots, 20)$ are **boundary vertices** and $v^i (i = 1, 2, \cdots, 10)$ are **interior vertices**. If the valence 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 $\mathscr{M}$ connects adjacent vertices. In Figure 4(a), $v_b^{15} v_b^{20}, v^1 v^2, v^1 v_b^1$ are edges of $\mathscr{M}$. Here, $v_b^{15} v_b^{20}$ is a **boundary edge** of $\mathscr{M}$, while $v^1 v^2, v^1 v_b^1$ are interior edges. Quadrilaterals with curved edges, such as $v_b^{15} v_b^{20} v^{10} v^1$, $v^3 v^4 v_b^{18} v_b^{17}$ and $v_b^5 v_b^6 v^6 v^5$, are treated as cells of $\mathscr{M}$.

Local frame of a cell $\boldsymbol{C}$**:** 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.

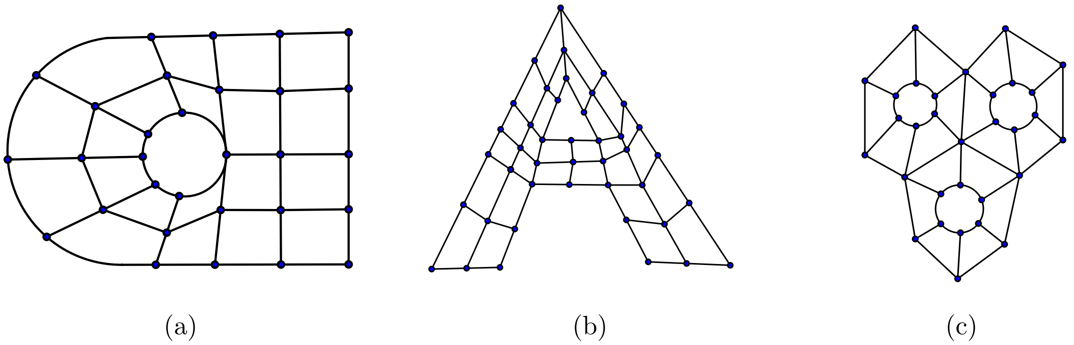


Figure 3: Examples of the topology of parametric meshes ($\mathscr{M}$s)

Local parameters of a cell C***:*** A global spline on $\mathscr{M}$ 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 according to the local parameters (s_C, t_C).

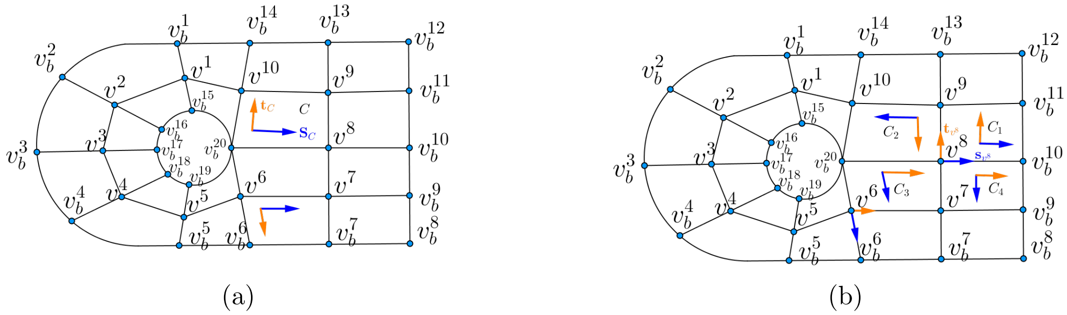


Figure 4: An example of $\mathscr{M}$ for introducing the concepts of parametric meshes.

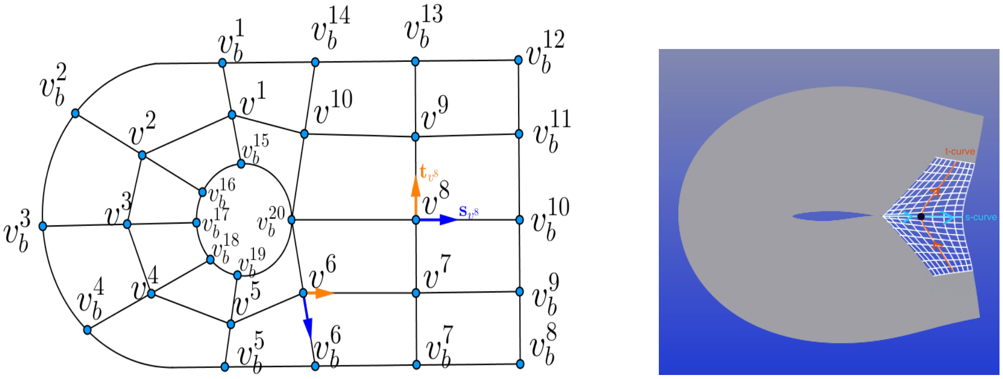
Associated with each vertex of $\mathscr{M}$, splines will be introduced to represent parametrizations in the following context. Based on the different types of vertices, splines will be classified into two classes.

1. For an extraordinary vertex v_{ir}, there is a spline $B^p_{v_{ir}}$ 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 $B^p_{v_r}$, $B^s_{v_r}$ and $B^t_{v_r}$ will be defined over $\mathscr{M}$. For $B^p_{v_r}$, its coefficient $\mathbf{P}_{v_r} \in \mathbb{R}^3$ stands for the position of v_r. $B^p_{v_r}$ is called the position spline of v_r. $B^s_{v_r}$ and $B^t_{v_r}$ are called **tangent splines**.

To explain the coefficients of $B^s_{v_r}$ and $B^t_{v_r}$, 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 $(\mathbf{s}_{v^8}, \mathbf{t}_{v^8})$ that is the same as the local frame of C_1. $\mathbf{T}^s_{v_r}$ (or $\mathbf{T}^t_{v_r}$) is the tangent vector of the s-curve (or t-curve) at v_r, where the s-curve (or t-curve) is the image of the mesh grid line of parametrization along the $\mathbf{s}_{v_r}$ (or $\mathbf{t}_{v_r}$) direction. For example, in Figure

5(b), the domain is an image of one of the parametrizations defined over $\mathscr{M}$. By this parametrization, the s-curve consists of the image of the edges $v_b^{20}v^8$, $v^8v_b^{10}$ of the direction that matches with the local frame $\mathbf{s}_{v^8}$ attached to v^8. The t-curve can be defined similarly.



(a) $\mathscr{M}$

(b) The image of one of parametrizations

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

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

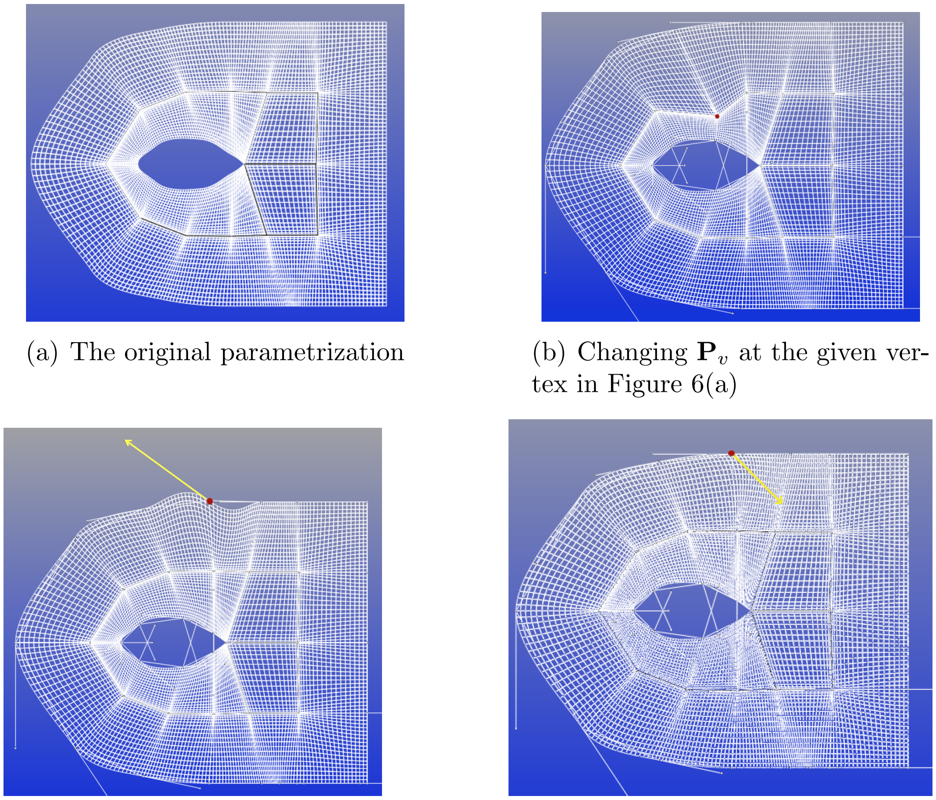
$$\mathscr{P} = \sum_{v \in V} \mathbf{P}_v B_v^p + \sum_{v_r \in V_r} (\mathbf{T}_{v_r}^s B_{v_r}^s + \mathbf{T}_{v_r}^t B_{v_r}^t) \qquad (1)$$

where V and V_r are the set of vertices and the set of regular vertices of $\mathscr{M}$, respectively. In this paper, parametrizations 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 parametrization. In Figure 6(b), by changing the $\mathbf{P}_v$ 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 parametrization has been changed around the given vertex.

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

$$\left| \begin{matrix} \frac{\partial x(s_C,t_C)}{\partial s_C} & \frac{\partial x(s_C,t_C)}{\partial t_C} \\ \frac{\partial y(s_C,t_C)}{\partial s_C} & \frac{\partial y(s_C,t_C)}{\partial t_C} \end{matrix} \right|_p .$$

Remark 2.1. *1. In [20], at the regular vertex v^r, there is another spline $B_{v^r}^{st}$. From the view of differential calculus, elements can be approximated by parallelograms if they are small enough and distribute uniformly on a given physical domain. There is a standard uniform map from a rectangle to a parallelogram by polynomials of degree 1. If we represent these polynomials by Hermite splines in [20], the coefficients of $B_{v^r}^{st}$ are zeros. Based on this observation, in this paper, we develop the reparametrization algorithm*



(a) The original parametrization

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

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

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

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

only based on B_v^p, $B_{v_r}^s$ and $B_{v_r}^t$. For the uniformity of the resulting parametrization by the reparametrization algorithm in this paper, we state in the second point of Remark 3.3 and provide Example 3.2 later.

2. If the Jacobian of $\mathscr{P}$ at p is zero, p is called a singularity of $\mathscr{P}$ and $\mathscr{P}$ is singular at p. Test functions used in isogeometric analysis compose the inverse of a parametrization with spline basis functions. However, $\mathscr{P}$ is singular at extraordinary vertices. To guarantee the regularity of test functions on a physical domain, the parametrization algorithm developed in this paper satisfies the H^1 integrability assumption outlined in [17] (see Theorem 3.4).

2.2. Parametrizations of planar physical domains

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

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

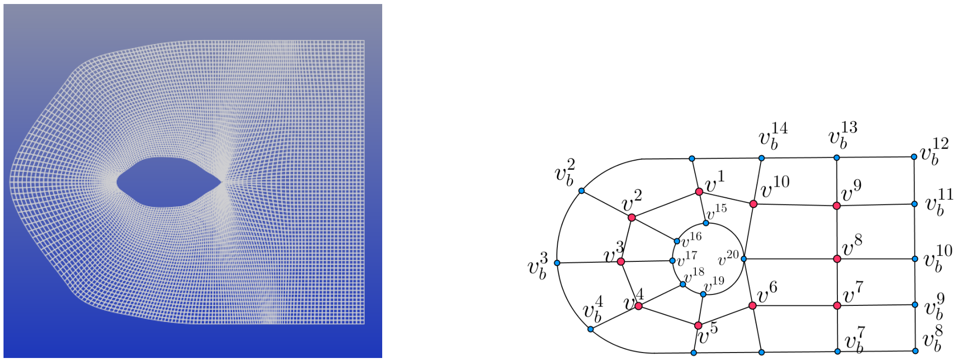
where $\mathscr{P}_p = \sum_{v \in V} \mathbf{P}_v B_v^p$, $\mathscr{P}_t = \sum_{v_r \in V_r} (\mathbf{T}_{v_r}^s B_{v_r}^s + \mathbf{T}_{v_r}^t B_{v_r}^t)$. $\mathbf{P}_v$ is the position coordinates of the vertex v. $\mathbf{T}_{v_r}^s$ and $\mathbf{T}_{v_r}^t$ are the tangent vectors of the s-curve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

$\mathscr{P}_p$: For this part of the parametrization, the coefficients of the position splines are the positions of the vertices on the physical domain. We modify $\mathscr{P}_p$ by distributing the layout of an undirected graph [22, 23] uniformly in Section 3.1. Take the parametrization presented in Figure 6(a), for example. This parametrization is only represented by the position splines at each interior vertex, i.e., the coefficients of $\mathbf{T}^s_{v_r}, \mathbf{T}^t_{v_r}$ are zero. The parametric lines gather at s-curves and t-curves at each vertex. The following lemma explains this phenomenon and it is proved in Section 6.

Lemma 2.2. *The Jacobian of $\mathscr{P}_p$ is zero along s-curves and t-curves at any vertex of $\mathscr{M}$.*

Therefore, it is necessary to add a non-trivial $\mathscr{P}_t$ to obtain a uniform parametrization within each element and a better Jacobian.

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



(a) Adding an non-trivial $\mathscr{P}_t$ at the regular interior vertices to the parametrization 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 parametrization algorithm

In this section, we develop a reparametrization algorithm with fixed boundaries of $\mathscr{P}_0$ to obtain a uniform injective H^1-parametrization, where $\mathscr{P}_0$ is a given initial parametrization.

This reparametrization algorithm consists of two steps (Sections 3.1 and 3.2). In the first step (Section 3.1), we distribute the layout of an undirected graph with fixed vertex positions on the boundaries as uniform as possible within the given physical domain. In other words, we re-determine the coefficients of position splines at interior vertices such that elements on the physical domain distribute as uniformly as possible. The second step is the smoothing algorithm, presented in Section 3.2. By modifying the tangent splines' coefficients at each interior regular vertex, this step ensures that the resulting parametrization is as uniform as possible in each element. At the end of this section (Section 3.3), we conclude this reparametrization algorithm with fixed boundaries and analyze it.

3.1. The force-directed algorithm

To distribute the layout of an undirected graph with fixed vertex positions on boundaries as uniformly as possible, in this section, we present an algorithm based on the mass-spring model and the point-charge model, which is used as an auxiliary technique.

On the one hand, considering a physical phenomenon, a linear elastic, isotropic and homogeneous membrane with fixed boundaries must be non-overlapping and uniform if the boundaries are non-overlapping. Because the mass-spring model can be derived from continuum mechanics [24], it is frequently used to model deformable objects in computer graphics [25, 26, 27, 28]. We simulate this elastic deformation process to reparametrize the physical domain uniformly and without overlapping by the classical mass-spring model. In addition, in graph drawing of graph theory, force-directed algorithms known as "spring embedders" are among the most flexible methods for calculating layouts of simple undirected graphs (see [29], a survey [30]). These force-directed algorithms tend to be aesthetically pleasing, exhibit symmetries, and tend to produce crossing-free layouts for planar graphs.

On the other hand, the point-charge model is introduced as an auxiliary technique. If we simulate this elastic model only by the mass-spring model on a concave planar physical domain, the interior vertices at concave corners may be outside of this given physical domain. Unlike in applications in computer graphics and graph drawing, the boundary description is important for solving PDEs numerically. Thus, we suppose that the boundaries of a physical domain and the vertices of its undirected graphs carry the same type of charges to restrict the positions of the interior vertices within the given physical domain.

In conclusion, a spring-dominant energy that evolves with time is presented. Let $P_i(t)$ be the position according to $v_i \in V^o$ $(i = 1, 2, \cdots, n)$ and $P_i(t)$ evolves with time, t.

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

where k and μ 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 $\mathbf{F}_i^{ch}(t) = \int_{P\in\partial\Omega} \overrightarrow{P_i(t)P}/(||\overrightarrow{P_i(t)P}||_{L_2})^2 ds$, where $N^1(P_i(t))$ is the 1-neighborhood of $P_i(t)$. V^o is the set of interior vertices of $\mathscr{M}$. Ω 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 $P_i(t)$ to confine $P_i(t)$ within Ω if the initial position of $P_i(t)$ is within Ω. We take $k \gg \mu > 0$ such that the main forces come from the mass-spring model, i.e., this model is spring-dominant. If Ω is convex, μ can be set as 0. When all of the positions $P_i(t)$ are in balance, the spring-dominant energy in (2) reaches its minimal value.

To minimize the spring-dominant energy that evolves with time, explicit or implicit time integration methods are adopted [31, 32, 33]. The implicit time integration methods are stable but time-consuming; the explicit time integration methods are often fast, but not sufficiently robust [33].

For the applications in isogeometric analysis, the reparametrization algorithm developed in this paper is a pre-processing step for solving PDEs to obtain a more efficient and stable process for solving linear systems. However, explicit or implicit time integration methods introduce an additional linear system to solve the force-directed model, and the size of this linear system is positively associated with the number of vertices of parametric meshes. For example, in [32], the size of an involved linear system is $O(n)$, where n is the number of vertices. Thus, these algorithms may introduce another linear system that needs to be solved. To avoid introducing an additional linear system and find a fast solution of the force-directed model (2), we adopt the explicit time integration method and solve this model locally, i.e., the new position of $P_i(t)$ at $t_0 + \Delta t$ is predicted by $P_i(t)$ and the force acting on $P_i(t)$ at t (see Lemma 3.1).

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

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

We can choose a suitable time step Δt and parameters m_i, k, μ. Considering the expected uniformity, m_i is taken as 1. The new position of $P_i(t_0+\Delta t)$ after one iteration can be approximated by $P_i(t_0) + \Delta P_i(t_0)$. This lemma is proved in Section 6.

The algorithm goes as follows: Set parameters ϵ_0, Δt, k, μ and $MaxIterNumber$, where ϵ_0 is the parameter employed to judge whether this algorithm converges or not, Δt is the time step, k and μ are the parameters shown in Equation (2), and $MaxIterNumber$ is the maximum number of iteration of this algorithm.

For a discussion of robustness of this algorithm, please refer to Remark 3.5 in Section 3.3.

Algorithm 1 Local Energy Reducing Algorithm

```
Require: Input the original positions of all of the interior points $P_i$, the parametric mesh $\mathscr{M}$, $\epsilon_0$, $\Delta t$, $k$, $\mu$ and $MaxIterNumber$. 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 $P_i$ 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
    $\epsilon \leftarrow \epsilon / N^o$
  end while
```

In this algorithm, ϵ is a parameter that is used to store the average of the relative displacement of all interior P_i for each iteration. If ϵ is small enough, $\{P_i\}$ should be in a balance state, i.e., if $\epsilon < \epsilon_0$, this algorithm converges. N^o is the number of interior points.

3.2. The smoothing algorithm

Due to Algorithm 1, the position splines' coefficients at the interior vertices have been obtained. In this section, to obtain uniform parametrizations within each element, a method for determining the coefficients of tangent splines at the interior regular vertices is presented. To simplify the process, we assume that all of 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 $\mathscr{M}$ be a parametric mesh and C be 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. Denote the positions of v_1, v_2, v_3, v_4 on the plane of the physical domain as P_1, P_2, P_3, P_4, respectively. By taking,*

(1). $\mathbf{T}^s_{v_1} = \mathbf{T}^s_{v_2} = \overrightarrow{P_1P_2}$, $\mathbf{T}^t_{v_1} = \mathbf{T}^t_{v_4} = \overrightarrow{P_1P_4}$, $\mathbf{T}^t_{v_3} = \mathbf{T}^t_{v_2} = \overrightarrow{P_2P_3}$, $\mathbf{T}^s_{v_3} = \mathbf{T}^s_{v_4} = \overrightarrow{P_4P_3}$, if v_1, v_2, v_3, v_4 are regular vertices of $\mathscr{M}$, and

(2). $\mathbf{T}^s_{v_2} = \overrightarrow{P_1P_2}$, $\mathbf{T}^t_{v_4} = \overrightarrow{P_1P_4}$, $\mathbf{T}^t_{v_3} = \mathbf{T}^t_{v_2} = \overrightarrow{P_2P_3}$, $\mathbf{T}^s_{v_3} = \mathbf{T}^s_{v_4} = \overrightarrow{P_4P_3}$, if a vertex v_1 is an extraordinary vertex of $\mathscr{M}$,

a parametrization $\mathscr{P}$ can be constructed by Equation (1). If P_1, P_2, P_3, P_4 form a convex quadrilateral, $\mathscr{P}$ is injective over C.

Proof. Let $\mathbf{P}_i = \overrightarrow{OP_i}$, where $O(0,0)$ 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 $\mathscr{M}$,

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

where $B_i^{(n)}(s)$ is the i-th Bézier polynomial of degree $(n-1)$, the Bézier coeffients $C = (C_{i,j})_{i,j=0}^{3} =$

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

By derivation,

$$\frac{\partial \mathbf{T}}{\partial s} = \lambda_{1,1}(s,t)(\mathbf{P}_3 - \mathbf{P}_4) + \lambda_{1,2}(s,t)(\mathbf{P}_2 - \mathbf{P}_1), \tag{4}$$

where

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

$$\frac{\partial \mathbf{T}}{\partial t} = \lambda_{2,1}(s,t)(\mathbf{P}_4 - \mathbf{P}_1) + \lambda_{2,2}(s,t)(\mathbf{P}_3 - \mathbf{P}_2), \tag{5}$$

where

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

Thus,

$$\begin{aligned} \frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} =& \lambda_{1,1}\lambda_{2,1}(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_4 - \mathbf{P}_1) + \lambda_{1,1}\lambda_{2,2}(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_2) \\ &+ \lambda_{1,2}\lambda_{2,1}(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1) + \lambda_{1,2}\lambda_{2,2}(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2). \end{aligned} \tag{6}$$

Moreover,

$$\frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} = \det \begin{pmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & 0 \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & 0 \end{pmatrix} = J(s,t)\mathbf{k},$$

where $\mathbf{i}, \mathbf{j}$ are unit vectors along the x-direction and the y-direction on the physical domain, respectively; $\mathbf{k} = \mathbf{i} \times \mathbf{j}$; $J(s,t)$ is the Jacobian of $\mathbf{T}(s,t) = (x(s,t), y(s,t))$.

Because the quadrilateral given by P_1, P_2, P_3, P_4 is convex, $(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_4 - \mathbf{P}_1)$, $(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_2)$, $(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1)$ and $(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2)$ have the same direction. They are along the direction of $\mathbf{k}$ or $-\mathbf{k}$. Considering $\lambda_{i,j}(s,t) \geq 0$ and (6), $J(s,t)$ is positive or negative, if and only if $||\frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t}|| \neq 0$. Thus, we consider the solution of

$$||\frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t}|| = 0. \tag{7}$$

Because $(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_4 - \mathbf{P}_1)$, $(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_2)$, $(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1)$ and $(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2)$ have the same direction and $\lambda_{i,j}(s,t) \geq 0$ in (6), $(s,t) \in [0,1] \times [0,1]$ satisfies (7) if and only if

$$\begin{cases} \lambda_{1,1}(s,t)\lambda_{2,1}(s,t) = 0; \\ \lambda_{1,1}(s,t)\lambda_{2,2}(s,t) = 0; \\ \lambda_{1,2}(s,t)\lambda_{2,1}(s,t) = 0; \\ \lambda_{1,2}(s,t)\lambda_{2,2}(s,t) = 0. \end{cases} \tag{8}$$

Moreover, $\lambda_{1,1}(s,t) = 0$ iff $t = 0$; $\lambda_{1,2}(s,t) = 0$ iff $t = 1$; $\lambda_{2,1}(s,t) = 0$ iff $s = 1$; and $\lambda_{2,2}(s,t) = 0$ iff $s = 0$. Thus, $\forall (s,t) \in [0,1] \times [0,1]$, which does not satisfy (8). i.e., $J(s,t) > 0$ (or < 0) for all $(s,t) \in C$, i.e., $\mathscr{P}|_C(s,t)$ is injective.

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

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

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

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

By derivation,

$$\frac{\partial \mathbf{T}}{\partial s} = \mu_{1,1}(s,t)(\mathbf{P}_3 - \mathbf{P}_4) + \mu_{1,2}(s,t)(\mathbf{P}_2 - \mathbf{P}_1) + \mu_{1,3}(s,t)(\mathbf{P}_3 - \mathbf{P}_1), \tag{9}$$

where,

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

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

where,

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

By (9) and (10),

$$\begin{aligned}\frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t} =& \mu_{1,1}\mu_{2,1}(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_4 - \mathbf{P}_1) + \mu_{1,1}\mu_{2,2}(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_2)\\ &+ \mu_{1,1}\mu_{2,3}(\mathbf{P}_3 - \mathbf{P}_4) \times (\mathbf{P}_3 - \mathbf{P}_1) + \mu_{1,2}\mu_{2,1}(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1)\\ &+ \mu_{1,2}\mu_{2,2}(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2) + \mu_{1,2}\mu_{2,3}(\mathbf{P}_2 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_1)\\ &+ \mu_{1,3}\mu_{2,1}(\mathbf{P}_3 - \mathbf{P}_1) \times (\mathbf{P}_4 - \mathbf{P}_1) + \mu_{1,3}\mu_{2,2}(\mathbf{P}_3 - \mathbf{P}_1) \times (\mathbf{P}_3 - \mathbf{P}_2).\end{aligned} \tag{11}$$

Moreover, because the quadrilateral given by P_1, P_2, P_3, P_4 is convex, the following vectors share the same direction:

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

Thus, similar to **Case 1**, $J(s,t)$ is positive or negative, if and only if $||\frac{\partial \mathbf{T}}{\partial s} \times \frac{\partial \mathbf{T}}{\partial t}|| \neq 0$, i.e., we consider

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

if and only if, in Equation (11),

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

In addition,

$$\begin{aligned}&\mu_{1,1}(s,t) = 0 \text{ iff } t = 0;\ \mu_{1,2}(s,t) = 0 \ \text{ iff } t = 1 \text{ or } s = 0;\\ &\mu_{1,3}(s,t) = 0 \text{ iff } s = 0 \text{ or } s = 1 \text{ or } t = 0 \text{ or } t = 1;\\ &\mu_{2,1}(s,t) = 0 \ \text{ iff } s = 1 \text{ or } t = 0; \mu_{2,2}(s,t) = 0 \ \text{ iff } s = 0;\\ &\mu_{2,3}(s,t) = 0 \text{ iff } s = 0 \text{ or } s = 1 \text{ or } t = 0 \text{ or } t = 1;\end{aligned}$$

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

$$\begin{cases} \mathscr{J}(0,0) = 0; \\ \mathscr{J}(s,t) > 0(\text{or } < 0),\ (s,t) \neq (0,0) \end{cases} \tag{13}$$

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

For Theorem 3.2, there is a remark:

Remark 3.3.

1. *If an initial layout of the undirected graph consists of non-overlapping convex quadrilaterals, this convex structure is maintained by the spring-dominant model* (2) *because a non-convex structure is not balanced. We can choose a suitable initial parametrization such that its layout of the undirected graph consists of non-overlapping convex quadrilaterals. For the details of the existence of satisfied initial parametrizations, please refer to Remark 3.5.*
2. *Theorem 3.2 provides a method to set the coefficients of the tangent splines such that the resulting parametrization is as uniform as possible within an element. For example, in **Case 1**, if a quadrilateral formed by P_1, P_2, P_3, P_4 is a parallelogram, $\frac{\partial \mathbf{T}}{\partial s}$ and $\frac{\partial \mathbf{T}}{\partial t}$ are constants by Equations* (4) *and* (5)*, i.e., $\mathscr{P}|_C(s,t)$ is uniform. Especially, in the h-refinement process, the elements' sizes on the physical domain become smaller and most of the cells satisfy the condition of **Case 1**. By Algorithm 1, the vertices of its undirected graph distribute as uniformly as possible on the physical domain. Thus, quadrilaterals can be approximated by parallelograms when the size of elements is small and uniform enough. In Example 3.2, there is a numerical experiment about the uniformity of H^1-parametrizations by the reparametrization algorithm developed in this paper.*

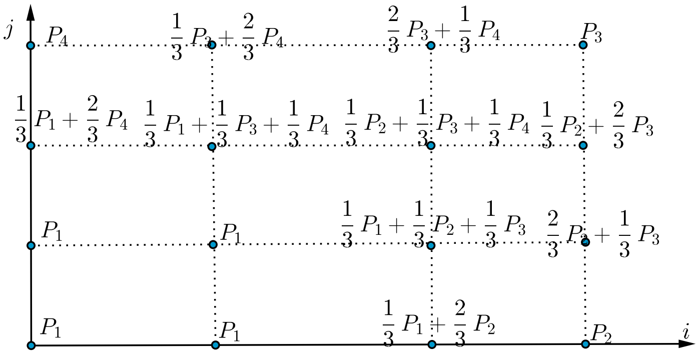

j
P_4
$\frac{1}{3}P_3 + \frac{2}{3}P_4$
$\frac{2}{3}P_3 + \frac{1}{3}P_4$
P_3
$\frac{1}{3}P_1 + \frac{2}{3}P_4$
$\frac{1}{3}P_1 + \frac{1}{3}P_3 + \frac{1}{3}P_4$
$\frac{1}{3}P_2 + \frac{1}{3}P_3 + \frac{1}{3}P_4$
$\frac{1}{3}P_2 + \frac{2}{3}P_3$
P_1
P_1
$\frac{1}{3}P_1 + \frac{1}{3}P_2 + \frac{1}{3}P_3$
$\frac{2}{3}P_2 + \frac{1}{3}P_3$
P_1
P_1
$\frac{1}{3}P_1 + \frac{2}{3}P_2$
P_2
i

Figure 8: Control Points $\mathbf{P}_{(i,j)}$ in **Case 2**

Theorem 3.4. *As mentioned in Theorem 3.2, $\mathscr{P}|_C$ satisfies the H^1-integrability assumption, i.e., by this parametrization the test functions are H^1 on $\mathscr{P}(C)$.*

Proof. For **Case 1**, based on $J(s,t) > 0$ (or < 0) and the continuity of $\mathscr{P}|_C(s,t)$, there exists a C^1 inverse map of $\mathscr{P}|_C(s,t)$ locally. Thus, the test functions on the physical domain are C^1 ($\subset H^1$). In the following, a detailed proof of **Case 2** is presented.

By the Jacobian of $\mathscr{P}|_C$ in (13), there is only one isolated singularity at P_1. Based on the Bézier coefficients C in **Case 2**, the control points $\{\mathbf{P}_{(i,j)}\}_{i,j=0}^3$ are shown in Figure 8.

In Assumption 5.1 of [17], the degree is $\mathbf{p} = (3,3)$ and we take $\alpha = (2,2)$, i.e., $\alpha_1 = 2, \alpha_2 = 2$. The labels are shown in Figure 9. The control points $\{\mathbf{P}_{(i,j)}\}_{i,j=0}^3$ satisfy the following:

1. The control points $\mathbf{P}_{(i,j)} = P_1$ when $(i,j) \in \mathbb{D}_\alpha$, and the other control points $\mathbf{P}_{(i,j)} \neq P_1$. Here, $\mathbb{D}_\alpha = \{(i,j) : 0 \le i \le \alpha_1 - 1, 0 \le j \le \alpha_2 - 1\}$;
2. The triangles $\triangle(\mathbf{P}_{(0,0)}, \mathbf{P}_{(\alpha_1,0)}, \mathbf{P}_{(\alpha_1,1)})$, $\triangle(\mathbf{P}_{(0,0)}, \mathbf{P}_{(\alpha_1,0)}, \mathbf{P}_{(0,\alpha_2)})$, and $\triangle(\mathbf{P}_{(0,0)}, \mathbf{P}_{(0,\alpha_2)}, \mathbf{P}_{(1,\alpha_2)})$ do not degenerate because the quadrilateral given by P_1, P_2, P_3, P_4 is convex.

Thus, $\mathscr{P}|_C$ satisfies H^1-integrability assumption 5.1 outlined in [17], i.e., by this parametrization, the test functions are H^1 on $\mathscr{P}(C)$.

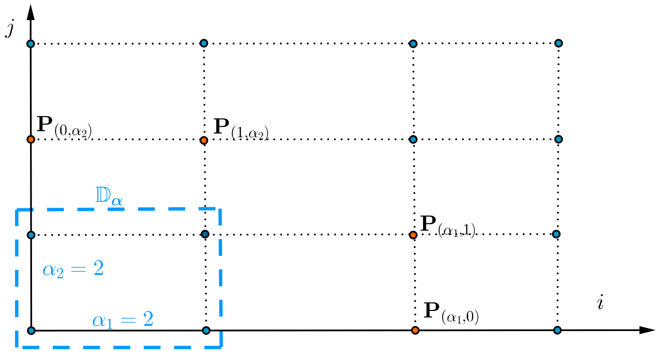

j
$\mathbf{P}_{(0,\alpha_2)}$
$\mathbf{P}_{(1,\alpha_2)}$
$\mathbb{D}_\alpha$
$\mathbf{P}_{(\alpha_1,1)}$
$\alpha_2 = 2$
$\alpha_1 = 2$
$\mathbf{P}_{(\alpha_1,0)}$
i

Figure 9: Labels in Assumption 5.1 of [17]

□

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 $\mathscr{M}$ and assume that its position has been determined by Algorithm 1. There are four cells that take v_i as their corner vertex because 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 $P_i = \mathscr{P}(v_i)$ and $N^1(P_i) = \{P_i^1, P_i^2, P_i^3, P_i^4\}$ shown in Figure 10, and 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.,

$$\mathbf{T}_{v_i}^s = 1/2\overrightarrow{P_iP_i^2} + 1/2\overrightarrow{P_i^4P_i};\ \mathbf{T}_{v_i}^t = 1/2\overrightarrow{P_iP_i^1} + 1/2\overrightarrow{P_i^3P_i}. \tag{14}$$

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 $P_iP_i^2$ and the direction of the t-curve at v is along $P_iP_i^1$, then,

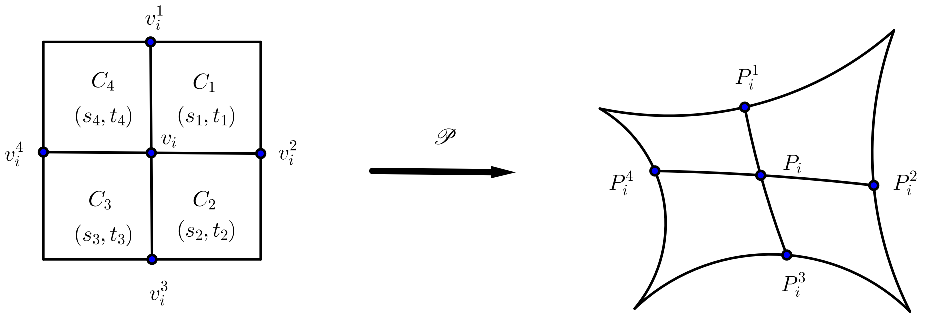


Figure 10: The interior regular vertex P_i and $N^1(P_i) = \{P_i^1, P_i^2, P_i^3, P_i^4\}$

$$\mathbf{T}^s_{v_i} = \frac{1}{2\ell_1}\overrightarrow{P_i^4 P_i} + \frac{1}{2\ell_2}\overrightarrow{P_i P_i^2};\ \mathbf{T}^t_{v_i} = \frac{1}{2\ell_3}\overrightarrow{P_i^3 P_i} + \frac{1}{2\ell_4}\overrightarrow{P_i P_i^1}, \tag{15}$$

where ℓ_1 is the distance between v_i^4 and v_i considered in C_4, ℓ_2 is the distance between v_i and v_i^2 considered in C_1, ℓ_3 is the distance between v_i^3 and v_i considered in C_2, and ℓ_4 is the distance between v_i and v_i^1 considered in C_1. Select similarly $\mathbf{T}^s_{v_i}, \mathbf{T}^t_{v_i}$ in the other cases of the local frame at v_i. In the following, we use Example 3.1 to explain (15).

Example 3.1. *We consider a parametric mesh obtained by uniformly subdividing $C = [0, 1] \times [0, 1]$ in Theorem 3.2 into 4 subcells. Using the labels in Figure 10, C_1, C_2, C_3, C_4 have the same local frame and $v_i(1/2, 1/2), v_i^1(1/2, 1), v_i^2(1, 1/2), v_i^3(1/2, 0), v_i^4(0, 1/2)$. Thus, $\ell_1 = \ell_2 = \ell_3 = \ell_4 = 1/2$ and*

$$\begin{aligned}
&P_i = \mathscr{P}(v_i) = 1/4(P_1 + P_2 + P_3 + P_4);\\
&P_i^1 = \mathscr{P}(v_i^1) = 1/2(P_3 + P_4);\ P_i^3 = \mathscr{P}(v_i^3) = 1/2(P_1 + P_2);\\
&P_i^2 = \mathscr{P}(v_i^2) = 1/2(P_2 + P_3);\ P_i^4 = \mathscr{P}(v_i^4) = 1/2(P_1 + P_4);
\end{aligned}$$

where $\mathscr{P}(s, t)$ maps C to the quadrilateral formed by P_1, P_2, P_3, P_4 defined in **Case 1** *of Theorem 3.2. By* (15)*,*

$$\mathbf{T}^s_{v_i} = 1/2(P_2 - P_1) + 1/2(P_3 - P_4);\ \mathbf{T}^t_{v_i} = 1/2(P_4 - P_1) + 1/2(P_3 - P_2).$$

By Theorems 3.2 and 3.4 and Equation (15), we can re-set the coefficients of the tangent splines such that the resulting parametrizations are as uniform as possible in each element.

3.3. Reparametrization algorithm

Base on Sections 3.1 and 3.2, we present the complete reparametrization algorithm with fixed initial boundaries and analyze its initial parametrization as well as the uniformity of H^1-parametrizations.

Algorithm 2 The Reparametrization Algorithm

```
Require: Input the original positions of all the interior points $P_i$, the parametric
  mesh $\mathscr{M}$, the initial parametrization $\mathscr{P}_0$, and the parameters $\epsilon_0$, $\Delta t$, $k$, $\mu$ and
  $MaxIterNumber$ in Algorithm 1.
Ensure:
  % Distribute elements on a given physical domain as uniformly as possible:
  Input the parameters and compute the balance positions of interior points, $\{P_i^b\}$, by
  Algorithm 1.
  $\{P_i\} \leftarrow \{P_i^b\}$.
  % Make the resulting parametrization as uniformly as possible in each element:
  for $\{P_i\}$ do
    if $v_i$ is a regular vertex, where then
      Compute $P_i$'s 1-neighbourhood $N^1(P_i)$ based on the topology of $\mathscr{M}$
      Determine $\mathbf{T}_{v_i}^s$ and $\mathbf{T}_{v_i}^t$ by Equations (15) in Section 3.2, where $v_i$ is the vertex
      corresponding to $P_i$.
    end if
  end for
  % Restore the given boundaries:
  for the boundary vertex $v_j^b$ of $\mathscr{M}$ do
    The coefficients of splines defined at $v_j^b$ inherit from $\mathscr{P}_0$.
  end for
```

In Sections 3.1 and 3.2, there are some requirements regarding the initial parametrization of the reparametrization algorithm (Algorithm 2). The following remark addresses the choice of initial parametrizations.

Remark 3.5. *In Algorithm 1, we adopt the explicit time integration method and solve the model* (2) *locally to avoid introducing another linear system and to find a solution quickly. Besides all the initial vertex positions should be within the physical domain, an initial parametrization whose vertices are not far from their balance positions can enhance the robustness of Algorithm 1. In other words, we try to obtain a uniform initial parametrization. Moreover, in Section 3.2, an initial parametrization is expected to satisfy that its layout of the undirected graph consists of non-overlapping convex quadrilaterals. Based on the applications noted in the introduction, there are two ways to construct initial parametrizations:*

1. Construct an initial parametrization from given boundaries. Denote the physical domain bounded by boundaries as Ω. We construct an initial parametrization by following these steps:

(a). Choose a polygon $\Omega_p \subset \Omega$ to approximate Ω, where the boundary vertices of Ω_p are on the boundaries of Ω.

(b). Triangulate of Ω_p as uniformly as possible, perhaps using Delaunay triangulations [34, 35]. It helps to obtain a uniform distribution of the layout of the resulting graph.

(c). For each triangle, connect its barycenter with the midpoints of its sides. A non-overlapping convex quadrilateral decomposition of Ω_p is obtained.

(d). For a boundary vertex of this decomposition of Ω_p, its coefficients are determined by describing these given boundaries of Ω by its splines, such as fitting the boundaries; for an interior point of this decomposition, its coefficient of the position splines is determined by its current position, and its coefficients of the tangent splines are 0 *if there are tangent splines.*

Then, the initial parametrization that meets the requirements is obtained.

2. An initial parametrization is a H^1-parametrization defined on a coarse parametric mesh. This case exists in the h-refinement process of isogeometric analysis. After representing this H^1-parametrization on the finer parametric mesh, it is a good initial parametrization because H^1-parametrizations are uniform within elements. For example, after subdividing the parametric mesh in Example 3.1, $\mathbf{F}^{sp}_{P_i} = \mathbf{0}$, which means P_i is near to its balance position on the finer mesh.

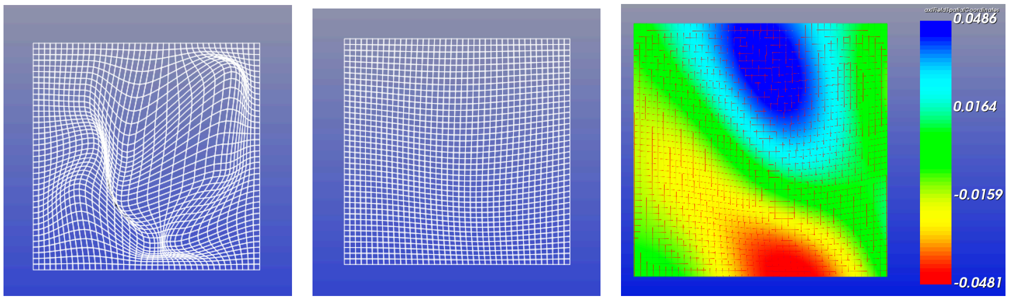
To test the uniformity of H^1-parametrizations, in Example 3.2, we compare the H^1-parametrization with the standard uniform parametrization of a rectangle.

Example 3.2. *Suppose the physical domain $\Omega = [0,2] \times [0,2]$ and the parametric domain $\Sigma = [0,4] \times [0,4]$. The standard uniform parametrization is expected as $\mathscr{Q}_0(s,t) = (0.5s, 0.5t)$. Its Jacobian $J_{\mathscr{Q}_0}(s,t) = 0.25$.*

We input an initial parametrization $\mathscr{P}_0$ shown in Figure 11(a) that has the same boundary description as $\mathscr{Q}_0$. The parametric mesh is a tensor-product mesh with 1681 vertices. Take the parameters $\varepsilon_0 = 0.00001, \Delta t = 0.1, k = 1, \mu = 0$. Then, by the reparametrization algorithm (Algorithm 2), we obtain the H^1-parametrization $\mathscr{P}_1$ shown in Figure 11(b). In Figure 11(c), we compare the Jacobian of $\mathscr{P}_1$ with $J_{\mathscr{Q}_0}$. There is no too much difference between the H^1-parametrization and the standard uniform parametrization, i.e., the H^1-parametrization shown in Figure 11(b) is uniform.

4. Numerical experiments

In this section, experiments are presented. As we state in the introduction, based on the results of the condition numbers' bounds in isogeometric analysis and the classical FEM, we expect a uniform parametrization to obtain a smaller condition number that



(a) $\mathscr{P}_0$ (b) $\mathscr{P}_1$ (c) $J_{\mathscr{P}_1} - J_{\mathscr{Q}_0}$

Figure 11: The H^1-parametrization of the rectangle $[0,2] \times [0,2]$ by the reparametrization algorithm (Algorithm 2)

brings efficiency and numerical stability. Considering to solve the elliptic boundary problem (16) in isogeometric analysis framework, we compare their condition numbers under parametrizations that have different uniformity, where basis functions are chosen as Hermite basis functions of the spline space over $\mathscr{M}$ defined in [20]. In the following experiments, we find that the condition numbers become smaller when the parametrization becomes more uniform.

$$\begin{aligned} -\Delta u &= f; \\ u|_{\partial\Omega} &= 0. \end{aligned} \tag{16}$$

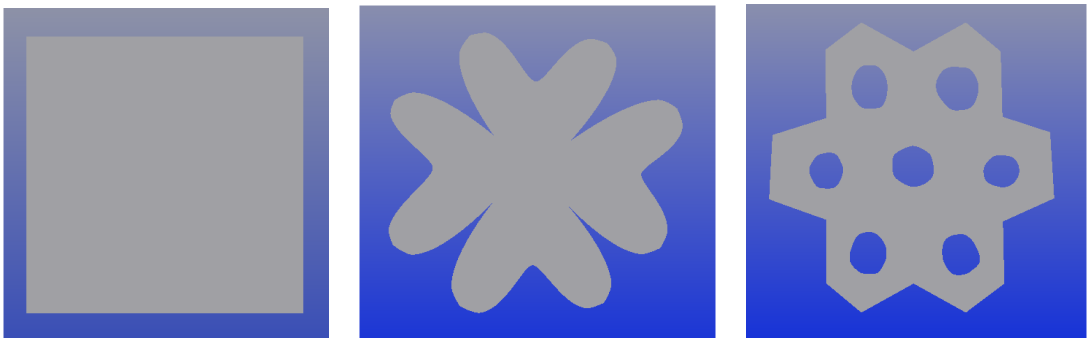
The first experiment is solving the elliptic boundary problem (16) on a square domain shown in Figure 12(a). Here we compare the condition numbers with different parametrizations. These parametrizations include a standard uniform parametrization, a parametrization with singularities and a parametrization without singularities. The second experiment is solving (16) on the four leaf clover-shape physical domain, as shown in Figure 12(b). This physical domain has complex geometric features: sharp corners and non-convex shapes. The third experiment is solving (16) on the porous physical domain illustrated in Figure 12(c), which has complex topological features.

4.1. A square domain

In this experiment, we compare the condition numbers by different parametrizations of a square domain. In order to make a comparison of test results, parametric meshes of these parametrizations share the same number of cells, i.e., the numbers of elements on the square domain are the same, where the numbers of elements on the square domain are chosen as 288, 800, 1568, 2592 respectively. For instance, in Figures 13, 14(a), 14(b), 15(a) and 15(b), there are 288 elements on the square domain by these parametrizations.

In Figures 16 and 17, the condition numbers by $\mathscr{P}^i(s,t)$ and its H^1-parametrization $\mathscr{P}^i_{H^1}(s,t)$ are presented. By the results in Figures 16 and 17,

$$Cond(\mathscr{P}^i_{H^1}(s,t)) < Cond(\mathscr{P}^i(s,t)),$$



(a) A square domain

(b) A four leaf clover-like physical domain

(c) A perforated model structure physical domain

Figure 12: The physical domains considered in Section 4

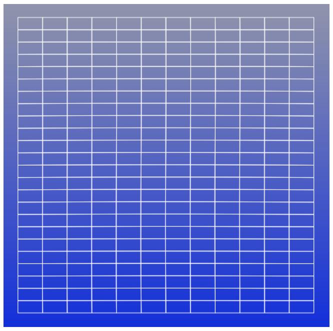
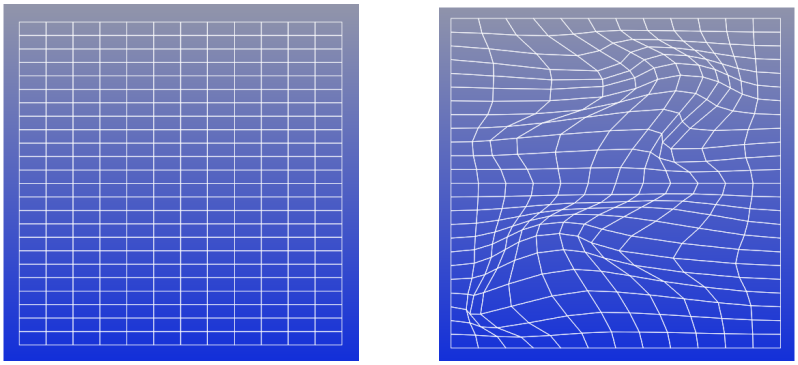


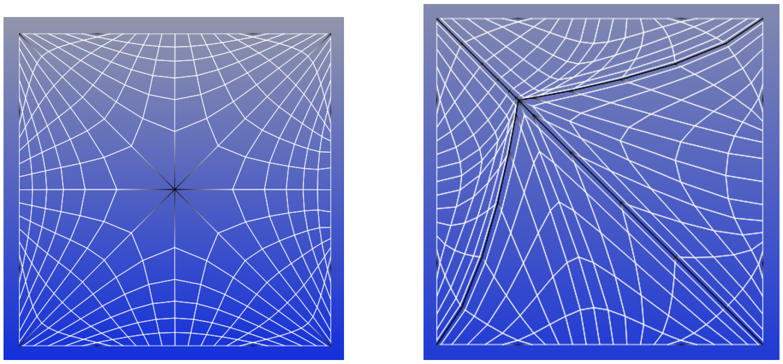
Figure 13: A standard uniform parametrization $I(s,t)$



(a) $\mathscr{P}^1_{H^1}(s,t)$

(b) $\mathscr{P}^1(s,t)$

Figure 14: The H^1-Parametrization $\mathscr{P}^1_{H^1}(s,t)$ and its initial parametrization $\mathscr{P}^1(s,t)$



(a) $\mathscr{P}^2_{H^1}(s,t)$ (b) $\mathscr{P}^2(s,t)$

Figure 15: The H^1-Parametrization $\mathscr{P}^2_{H^1}(s,t)$ and its initial parametrization$\mathscr{P}^2(s,t)$

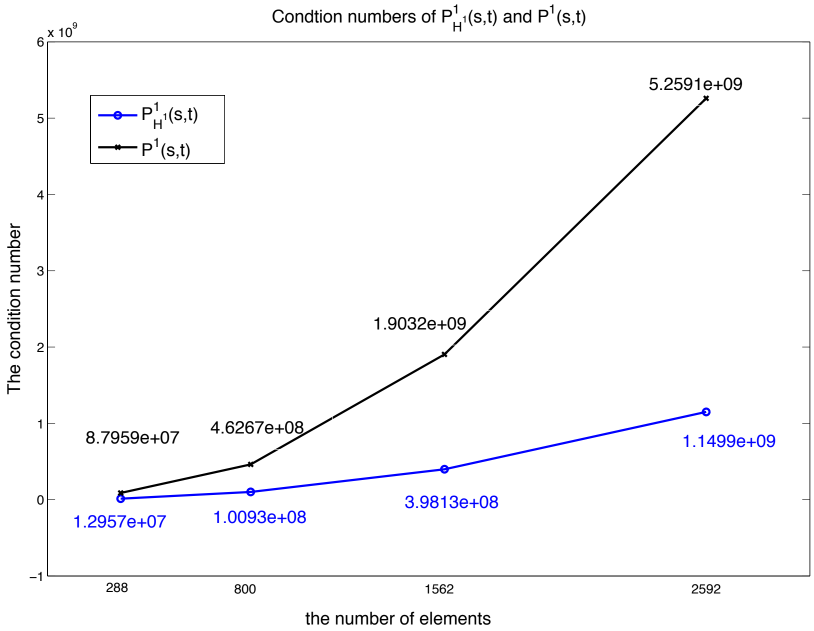


Figure 16: The condition numbers by $\mathscr{P}^1(s,t)$ and $\mathscr{P}^1_{H^1}(s,t)$

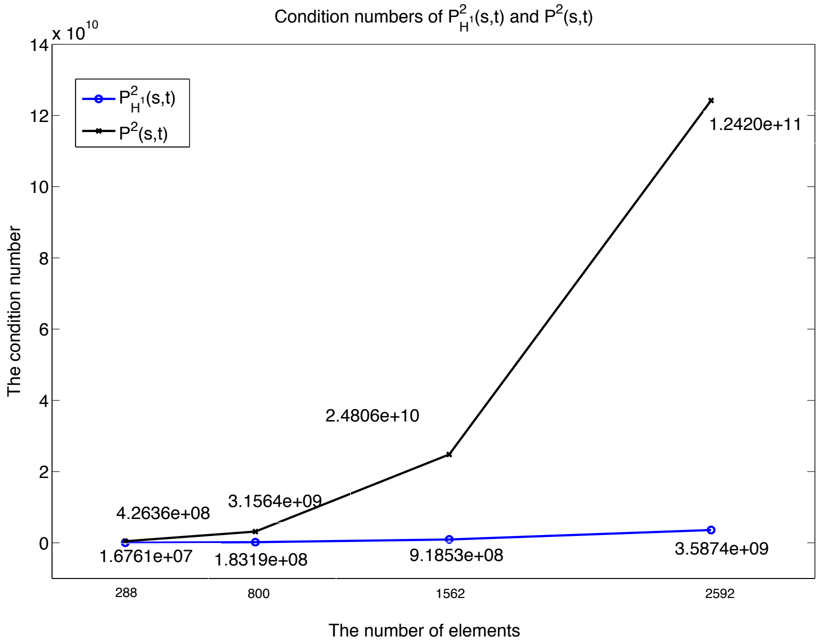


Figure 17: The condition numbers by $\mathscr{P}^2(s,t)$ and $\mathscr{P}^2_{H^1}(s,t)$

where $i = 1, 2$; $Cond(\mathscr{P}(s,t))$ is the condition number by $\mathscr{P}(s,t)$.

In Figure 18, for the same physical domain (the square domain), we compare the condition numbers by $\mathscr{P}^1_{H^1}(s,t)$, $\mathscr{P}^2_{H^1}(s,t)$ and the standard uniform parametrization $I(s,t)$. $\mathscr{P}^1_{H^1}(s,t)$ and $I(s,t)$ share the same structures of parametric meshes. In Figure 18,

$$Cond(\mathscr{P}^1_{H^1}(s,t)) \approx Cond(I(s,t)).$$

Moreover, for $\mathscr{P}^2_{H^1}(s,t)$ and $I(s,t)$, they are share different the structures of parametric meshes. In Figure 18,

$$Cond(I(s,t)) < Cond(\mathscr{P}^2_{H^1}).$$

4.2. A four leaf clover-like physical domain

In this experiment, a four leaf clover-like physical domain is taken as a physical domain. There are two initial parametrizations $\mathscr{P}^1$ and $\mathscr{P}^2$ shown in Figures 19(a) and 20(a). We modify them by Algorithm 2 and obtain H^1-parametrizations $\mathscr{P}^1_{H^1}$ and $\mathscr{P}^2_{H^1}$ shown in Figures 19(b) and 20(b) respectively, where the parameters are taken as $\epsilon_0 = 0.001$, $\Delta t = 0.1$, $k = 0.4$ and $\mu = 0.01$. By computing the condition numbers of the stiffness matrices of the elliptic boundary problem,

$$Cond(\mathscr{P}^1_{H^1}(s,t)) = 1.5404 \times 10^7 < Cond(\mathscr{P}^1(s,t)) = 1.1476 \times 10^9$$
$$Cond(\mathscr{P}^2_{H^1}(s,t)) = 1.2530 \times 10^7 < Cond(\mathscr{P}^2(s,t)) = 1.5692 \times 10^7.$$

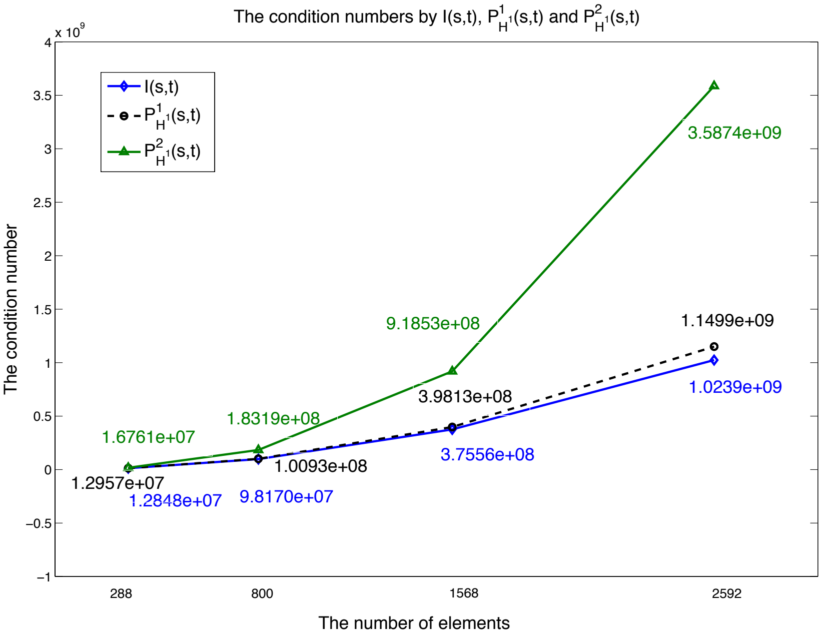
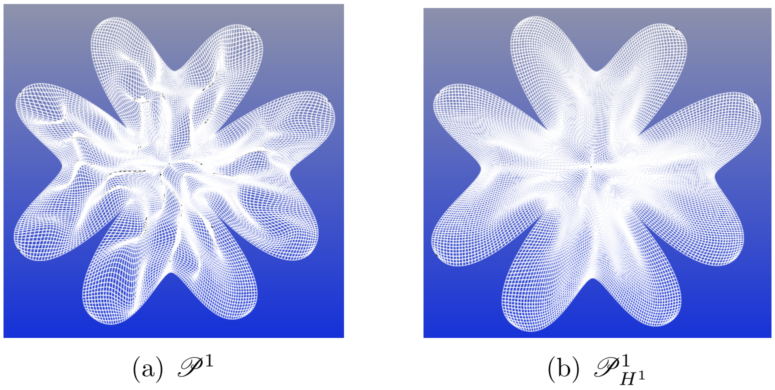


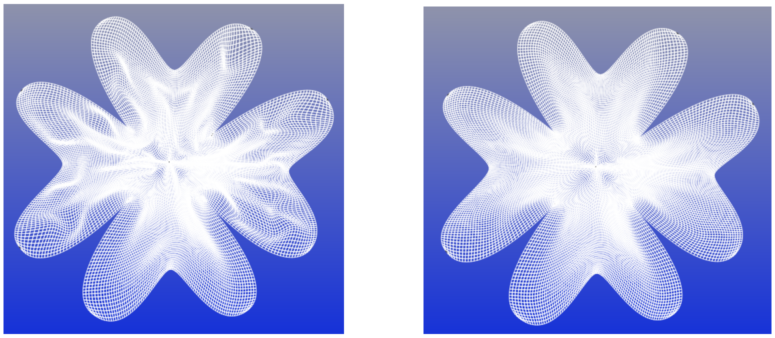
Figure 18: The condition numbers by $\mathscr{P}^1_{H^1}(s,t)$, $\mathscr{P}^2_{H^1}(s,t)$ and $I(s,t)$



(a) $\mathscr{P}^1$

(b) $\mathscr{P}^1_{H^1}$

Figure 19: $\mathscr{P}^1$ and $\mathscr{P}^1_{H^1}$



(a) $\mathscr{P}^2$ (b) $\mathscr{P}^2_{H^1}$

Figure 20: $\mathscr{P}^2(s,t)$ and $\mathscr{P}^2_{H^1}(s,t)$

Furthermore, by Algorithm 2, $\mathscr{P}^2$ converges to the H^1-parametrization $\mathscr{P}^2_{H^1}$ shown in Figure 20(b) after 107 iterations in Algorithm 1. In the convergence process, the positions of the vertices become more and more uniform and the convex decomposition is maintained. In Figures 21(a) to 21(d), we collect the vertices' positions in this process. We construct parametrizations by these positions (in Figures 21(a) to 21(d)) and by Equation (15) in Section 3.2. These parametrizations are denoted as $\mathscr{P}^2_a, \mathscr{P}^2_b, \mathscr{P}^2_c, \mathscr{P}^2_d$. Computing the condition numbers of the stiffness matrices of the elliptic boundary problem directly, we obtain $Cond(\mathscr{P}^2_a) = 1.5692 \times 10^7$, $Cond(\mathscr{P}^2_b) = 1.3926 \times 10^7$, $Cond(\mathscr{P}^2_c) = 1.3069 \times 10^7$ and $Cond(\mathscr{P}^2_d) = 1.2530 \times 10^7$, i. e.,

$$Cond(\mathscr{P}^2_a) > Cond(\mathscr{P}^2_b) > Cond(\mathscr{P}^2_c) > Cond(\mathscr{P}^2_d)$$

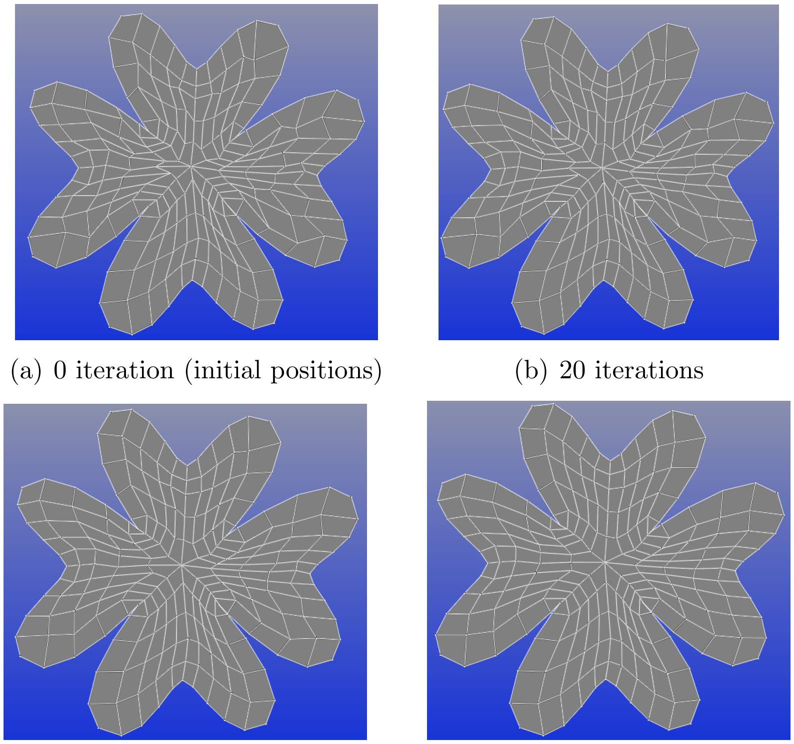
The condition number becomes smaller and smaller when the parametrization becomes more and more uniform.

4.3. A porous physical domain

We consider a porous structure as a physical domain that has a complex topological structure.

The initial parametrization $\mathscr{P}$ is given in Figure 22(a). The layout of the position of vertices of $\mathscr{P}$ in Figure 23(a) under this initial parametrization is a convex decomposition of the physical domain. By the reparametrization algorithm (Algorithm 2), $\mathscr{P}$ converges to the H^1-parametrization $\mathscr{P}_{H^1}$ shown in Figure 22(b) after 137 iterations in Algorithm 1, where $\epsilon_0 = 0.001$, $\Delta t = 0.1$, $k = 1$ and $\mu = 0.001$.

In the convergence process, the positions of vertices become more and more uniform and the convex decomposition is maintained. In Figures 23(a) to 23(d), we collect the vertices' positions in this process. We construct parametrizations by these positions (in Figures 23(a) to 23(d)) and by Equation (15) in Section 3.2. These parametrizations are denoted as $\mathscr{P}_a, \mathscr{P}_b, \mathscr{P}_c, \mathscr{P}_d$. Computing the condition numbers of the stiffness matrices of the elliptic boundary problem with Hermite basis functions directly, we obtain $Cond(\mathscr{P}_a) = 8.9074 \times 10^6$, $Cond(\mathscr{P}_b) = 5.7810 \times 10^6$, $Cond(\mathscr{P}_c) = 4.2948 \times$



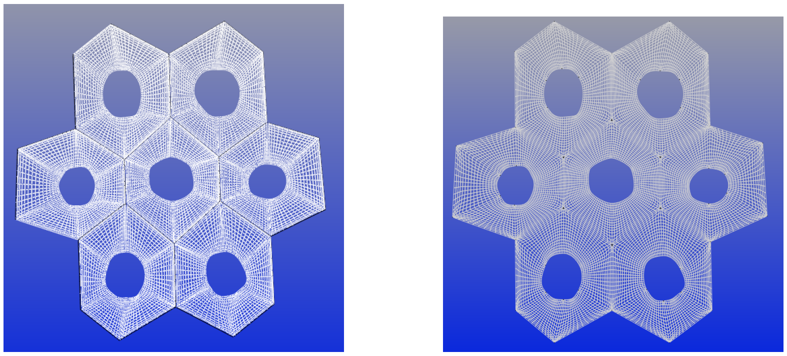
(a) 0 iteration (initial positions)

(b) 20 iterations

(c) 60 iterations

(d) 107 iterations

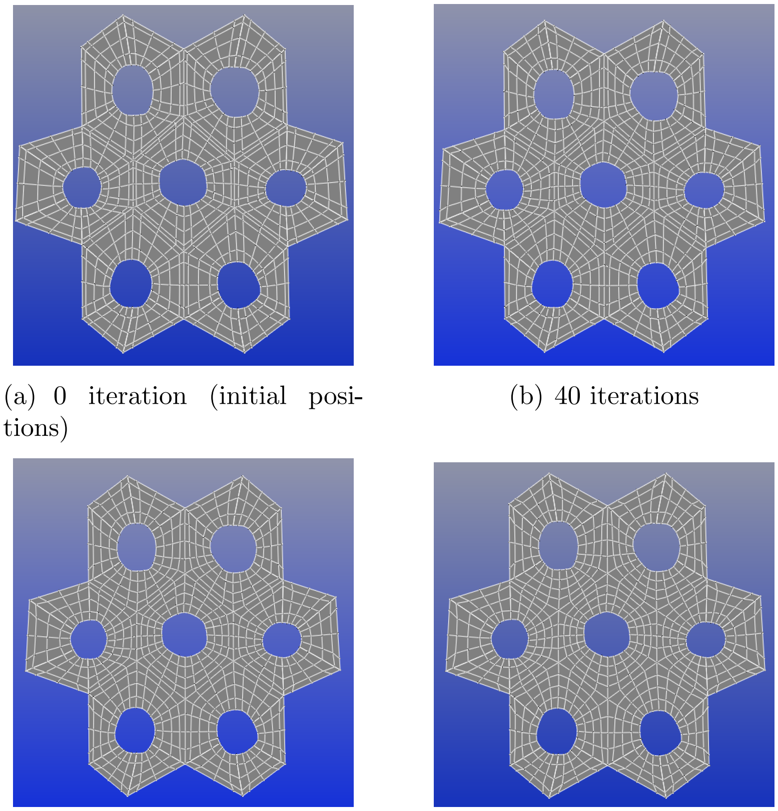
Figure 21: The positions of vertices in the convergence progress of Algorithm 1



(a) The initial parametrization $\mathscr{P}$

(b) The H^1- parametrization $\mathscr{P}_{H^1}$

Figure 22: The initial parametrization $\mathscr{P}$ and the H^1-parametrization $\mathscr{P}_{H^1}$



(a) 0 iteration (initial positions)

(b) 40 iterations

(c) 80 iterations

(d) 132 iterations

Figure 23: The positions of vertices in the convergence progress of Algorithm 1

10^6, $Cond(\mathscr{P}_d) = 3.4927 \times 10^6$, i.e.,

$$Cond(\mathscr{P}_a) > Cond(\mathscr{P}_b) > Cond(\mathscr{P}_c) > Cond(\mathscr{P}_d).$$

The condition number becomes smaller and smaller when the parametrization becomes more and more uniform.

5. Conclusions and future work

This paper presented a reparametrization method with fixed boundaries over complex planar physical domains. Bi-cubic splines with rigid transition maps in [20] are used to represent the complex physical domains globally. The design of the parametrization algorithm is based on the three basic requirements, i.e., injective parametrization, efficiency and numerical stability, and the regularity property of test functions. By the reparametrization algorithm, a global uniform injective H^1-parametrization is generated. By this reparametrization, an efficient and stable solution process can be expected.

Within isogeometric analysis, for a better future solution of PDEs over a complex physical domain, besides the parametrization algorithm shown in this paper, an algorithm should be developed to obtain a better boundary representation for a better approximation of the original physical domains.

6. Appendix

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

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

Suppose that $C = [0,1] \times [0,1]$ be a cell of $\mathscr{M}$ 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 $\mathscr{P}_p$ in [20], then, over C, $\mathscr{P}_p$ is a bicubic polynomial with vector coefficients, i. e.,

$$\mathscr{P}_p|_C(s,t) = \overrightarrow{OP_1} h_0(s)h_0(t) + \overrightarrow{OP_2} h_1(s)h_0(t) + \overrightarrow{OP_3} h_1(s)h_1(t) + \overrightarrow{OP_4} h_0(s)h_1(t), \quad (17)$$

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

Now we can prove Lemma 2.2:

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

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

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,

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

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

□

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,

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

□

Acknowledgments

The authors are grateful to the anonymous reviewers for their useful comments and suggestions. The authors are supported by the NSF of China (No. 11601114, No. 11526069), the Anhui Provincial Natural Science Foundation (No. 1608085QA14) and the Open Project Program of the State Key Lab of CAD&CG (Grant No. A1601), Zhejiang University.

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. "parametrization 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 parametrization 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 parametrization of computational domain from CAD boundary by variational harmonic method". *Journal of Computational Physics.* vol. 252, p. 275-289, 2013.

[8] E. Pilgerstorfer, B. Jüttler . "Bounding the influence of domain parametrization 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] W.G. Strang, G. Fix. "An Analysis of the Finite Element Method". *Prentice-Hall, New Jersey*, 1973.

[11] S.C. Brenner, R.L. Scott. "The Mathematical Theory of Finite Element Methods". *Springer*, 2005.

[12] J. Shewchuk. "What is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy and Quality Measures". *Tech. report, Department of Computer Science, University of California, Berkeley, CA,* 2003.

[13] Q. Du, D. Wang, L. Zhu. "On mesh geometry and stiffness matrix conditioning for general finite element spaces". *SIAM Journal on Numerical Analysis*, vol. 47, p. 1421-1444, 2009.

[14] E. Barragy, G. Carey. "Preconditioners for high degree elements". *Comput. Methods Appl. Mech. Engrg.*, vol. 93, p. 97-110, 1991.

[15] G. Carey, E. Barragy, "Basis function selection and precondition high degree finite element and spectral methods". *BIT Numerical Mathematics*, vol. 29, p. 794-804, 1989.

[16] N. Hu, X. Guo, I. Katz. "Bounds for eigenvalues and condition number in the p-version of the finite element methods". *Math. Comp.*, vol. 67, p. 1423-1450, 1998.

[17] 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.

[18] T. Takacs, B. Jüttler. "H^2 regularity properties of singular parameterzations in isogeometric analysis". *Graphical Models*, vol. 74, p. 361–372, 2012.

[19] H. Kim, Y. Seo, and S. 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.

[20] M. Wu, B. Mourrain, A. Galligo, B. Nkonga. "Hermite type Spline spaces over rectangular meshes with arbitrary topology". *accepted by Communications in Computational Physics.* https://hal.inria.fr/hal-01196428v3/document, version 3, 2016

[21] U. Reif. "A refineable space of smooth spline surfaces of arbitrary topological genus". *Journal of Approximation Theory*, vol. 90, p. 174-199, 1997.

[22] T. Kamada, S. Kawai. “An algorithm for drawing general undirected graphs”. *Information Processing Letters.* vol. 31, p. 7-15, 1989.

[23] B. Becker, G. Hotz. “On the optimal layout of planar graphs with fixed boundary”. *SIAM J. Comput.* vol. 16, p. 946-972, 1987.

[24] O. Etzmuss, J. Gross, W. Strasser. “Deriving a particle system from continuum mechanics for the animation of deformable objects”. *IEEE Trans. Vis. Comput. Graph*, vol. 9, p. 538-550, 2003.

[25] D. Terzopoulos, K. Fleischer. “Deformable models”. *The Visual Computer*, vol. 4, p. 306-331, 1988.

[26] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson. “Physically Based Deformable Models in Computer Graphics”. *Computer Graphics Forum*, vol. 25, p. 809-836, 2006.

[27] R. Bridson, S. Marino, R. Fedkiw. “Simulation of clothing with folds and wrinkles”. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, ISBN: 1-58113-659-5*, p. 28-36, 2003.

[28] A. Selle, M. Lentine, R. Fedkiw. “A Mass Spring Model for Hair Simulation”. *ACM Trans. Graph.*, vol. 27, p. 64:1-64:11, 2008.

[29] P. Eades, X. Lin. “Spring algorithms and symmetry”. *Theoretical Computer Science*, vol. 240, p. 379-405, 2000.

[30] S. Kobourov. “Spring Embedders and Force Directed Graph Drawing Algorithms”. $http://www.cs.arizona.edu/\ kobourov/fd\_survey.pdf$, 2012.

[31] A. Stern, E. Grinspun. “Implicit-explicit variational integration of highly oscillatory problems”. *Multiscale Modeling* & *Simulation*, vol. 7, p. 1779-1794, 2009.

[32] T. Liu, A. Bargteil, J. O’Brien, L. Kavan. “Fast Simulation of Mass-Spring Systems”. *ACM Transactions on Graphics*, vol. 33, p. 214:1–214:7, 2013.

[33] B. Eberhardt, O. Etzmuβ, M. Hauth. “Implicit-Explicit Schemes for Fast Animation with Particle Systems”. *Computer Animation and Simulation 2000, Proceedings of the Eurographics Workshop in Interlaken, Springer Vienna*, p. 137-151, 2000.

[34] B. Joe. “Delaunay Triangular Meshes in Convex Polygons”. *SIAM J. Sci. and Stat. Comput.*, vol. 7, p. 514-539, 1984.

[35] J. Shewchuk. “Delaunay refinement algorithms for triangular mesh generation”. *Computational Geometry*, vol. 22, p. 21-74, 2002.

[36] R. Bank, R. Smith. “Mesh smoothing using a posteriori error estimates”. *SIAM J. Numer. Anal.*, vol. 34, p. 979-997, 1997.

[37] P. Knupp. “Matrix norms and the condition number: A general framework to improve mesh quality via node-movement”. *in Proceedings of the 8th International Meshing Roundtable (Lake Tahoe, CA)*, p. 13-22, 1999.

[38] A. Liu, B. Joe. “Relationship between tetrahedron shape measures”. *BIT Numerical Mathematics*, vol. 34, p. 268-287, 1994.