INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.3189

Parameterization of planar curves immersed in triangulations with application to finite elements

Ramsharan Rangarajan and Adrian J. Lew*, †

Department of Mechanical Engineering, Stanford University, Stanford, CA 94305-4040, U.S.A.

SUMMARY

We construct a method for the parameterization of a class of planar piecewise C^2-curves over a collection of edges in an ambient triangulation. The map from the collection of edges to the curve is the closest-point projection. A distinguishing feature of the method is that edges in the ambient triangulation need not interpolate the curve. We formulate conditions on the ambient triangulations so that the resulting parameterization over its selected edges is (i) bijective, (ii) maps simple, connected collection of edges to simple, connected components of the curve, and (iii) is C^1 within each edge of the collection. These properties of the parameterization make it particularly useful in the construction of high-order finite element approximation spaces on planar curves immersed in triangulations. We discuss this application and illustrate it with numerical examples.

The parameterization method applies to a large class of planar curves, including most ones of interest in engineering and computer graphics applications, and to a large family of triangulations, including acute-angled triangulations. Copyright © 2011 John Wiley & Sons, Ltd.

Received 20 September 2010; Revised 4 February 2011; Accepted 12 February 2011

KEY WORDS: immersed boundary; curve parameterization; closest-point projection; high-order finite elements

1. INTRODUCTION

We construct a method to parameterize a class of planar piecewise C^2-curves over edges of an ambient triangulation. A distinguishing feature of the method is that edges of the triangulation need not interpolate the given curve. Instead, we identify a collection of edges in the triangulation and map them onto the curve with its closest-point projection.

Such a parameterization is useful in the context of immersed boundary methods for partial differential equations posed on curves and domains with curved boundaries in two spatial dimensions. These are numerical schemes that do not require a spatial discretization that conforms to the domain of the problem. Such schemes are attractive because they alleviate the problem of meshing the domain. Instead, the domain is approximated over a background mesh. In most such methods, curves and curved boundaries are approximated by piecewise linear segments. The consistency errors that such approximations introduce in the numerical scheme are just of the right order to not dominate the convergence rate for low-order methods (up to second-order, [1, 2]). However, they typically dominate the properties of high-order schemes, see [3, 4]. For this reason, high-order immersed boundary methods are quite rare. To design such methods, it is imperative to be able to

*Correspondence to: Adrian J. Lew, Department of Mechanical Engineering, Stanford University, Durand 207, Stanford, CA 94305-4040, U.S.A.
†E-mail: lewa@stanford.edu

systematically improve the approximation of curves and curved boundaries over non-conforming meshes.

In this paper, we describe an immersed boundary method for problems on a class of planar piecewise C^2-curves. This is achieved by using the parameterization and crucially exploiting some of its properties to construct finite element spaces on these curves. We demonstrate the performance of the method in Section 7 by solving the steady-state heat equation on curves and computing the deformation of curved Timoshenko beams under load. We observe optimal convergence of the computed solutions with piecewise linear, quadratic and Hermite shape functions. Related work on the solution of partial differential equations posed on implicit surfaces can be found in [5–8].

The accurate representation of curved boundaries provided by the parameterization method could also enable the construction of high-order immersed boundary methods over planar domains. The idea is to use the parameterization described here to map triangles in the vicinity of a boundary/interface to curved ones, to achieve the accuracy required in the representation of the domain. A related idea can be found in [9]. In combination with the method for immersed curves described here, such an algorithm would be useful in the numerical solution of problems involving coupled partial differential equations in the interior and boundary of the domain. For example, in problems involving deformable structures with thin shell stiffeners on the boundary, or for hydraulic fracture problems. Since the description of the parameterization is already quite rich in details, we do not explore this application further here.

Parameterizing a planar curve over a collection of edges in an ambient triangulation requires (i) the selection of edges to include in the collection, and (ii) the construction of a bijective map from these edges onto the curve. For a simple and closed curve, our choices are sketched in Figure 1. As shown in the figure, we select all edges that have both vertices on one side of the curve and that belong to triangles with one vertex on the other side. For the map, we choose the closest-point projection onto the curve. In Section 2, we describe the parameterization method for the class of C^2-regular boundaries and in Section 5, we extend it to a class of planar piecewise C^2-curves we term *spliced* C^2-curves. The latter family of curves includes curves with corners, end points, self-intersections and T-junctions. In both cases, we discuss conditions for the combination of these two choices to define an injective parameterization, see Section 3. As a result, we restrict the class of triangulations for which the method is applicable to those in which certain angles are acute. This is still a very large class of meshes. In particular, the method is applicable to all

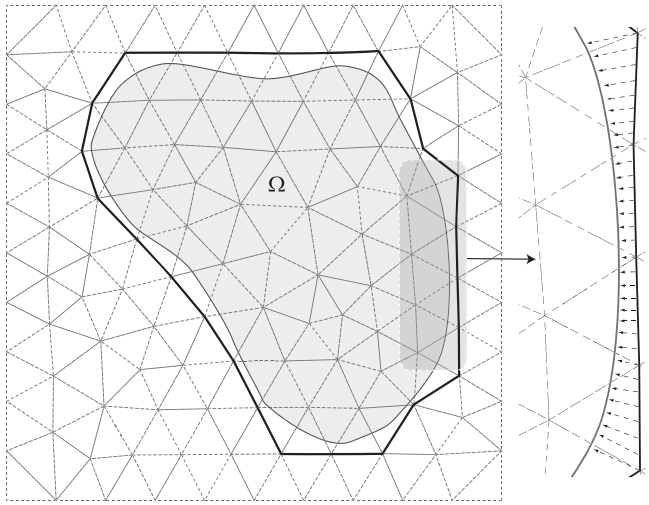


Figure 1. Simple depiction of the idea of parameterizing the boundary of a curved domain Ω over a collection of edges (in thick solid black lines) in an ambient triangulation. Each point on this collection of edges is mapped to its closest-point in the boundary $\partial\Omega$. Note that the triangulation is not assumed to interpolate the boundary.

DOI: 10.1002/nme

acute-angled triangulations (with sufficiently small mesh size) and as discussed in Section 4, to triangulations constructed by subdivision of a certain class of quadrilateral meshes.

Through examples in Sections 2–5 and 7, we showcase the following key properties of these families of curves and triangulations, which hold for *any sufficiently small mesh size*:

(i) The closest-point projection from the collection of edges selected for parameterization onto the curve is a bijection. The numerical examples in Sections 3, 4 and 7 provide evidence that the Jacobian for this parameterization is bounded and away from zero independent of the mesh size.
(ii) The collection of edges selected for parameterizing a simple connected curve is itself simple and connected. As we shall discuss in Section 6, this property notably simplifies the construction of finite element spaces of continuous functions over the curve.

Rigorous arguments to prove these statements will be provided elsewhere, along with estimates of how small the mesh size needs to be.

Issues relevant to practical implementations of the parameterization method are discussed at length in Section 4 and in the Appendix. In the former, we discuss alternatives for constructing adaptively refined triangulations suitable for the parameterization method. In the latter, we provide simple formulae for computing the closest-point projection and its derivatives for two common curve representations. In Section 4, we also highlight how some of the assumptions on the mesh can be weakened. These considerations arguably widen the context of applicability of the method.

2. PARAMETERIZATION METHOD FOR PLANAR C^2-REGULAR BOUNDARIES

We begin by describing the parameterization method for the class of planar C^2-regular boundaries. To this end, it is convenient to first introduce a few definitions. Henceforth, $\Omega \subset \mathbb{R}^2$ is a bounded open set with boundary $\Gamma = \partial\Omega$. The *signed distance function* to Γ, $\phi : \mathbb{R}^2 \to \mathbb{R}$, is defined as

$$\phi(p) = \begin{cases} -\min\limits_{q \in \partial\Omega} d(p,q) & \text{if } p \in \Omega, \\ \min\limits_{q \in \partial\Omega} d(p,q) & \text{otherwise,} \end{cases}$$

where $d(\cdot,\cdot)$ is the Euclidean distance in $\mathbb{R}^2$. Closely related to the signed distance function is the *closest-point projection* onto Γ, $\pi : \mathbb{R}^2 \to \Gamma$, which computes the point(s) in Γ closest to a given point in $\mathbb{R}^2$, i.e.,

$$\pi(p) = \arg\min_{q \in \Gamma} d(p,q).$$

Observe that by definition, $\Gamma = \phi^{-1}(\{0\})$ and $\Omega = \phi^{-1}((-\infty, 0))$. Also, since distinct points in Γ may be equidistant from a given point in $\mathbb{R}^2$, π may be multi-valued at some points in $\mathbb{R}^2$. We say that π is well defined at $p \in \mathbb{R}^2$, if $\pi(p)$ is single valued.

In particular, close enough to a smooth enough boundary π is well defined. A smooth enough boundary in this case is a C^2-regular boundary [10]. We say that the bounded open set $\Omega \subset \mathbb{R}^2$ is a C^2-regular domain and that it has a *C^2-regular boundary* Γ if there exists $\Psi \in C^2(\mathbb{R}^2, \mathbb{R})$ such that

$$\Omega = \{x \in \mathbb{R}^2 : \Psi(x) < 0\} \quad \text{and} \quad \Psi(x) = 0 \Rightarrow |\nabla\Psi| \geqslant 1.$$

The function Ψ is called a defining function for Ω.

We will additionally need to consider an ambient triangulation $\mathscr{T}_h$ in $\mathbb{R}^2$, where h indicates the maximum diameter of a triangle in $\mathscr{T}_h$ and $K \in \mathscr{T}_h$ denotes the triangle K in $\mathscr{T}_h$. Edges in the ambient triangulation will be used to parameterize Γ. To define the parameterization method, and to later provide enough conditions for it to succeed, we introduce the following definitions related

DOI: 10.1002/nme

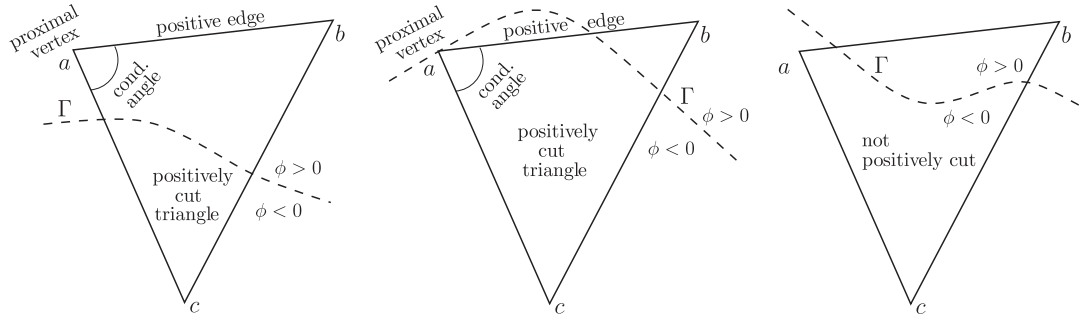


Figure 2. Examples to illustrate the definition of a triangle positively cut by Γ, its positive edge, proximal vertex and conditioning angle. Also shown is an example of a triangle that is not positively cut although it is intersected by Γ.

to $\mathscr{T}_h$:

(i) We say that Γ is *immersed* in the triangulation $\mathscr{T}_h$ if Γ is contained in the domain triangulated by $\mathscr{T}_h$, i.e. $\Gamma \subset \bigcup_{K\in\mathscr{T}_h} \overline{K}$.
(ii) A triangle $K \in \mathscr{T}_h$ is *positively cut* by Γ if $\phi \geqslant 0$ at precisely two vertices of K.
(iii) An edge in $\mathscr{T}_h$ is a *positive edge* if it joins the two vertices at which $\phi \geqslant 0$ of some triangle in $\mathscr{T}_h$ positively cut by Γ.
(iv) The *proximal vertex* of a triangle positively cut by Γ is the vertex of its positive edge closest to Γ. When both vertices of a positive edge are equidistant from Γ, either one of its vertices can be designated as the proximal vertex.
(v) The *conditioning angle* of a triangle positively cut by Γ is the interior angle at its proximal vertex.
(vi) The conditioning angle of $\mathscr{T}_h$ is the maximum among all conditioning angles of triangles positively cut by Γ in $\mathscr{T}_h$.

These definitions are illustrated in Figure 2 with specific examples.

2.1. The parameterization method

We can now define the parameterization method for any C^2-regular boundary Γ. Assume then that Γ is immersed in $\mathscr{T}_h$, and define

$$\Gamma_h = \{\text{Union of all positive edges in } \mathscr{T}_h\}.$$

Then, under some conditions briefly described next, the parameterization of Γ is the map $\pi : \Gamma_h \to \Gamma$. A simple pseudocode for the computation of Γ_h is shown in Algorithm 1.

For $\pi : \Gamma_h \to \Gamma$ to define a parameterization, it needs to be at least onto, so that all points in Γ are the closest-point projection of some point in Γ_h. Additionally, it is convenient to have $\pi : \Gamma_h \to \Gamma$ as an injective map, so that no two points in Γ_h map to the same point in Γ. As we shall see later, this may not always be the case. It is possible to prove, however, that if we consider a family of quasi-uniform meshes $\{\mathscr{T}_h\}_h$ (see, e.g. [11]) with $h \searrow 0$ such that Γ is immersed in each $\mathscr{T}_h$, and

A.1 the conditioning angle of each $\mathscr{T}_h$ is smaller than $\theta_0 < \pi/2$, for some θ_0 independent of h, and
A.2 h is small enough,

then the map $\pi : \Gamma_h \to \Gamma$ is injective and onto. This guarantees that by considering fine enough meshes with good element quality, it will be possible to construct an injective and onto parameterization $\pi : \Gamma_h \to \Gamma$ with the method introduced here. We discuss the reasons for these conditions in Section 3, and will present the proof elsewhere.

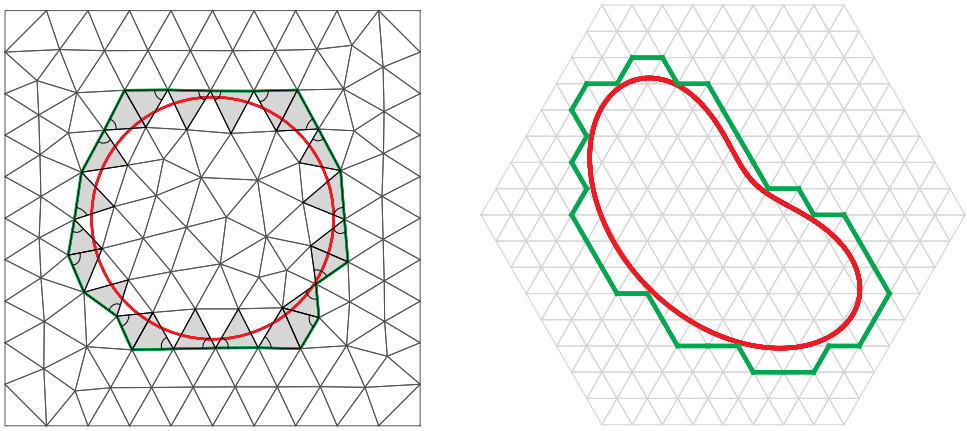
DOI: 10.1002/nme

Algorithm 1 Identification of Γ_h for C^2-regular boundaries

Input: Curve Γ with defining function Ψ, triangulation $\mathscr{T}_h$
Require: Γ is immersed in $\mathscr{T}_h$

{*Identify positive edges*}
$\mathscr{E}_h \leftarrow \emptyset$
for all triangles $K \in \mathscr{T}_h$ **do**
  $\{v_a, v_b, v_c\} \leftarrow$ vertices of K ordered such that $\Psi(v_c) \leqslant \Psi(v_a), \Psi(v_b)$
  if $\Psi(v_a), \Psi(v_b) \geqslant 0$ and $\Psi(v_c) < 0$ **then** {K *is positively cut by* Γ}
    Append edge joining v_a and v_b to $\mathscr{E}_h$
  end if
end for

Output: $\Gamma_h = \bigcup_{e \in \mathscr{E}_h} e$



(a) (b)

Figure 3. Illustrating the parameterization method for C^2-regular boundaries. In (a), a unit circle is immersed in an unstructured triangulation of a square and in (b), the crooked egg curve is immersed in a mesh of equilateral triangles. Positive edges identified for parameterization of the curves are highlighted in green.[‡]

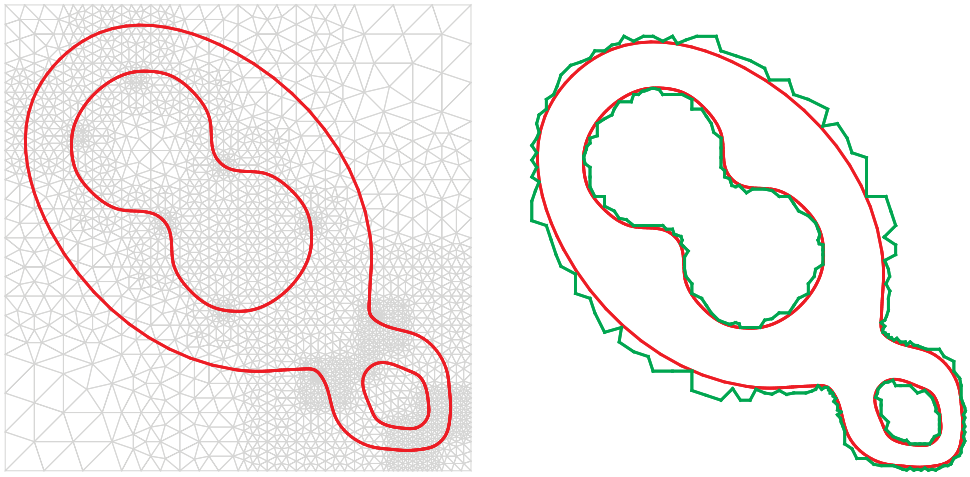
2.2. *Illustrative examples*

The following simple examples illustrate how the parameterization method works. In the three examples below, we identify the set Γ_h, and check that $\pi : \Gamma_h \to \Gamma$ is injective and onto; hence, it defines a parameterization of Γ over Γ_h. In the process, we highlight some notable features.

Example 1
Figure 3(a) shows a unit circle immersed in an unstructured triangulation of a square domain. Clearly, the circle is a C^2-regular boundary. Triangles that are positively cut by the circle are shaded in gray. In each of these triangles, the interior angle at the proximal vertex is indicated in the figure. These angles (a total of 22) were checked to be acute and hence condition **A**.1 is satisfied. The positive edges are highlighted in green. The circle is parameterized over these edges by its closest-point projection.

[‡]See electronic version of the article for interpretation of references to colors in captions for figures.

DOI: 10.1002/nme



(a) (b)

Figure 4. Parameterization of a C^2-regular boundary with multiple connected components. The curve is immersed in a mesh of acute-angled triangles, as shown in (a). The positive edges are shown in green in (b).

Example 2

The curve in Figure 3(b) is called the *crooked egg* curve and is the locus of points with $r = \sin^3\theta + \cos^3\theta$ for $\theta \in [0, 2\pi]$ in polar coordinates. It is immersed in a mesh of equilateral triangles. Since all angles in the mesh equal 60°, the conditioning angle of each triangle that is positively cut is again 60°. Therefore, condition **A**.1 is automatically satisfied. The positive edges identified for parameterization of the curve are colored in green.

Example 3

This example is shown in Figure 4. The curve is a C^2-regular boundary that has three connected components. Each connected component is a cubic spline. As seen from the figure, the curve is the boundary of a domain that is not simply connected. It is immersed in an unstructured mesh of acute-angled triangles, shown in Figure 4(a). Consequently, as in example 2, condition **A**.1 is automatically satisfied. Figure 4(b) shows the positive edges in green. In Section 3, we will examine the parameterization in this example in greater detail.

Remarks

Notice from the examples that the collection of positive edges identified for the parameterization of each connected component itself forms a simple connected curve. This can also be guaranteed under the conditions mentioned in Section 2.1, and as mentioned earlier, we shall prove elsewhere.

3. RATIONALE FOR THE PARAMETERIZATION AND CONDITIONS ON THE MESH

In this section, we discuss the rationale behind the specific choice of map and edges for defining the parameterization. We also motivate the reason behind the restrictions on the mesh, namely that the conditioning angle be acute (**A**.1) and the mesh size be small (**A**.2), by considering when the computed parameterization can be injective.

3.1. On the choice of the closest-point projection

Two attractive features of the closest-point projection are that it is defined in the entire plane of the curve and that it inherits the regularity properties of the curve, see [10, 12, 13]. In particular, since Γ is a C^2-regular boundary, π is guaranteed to be single valued and C^1 in a sufficiently small neighborhood of Γ. Furthermore, this map is an intrinsic property of the image (or trace) of the curve and not of the particular representation adopted for the curve. Hence, irrespective of

DOI: 10.1002/nme

whether Γ is given implicitly by a level set function or, for example, parametrically as a spline, the parameterization we compute is unaltered. The way to compute π can nevertheless depend on the representation adopted for Γ. Since the problem of computing the closest-point projection can be restated as one of finding local minimizers of the squared distance to the curve, efficient numerical algorithms can be used to compute it with the desired accuracy for common curve representations, such as splines [14–16]. We discuss more about the calculation of π and its derivatives in the Appendix.

3.2. On the choice of positive edges for parameterization

Having adopted the map π to parameterize the curve, there is considerable freedom in deciding which edges of the ambient triangulation to select as the domain of parameterization. It is convenient for these edges to belong to the neighborhood of Γ where π is well defined, for π may be multivalued at points far away from Γ. In principle, it is possible to compute/estimate the size of this neighborhood (see [10, Thm. 1.5]). In practice though, we can avoid such a calculation by selecting edges whose distance from Γ decreases with the mesh size h. Then, with mesh refinement, we can make h sufficiently small to ensure that π is well defined on each selected edge.

With the above discussion in mind, observe that by definition, positive edges belong to triangles that are intersected by Γ. Therefore, Γ_h is within a distance h from Γ. As the ambient mesh is refined, Γ_h gets progressively closer to Γ, eventually converging to it. In this sense, Γ_h is an ideal choice as domain of the parameterization. Furthermore, the identification of positive edges requires only an inspection of which vertices in the mesh lie in Ω. This can be done, for instance, by examining the sign of the signed distance function to Γ. In particular, there is no need to compute curve-triangle intersections.

3.3. On the acute conditioning angle assumption

Condition **A**.1 requires that each mesh $\mathscr{T}_h$ have a strictly acute conditioning angle. The rationale behind it is to ensure that the computed parameterization is injective. Figure 5 shows examples in which the restriction of π to Γ_h fails to be injective because condition **A**.1 is violated. In Figure 5(a), one of the positive edges is mapped to a point in Γ. In Figure 5(b), π is injective over each positive edge. However, the images under π of any two positive edges overlap in Γ. In both examples, there are positively cut triangles with obtuse conditioning angle. Additionally, notice that in both examples Γ can be replaced by a straight segment, and the same issue arises. This in fact demonstrates that without condition **A**.1, the method may render non-injective parameterizations irrespective of how small the mesh size is.

To explain why an acute conditioning angle can guarantee an injective parameterization, consider a positive edge e_{ab} in $\mathscr{T}_h$ that joins vertex v_a to v_b. By definition then, there is a triangle $K \in \mathscr{T}_h$

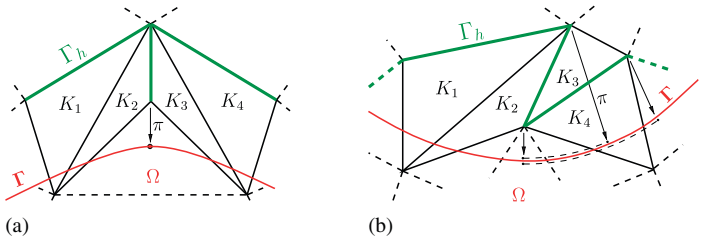


Figure 5. Examples illustrating how the restriction of π to the collection of positive edges can fail to be injective. Such a scenario can occur irrespective of the smoothness of the curve or the mesh size. However, with condition **A**.1 and a small enough mesh size (**A**.2), it is possible to guarantee injectivity: (a) triangles K_1, K_2, K_3 and K_4 are positively cut by Γ. The image of the positive edge common to K_2 and K_3 under π is a point in Γ. Notice that K_2 and K_3 have obtuse conditioning angles. (b) Triangles K_1, K_2 and K_4 are positively cut by Γ. The restriction of π to each positive edge is injective. However, their images in Γ overlap. Notice that K_2 has an obtuse conditioning angle.

DOI: 10.1002/nme

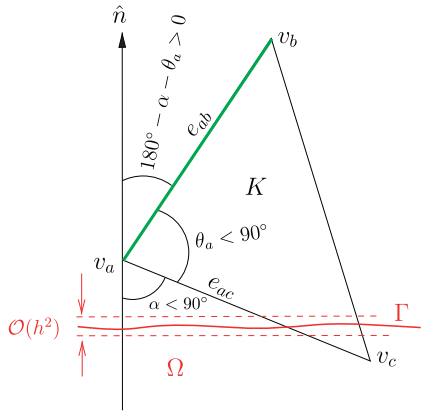


Figure 6. Rationale behind restriction on angles in the mesh. Triangle K is positively cut by Γ and has positive edge e_{ab} and proximal vertex v_a. The unit outward normal to Γ at $\pi(v_a)$ is denoted by $\hat{n}$. Since Γ is a smooth curve, it can be locally bounded between two parallel lines that are perpendicular to $\hat{n}$ and separated by distance that is $\mathcal{O}(h^2)$. Using this along with the facts that $v_a \in \mathbb{R}^2 \setminus \Omega$ and $v_c \in \Omega$ shows that the angle between edge e_{ac} and $\hat{n}$ has to be greater than $90^\circ - \mathcal{O}(h)$. In the figure, this angle is depicted to be strictly obtuse. If the interior angle at v_a in triangle K is acute, i.e. $\theta_a < 90^\circ$, then the angle between edge e_{ab} and $\hat{n}$ is non-zero for sufficiently small h. The assumption that v_a is at least as close to Γ as v_b makes it impossible for the angle between e_{ab} and $\hat{n}$ to be 180°.

with vertices $\{v_a, v_b, v_c\}$ such that $v_a, v_b \in \mathbb{R}^2 \setminus \Omega$ and $v_c \in \Omega$, see Figure 6. Let h be sufficiently small so that π is well defined in e_{ab}. If π fails to be injective over e_{ab}, then there are points $p, q \in e_{ab}$ such that $\pi(p) = \pi(q)$. Since p is close to Γ, the line joining p and $\pi(p)$ is normal to Γ at the point $\pi(p)$. Therefore, $p - \pi(p)$ and $q - \pi(p)$ are both parallel to the normal to Γ at $\pi(p)$. This in turn implies that $p - q$, and hence the edge e_{ab} is parallel to the normal to Γ at $\pi(p)$. Thus, we conclude that π fails to be injective over e_{ab} precisely when $\pi(e_{ab})$ is a point in Γ. Therefore, to ensure that $\pi|_{e_{ab}}$ is injective, it suffices to ensure that e_{ab} is not parallel to the normal to Γ at either $\pi(v_a)$ or $\pi(v_b)$. This is precisely the purpose of the acute conditioning angle assumption as explained below.

As depicted in Figure 6, suppose that v_a is the proximal vertex of K. Let $\hat{n}$ be the unit outward normal to Γ at $\pi(v_a)$ and let e_{ac} be the edge joining vertex v_a to vertex v_c. Since $v_a \in \mathbb{R}^2 \setminus \Omega$ and $v_c \in \Omega$, using the smoothness of Γ, it is possible to show that the angle between $\hat{n}$ and e_{ac} is greater than $90^\circ - Ch$, where $C > 0$ is a constant independent of h but dependent on the curve. Then, as depicted in Figure 6, if the interior angle at v_a in triangle K is strictly acute, we cannot have $e_{ab} \cdot \hat{n} = 1$ for any h sufficiently small. More precisely, if the angle at v_a is smaller than some $\theta_0 < 90^\circ$ independent of h, we only need h to be small enough to ensure $Ch < 90^\circ - \theta_0$. The assumption that v_a is at least as close to Γ as v_b excludes the possibility of $e_{ab} \cdot \hat{n} = -1$, again, provided h is small. Hence, e_{ab} is not parallel to $\hat{n}$. Following the argument in the previous paragraph, this implies that π is injective on e_{ab}. In this way, we conclude that if a positively cut triangle K has an acute conditioning angle, then the restriction of π to its positive edge is injective.

More involved arguments are required to justify that a small mesh size and an acute conditioning angle are sufficient to guarantee global injectivity as well as the observations made at the end of the previous section. We now revisit example 3 of Section 2 and consider the parameterization shown in Figure 4. In light of the above discussion, for a point p on a positive edge, we examine (i) the angle between the positive edge and the normal to Γ at $\pi(p)$ and (ii) the Jacobian of the parameterization at p. Figure 7 shows a histogram of these two quantities upon uniformly sampling each of the 332 positive edges at 11 points. In particular, note that the angle is bounded away from 0° and 180° and that the Jacobian is bounded away from 0. In fact, all angles were found to lie between 21° and 159°. The conditioning angle of the mesh shown in Figure 4(a) was close to 80°. The values of the Jacobian were between 0.34 and 1.33.

DOI: 10.1002/nme

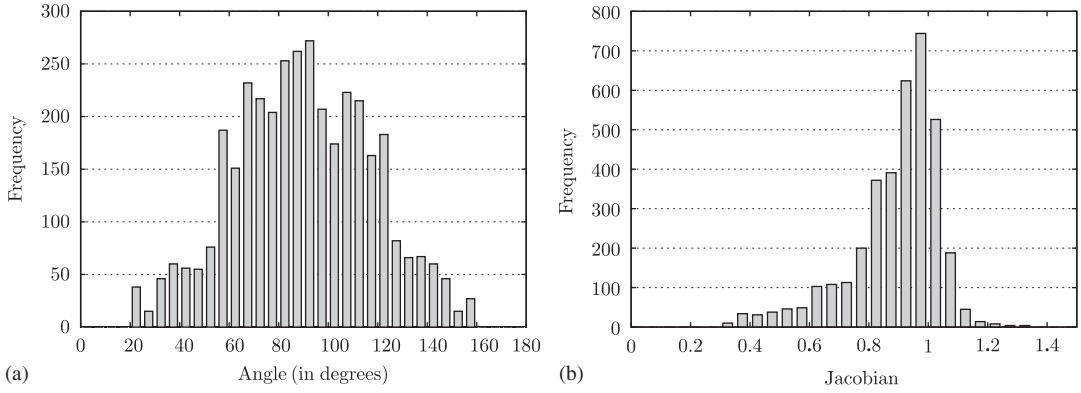


Figure 7. For points p distributed along the positive edges in Figure 4(b), shown here are histograms for (a) the angle between the positive edge containing p and the normal to Γ at $\pi(p)$ and (b) the Jacobian of the parameterization at p. Values of angles were contained in $[21^\circ, 159^\circ]$ and values of the Jacobian were contained in $[0.34, 1.33]$.

The injectivity of π over Γ_h does not deteriorate as the mesh is refined. In particular, for sufficiently small mesh sizes, the Jacobian of the parameterization is bounded away from 0 and ∞ irrespective of the mesh size. This is a key property for the construction of convergent finite element methods on Γ. We showcase this property with a numerical example in Section 7, see Figure 18(b). In Section 4.4, we discuss with an example the fact that an acute conditioning angle is sufficient although not necessary to ensure an injective parameterization. We also discuss the possibility of relaxing this angle condition.

3.4. On the requirement of small mesh size

The requirement of a small enough mesh size expressed by condition **A**.2 cannot be over emphasized. Even if condition **A**.1 is satisfied, a large mesh size can result in (i) the parameterization being non-injective, (ii) Γ_h not being a closed curve, or (iii) connected components of Γ_h not being simple. This is best illustrated through examples. The example shown in Figure 8 demonstrates that h needs to be small, at least when compared with the local radius of curvature of Γ. The triangulation shown on the left in Figure 8(a) consists only of equilateral triangles and therefore has conditioning angle 60° with Γ. However, because h is too large, the images under π of different positive edges overlap in Γ. When the background mesh is refined with a self-similar subdivision, thereby halving the mesh size and retaining the same conditioning angle, π is injective over the positive edges. In the examples shown in Figures 8(b) and (c), the curves Γ (in red) and Γ_h (in green) are topologically different precisely because the mesh size is too large. In Figure 8(b), unlike Γ, Γ_h is neither a closed curve nor is it a connected set. In Figure 8(c), Γ_h fails to be a simple curve although Γ is. When the meshes in Figures 8(b) and (c) are refined by a self-similar subdivision of triangles as done in Figure 8(a), these discrepancies are remedied.

4. ON TRIANGULATIONS WITH ACUTE CONDITIONING ANGLE

The acute conditioning angle condition requires certain angles in the background triangulation to be acute. The number of angles required to be acute is precisely the number of triangles that are positively cut. A simple way to ensure this is to just require that triangles in the vicinity of the curve be acute angled. Even simpler, require an acute-angled triangulation, i.e., a triangulation consisting of only acute-angled triangles. Since the polygon to be triangulated is quite arbitrary, we can even construct triangulations consisting of only equilateral triangles, as done in the example shown in Figure 3(b).

DOI: 10.1002/nme

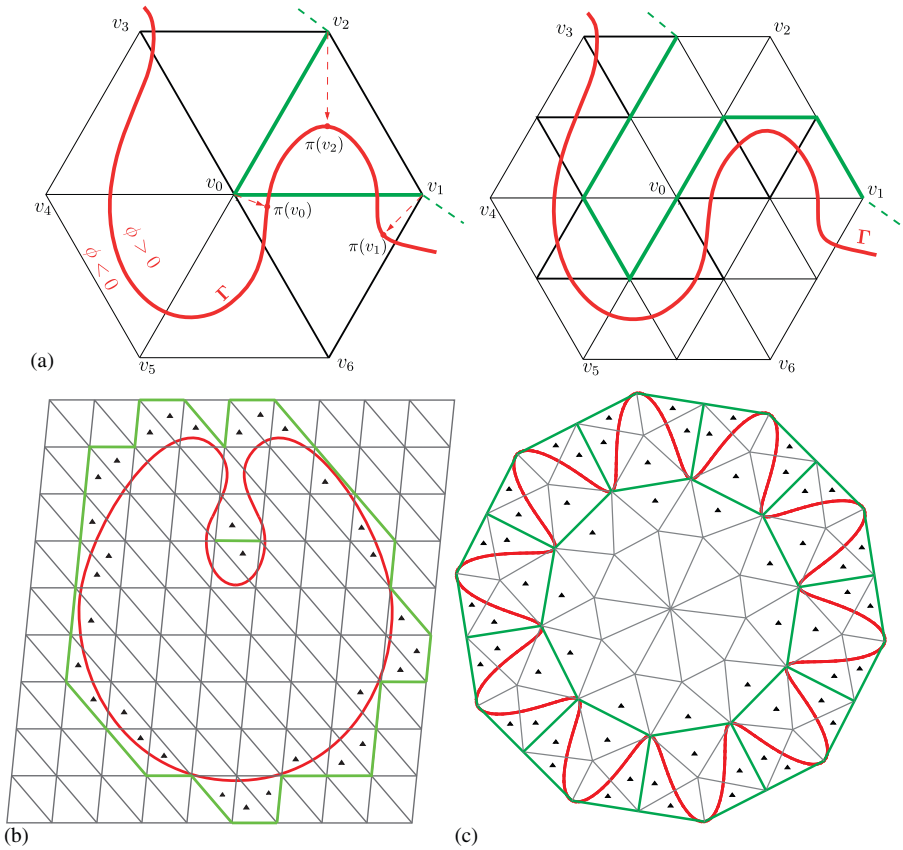


Figure 8. Examples to emphasize the requirement of a small mesh size in the parameterization method: (a) π fails to be injective on the positive edges (in green) joining v_0 with v_1 and v_2 because their images overlap in Γ (in red). This is rectified by simply refining the mesh with a self-similar subdivision of triangles; (b) positively cut triangles are indicated by small black markers. While Γ (in red) is a closed and connected curve, Γ_h (in green) is neither; notice the isolated positive edge near the neck of Γ. This is remedied by subdividing the background mesh; and (c) positively cut triangles are marked by black markers. While Γ (in red) is a simple curve, Γ_h (in green) is not. Clearly, the parameterization over positive edges is not injective. This is again rectified by subdividing the background mesh.

In this section, we discuss ways to construct and refine triangulations that satisfy the acute conditioning angle assumption. We showcase a way to construct adaptively refined acute triangulations from quadtrees and a construction of triangle meshes based on subdividing certain classes of quadrilateral meshes. Triangulations resulting from either construction automatically have an acute conditioning angle. Finally, we also discuss the possibility of relaxing this assumption.

4.1. Adaptively refined acute-angled triangulations

In practical applications, it is desirable to use meshes that are adaptively refined based on the local features of the curve. For example, the mesh size could be related to the local curvature. A convenient and commonly adopted way of constructing adaptively refined triangulations is with quadtree spatial decompositions, see [17, 18]. This is usually done by constructing stencils of triangulations for each square in an adaptively refined quadtree. Since neighboring squares can have different levels of refinement, these stencils need to be carefully constructed. The quadtree is therefore balanced prior to triangulation, so that neighboring squares have sides whose lengths differ by a factor of at most two. For the purpose of our method, just any triangulation stencil would not suffice since we require that some angles be acute. The stencils given in [19] result in an acute-angled triangulation. The interior angles in each triangle are bounded above by 80°.

DOI: 10.1002/nme

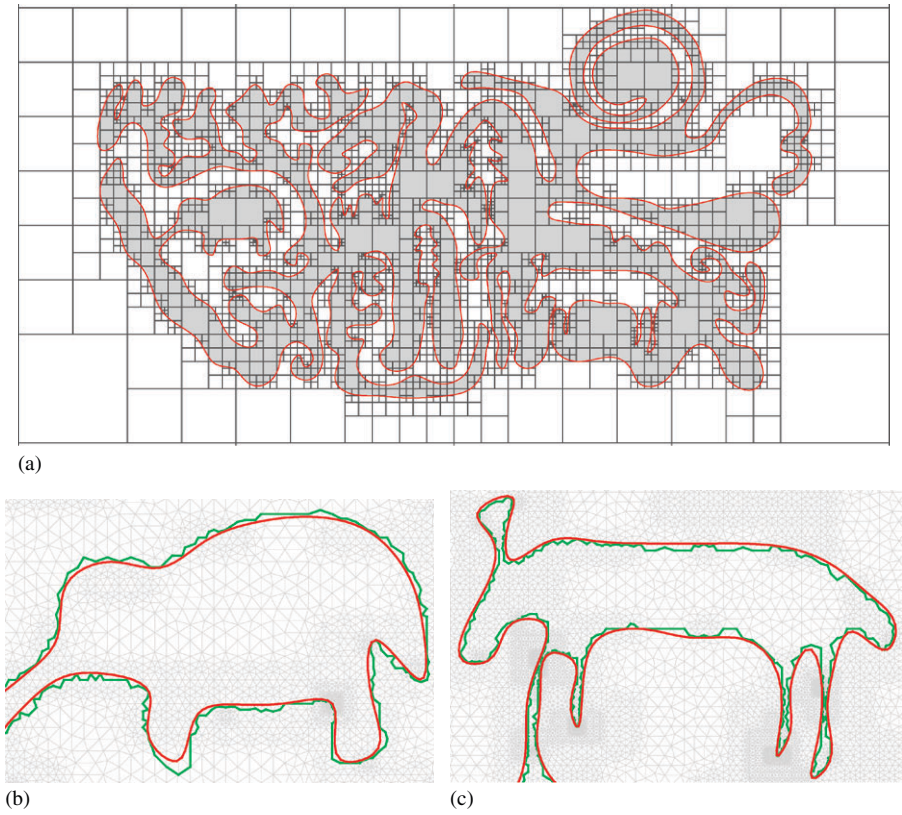


Figure 9. Adaptively refined acute-angled triangulations can be constructed by triangulating an adapted quadtree decomposition and using the stencils in [19]. Shown here is a cubic spline (in red) with many small features and widely varying curvatures. The domain enclosed by the curve is shaded in gray (notice the various animal and tree shapes). A background quadtree structure was constructed by decomposing a set of sample points on the curve into separate squares. Parts of the mesh resulting from triangulating the quadtree are shown in (b) and (c), along with the corresponding collection of positive edges. The mesh size distribution was not optimized; hence, some regions of the mesh are significantly finer than what is necessary to obtain a good parameterization. However, notice that smaller mesh sizes are needed in regions of the curve with large curvatures and small features.

Figure 9 shows an example of a curve with many small features and widely varying curvatures (notice the various animal shapes). The curve is a cubic spline that was constructed by approximately digitizing Figure 2.19 of [20]. To demonstrate that the curve is in fact closed and simple, the domain enclosed by it is shaded in gray. In order to construct an adaptively refined background mesh for this curve, a quadtree was built by requesting that a set of sample points on the curve belong to separate squares of the tree. Then the stencils from [19] were used to triangulate the quadtree. The size distribution was not optimized in any way; hence, some regions of the resulting mesh were much finer than what is needed to obtain a good parameterization of the curve. Since the resulting mesh was quite dense, it is not included here. Instead, two small sections of the mesh are shown in Figures 9(b) and (c). The entire collection of 6493 positive edges is shown in Figure 10. Observe that like the curve itself, the positive edges form a simple, closed curve. This was not the case with a coarse background mesh.

As was done for the example in Figure 4, we examine (i) the angle between positive edges and local normals and (ii) the Jacobian for the parameterization. Figures 11(a) and (b) show histograms for these two quantities. All angles were found to lie in the range $[12^\circ, 171^\circ]$ while values of the Jacobian belonged to $[0.15, 4.1]$. As before, angles are bounded away from 0° and 180° and the Jacobian is bounded away from 0. Numerical experiments and the discussion in Section 3.3 show that a smaller conditioning angle will improve these ranges.

DOI: 10.1002/nme

Figure 10. The collection of positive edges identified for the curve shown in Figure 9 is shown in green. The curve itself is shown in dashed lines.

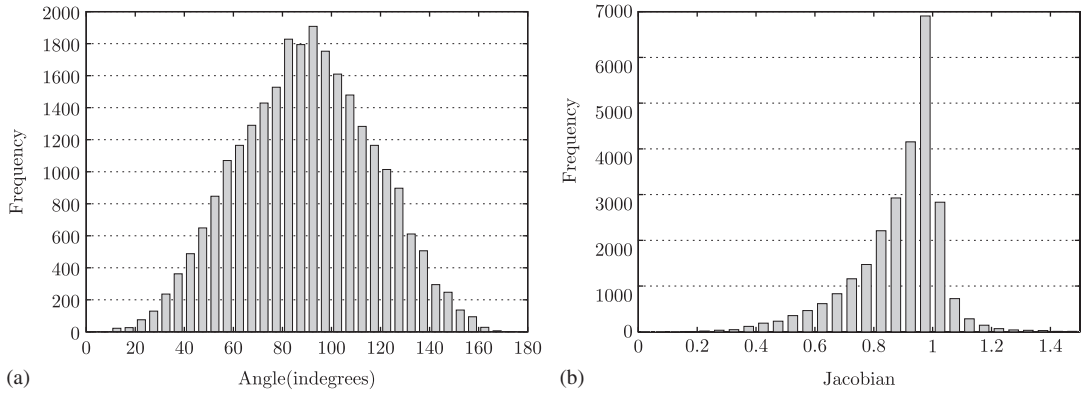


Figure 11. Examination of the parameterization for the example shown in Figure 10. Histogram plots for the (i) angle between positive edges and local normals and (ii) Jacobian of the parameterization are shown in (a) and (b), respectively.

4.2. Refinement of triangulations with acute conditioning angles

Since the parameterization method requires the mesh size h to be small, a systematic mesh refinement algorithm may be needed. Whether the refinement of a mesh with acute conditioning angle also has an acute conditioning angle depends both on how the curve intersects triangles in the mesh as well as on the refinement algorithm. For example, some mesh refinement strategies applied to a mesh of all acute-angled triangles may render meshes whose conditioning angles are obtuse or too close to 90°.

DOI: 10.1002/nme

In all the examples shown in Section 7, we have adopted a self-similar subdivision of triangles. In this refinement procedure, each triangle is divided into four triangles by joining midpoints of edges. The mesh size is halved with every refinement. Moreover, the refinement of an acute-angled triangulation remains acute-angled and hence the conditioning angle remains acute irrespective of the mesh size. This is not the case, for instance, with a longest-edge bisection scheme. Such approach may make the refinement of an acute-angled triangulation contain obtuse angles.

4.3. An algorithm based on subdivision of quadrilateral meshes

We describe now a simple construction of triangulations having acute conditioning angles by subdividing certain types of quadrangulations. Let $\mathcal{Q}_h$ be a mesh consisting of non-degenerate convex quadrilaterals over a polygon. We assume that in each quadrilateral in $\mathcal{Q}_h$ (at least the ones in the vicinity of the curve), the angles between each diagonal and its adjacent edges are acute. We construct a triangulation by splitting each quadrilateral in $\mathcal{Q}_h$ into two triangles by connecting a pair of diagonally opposite vertices. By selectively deciding which vertices to join, we can ensure that the resulting triangulation has an acute conditioning angle with Γ, see Figures 12(a) and (b).

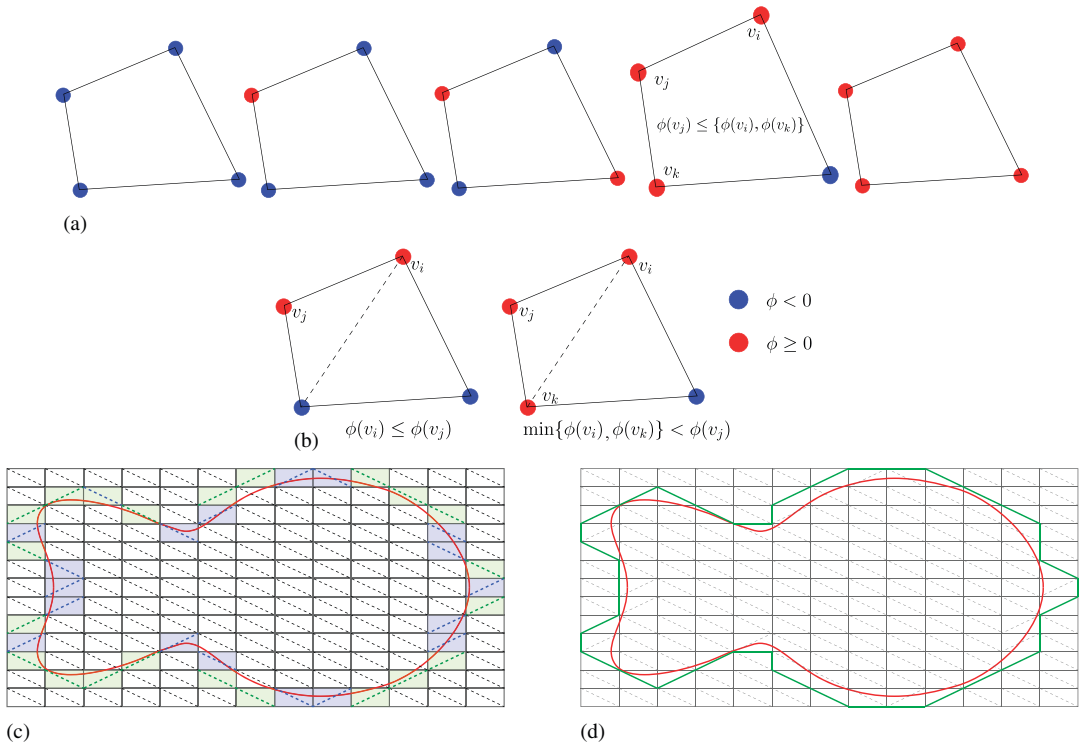


Figure 12. Constructing triangulations with acute conditioning angle by subdividing quadrilateral meshes in which the angles between each diagonal and its adjacent edges are acute. The stencils to subdivide each quadrilateral are shown in (a) and (b). Which stencil to use for each quadrilateral depends on the value of the signed distance function at the vertices. These stencils are constructed such that the resulting positively cut triangles have an acute conditioning angle. Either diagonal can be joined for the cases shown in (a) while a particular diagonal needs to be joined for the ones shown in (b). Figure (c) shows a spline immersed in a mesh of rectangles. Using the stencils in (a) and (b), each rectangle is split into two triangles. The rectangles shaded in blue with blue diagonals are the ones that correspond to the stencil on the left in (b). The rectangles shaded in green with green diagonals are the ones corresponding to the stencil on the right in (b). The resulting triangulation and positive edges are shown in (d).

DOI: 10.1002/nme

Consider a quadrilateral Q in $\mathcal{Q}_h$ with vertices $\{v_i, v_j, v_k, v_\ell\}$ ordered counter-clockwise. Let ϕ be the signed distance function to the curve to be parameterized. Denote by n_+ the number of vertices of Q where $\phi \geqslant 0$.

(i) Join any two diagonally opposite vertices if (or deciding based on the quality of resulting triangles)

  (a) $n_+ = 0$, or
  (b) $n_+ = 1$, or
  (c) $n_+ = 2$ with $\phi \geqslant 0$ at two diagonally opposite vertices, or
  (d) $n_+ = 3$ with $\phi(v_\ell) < 0$ and $\phi(v_j) \leqslant \min\{\phi(v_i), \phi(v_k)\}$.

  These are the cases shown in Figure 12(a). Two cases remain (see Figure 12(b)).

(ii) If $n_+ = 2$ with $\phi \geqslant 0$ at two adjacent vertices, say $0 \leqslant \phi(v_i) \leqslant \phi(v_j)$, join v_i and its diagonally opposite vertex.

(iii) If $n_+ = 3$ with $\phi(v_\ell) < 0$ and $\phi(v_j) > \min\{\phi(v_i), \phi(v_k)\}$, join vertices v_i and v_k.

The resulting collections of triangles that are positively cut and positive edges are easily identified by inspection. It is a simple exercise to check that because of the assumption on angles in Q, the triangles that are positively cut by the curve after subdivision of Q have acute conditioning angles. Of course, the above stencil is only modulo cyclic permutations of the vertex ordering $\{v_i, v_j, v_k, v_\ell\}$.

In addition to outlining a simple way to construct triangulations with acute conditioning angles, this construction also provides a method for parameterization of Γ over edges and diagonals of certain types of *quadrilateral meshes*. Structured meshes are an important family of quadrilateral meshes that satisfy the assumption of acute angles between diagonals and adjacent edges. Indeed, any mesh consisting of parallelograms, in particular, rectangles and squares, satisfies such an assumption. Figures 12(c) and (d) show an example with a spline immersed in a mesh of rectangles.

4.4. *Acute conditioning angle*: *a sufficient but not necessary condition for injectivity*

The rationale behind requiring an acute conditioning angle to compute an injective parameterization was discussed in Section 3.3. As mentioned there, it is easy to see that this is just a sufficient condition, not a necessary one. Figure 13 shows an instance in which an injective parameterization is computed despite the fact that each triangle that is positively cut has an obtuse conditioning angle.

It is in fact fairly obvious that an acute conditioning angle is not necessary. For once the positive edges have been identified, the closest-point projection from these edges onto the curve does not depend on the background mesh in any way. That is, the parameterization computed depends only on the edges selected and the curve itself, not on the connectivity of the background triangulation or on what angles the selected edges make with other edges in the triangulation. Instead, the acute conditioning angle condition provides a means to control the orientation of the positive edges with respect to the curve.

It should therefore be possible to relax the angle condition. Within the scope of the following discussion, we introduce some notation related to triangulations. By a triangulation $\mathcal{T}_h$ of a polygon, we mean a pairing (V, C), where V is the list of vertices, i.e. points in $\mathbb{R}^2$, and C,

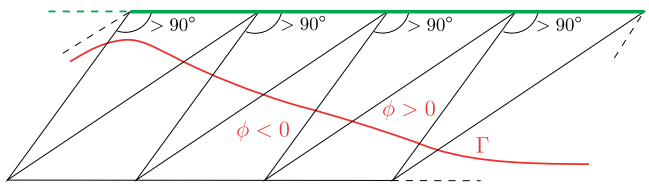


Figure 13. Qualitative demonstration of the fact that an acute conditioning angle is sufficient but not necessary for injectivity in the parameterization method. In the example shown, each triangle positively cut by Γ has an obtuse conditioning angle. However, the closest-point projection is injective over these edges.

DOI: 10.1002/nme

the connectivity table, is a collection of 3-tuples in $V \times V \times V$. By $(x, y, z) \in \mathscr{T}_h$, we refer to the triangle in $\mathscr{T}_h$ with connectivity $(x, y, z) \in C$. For the C^2-regular boundary Γ, we say that triangulations $\mathscr{T}_h = (V, C)$ and $\mathscr{T}'_{h'} = (V', C')$ are *positive edge equivalent with respect to* Γ or $\mathscr{T}_h \overset{\Gamma}{\sim} \mathscr{T}'_{h'}$, if

(i) Γ is immersed in both $\mathscr{T}_h$ and $\mathscr{T}'_{h'}$,
(ii) there exists a bijective map $\Phi : V \mapsto V'$ such that

(a) $(x, y, z) \in C \iff (\Phi(x), \Phi(y), \Phi(z)) \in C'$,
(b) $(x, y, z) \in \mathscr{T}_h$ is positively cut $\iff (\Phi(x), \Phi(y), \Phi(z)) \in \mathscr{T}'_{h'}$ is positively cut, and
(c) if $v \in V$ is the vertex of a positive edge, then $\Phi(v) = v$.

It follows from the definition of the equivalence relation $\overset{\Gamma}{\sim}$ that if $\mathscr{T}_h \overset{\Gamma}{\sim} \mathscr{T}'_{h'}$, then the two triangulations $\mathscr{T}_h$ and $\mathscr{T}'_{h'}$ have the *same* collection of positive edges. Therefore, our method computes the *same* parameterization for Γ with both triangulations. The key point is that for small h, we can guarantee an injective parameterization of Γ using $\mathscr{T}_h$ by simply knowing the existence of a member in its equivalence class with acute conditioning angle. This in turn reduces to determining if the map Φ in the definition of $\overset{\Gamma}{\sim}$ can be constructed. Doing so is easy for the example shown in Figure 13. The required map is the one that translates the vertices opposite to the positive edge in each triangle that is positively cut, by a small distance to the right. However, such a map may not exist, as is the case for the examples shown in Figure 5.

5. EXTENSION OF THE PARAMETERIZATION METHOD TO SPLICED C^2-CURVES

In this section, we extend the parameterization method to a larger class of curves, called spliced C^2-curves. This family of curves includes C^2-regular boundaries, and more notably, curves with endpoints, corners, self-intersections and T-junctions. A curve γ is a *spliced C^2-curve* if it can be written as $\gamma = \bigcup_{i=1}^{m} \gamma^i$, for some $m \in \mathbb{N}$, where for each $i = 1, \ldots, m$, the curve γ^i is a closed and connected subset of a C^2-regular boundary Γ^i, and these curves satisfy that

$$\gamma^i \cap \gamma^j = \partial_{\mathrm{r}} \gamma^i \cap \partial_{\mathrm{r}} \gamma^j \quad \forall i, j \in \{1, \ldots m\}, \;\; i \neq j,$$

where $\partial_{\mathrm{r}} \gamma^i$ is the boundary of γ^i in the topology relative to Γ^i. Each curve γ^i is called a *component curve* of γ, and each curve Γ^i is called the *parent curve* of the component curve γ^i. Finally, the points in $\bigcup_{i=1}^{m} \partial_{\mathrm{r}} \gamma^i$ are called the *corners* of γ.

As the name suggests, a spliced C^2-curve is constructed by splicing end-to-end, closed, connected subsets of C^2-regular boundaries. Spliced C^2-curves themselves need not be closed or be the boundary of an open set in $\mathbb{R}^2$. A spliced C^2-curve need not even be simple, but note that each point of self-intersection is necessarily a corner of the curve. Corners include not only visible kinks in the curve, i.e. where the tangent fails to be continuous, but also points where the tangent fails to be differentiable.

From the definition, it follows that a spliced C^2-curve with one component and no corners is a C^2-regular boundary. For example, the boundary of a disc is a C^2-regular boundary and hence a spliced C^2-curve. The boundary of a polygonal domain is a spliced C^2-curve, although not a C^2-regular boundary.

As with C^2-regular boundaries in Section 2, we say that a spliced C^2-curve γ is immersed in a triangulation $\mathscr{T}_h$ if γ is contained in the domain triangulated by $\mathscr{T}_h$.

We extend the notion of triangles being positively cut by γ in a component-wise manner. Consider a component curve γ^i of γ with parent curve Γ^i and denote the signed distance function and the closest-point projection of Γ^i by ${}^i\phi$ and ${}^i\pi$, respectively. We say that a triangle $K \in \mathscr{T}_h$ is positively cut by γ^i if

(i) ${}^i\phi \geqslant 0$ at precisely two vertices of K, say ${}^i\phi(v_1), {}^i\phi(v_2) \geqslant 0$, and
(ii) ${}^i\pi(v_1) \cup {}^i\pi(v_2) \subset \gamma^i$.

DOI: 10.1002/nme

Algorithm 2 Identification of positive edges for a component curve γ^i of a spliced C^2-curve.

Input: Component curve γ^i of a spliced C^2-curve, parent curve Γ^i with defining function ${}^i\Psi$ and closest-point projection ${}^i\pi$, triangulation $\mathscr{T}_h$
Require: γ^i is immersed in $\mathscr{T}_h$, corners of γ^i are vertices of $\mathscr{T}_h$

```
{Identify positive edges}
E_h^i ← ∅
for all triangles K ∈ T_h do
  {v_a, v_b, v_c} ← vertices of K ordered such that ^iΨ(v_c) ≤ ^iΨ(v_a), ^iΨ(v_b)
  if ^iΨ(v_a), ^iΨ(v_b) ≥ 0 and ^iΨ(v_c) < 0 then
    if ^iπ(v_a) ∪ ^iπ(v_b) ⊂ γ^i then {K is positively cut by γ^i}
      Append edge joining v_a and v_b to E_h^i
    end if
  end if
end for
Output: γ_h^i = ⋃_{e∈E_h^i} e
```

With this notion of triangle being positively cut, the notions of positive edges, proximal vertices and conditioning angles are defined just as in Section 2, but with respect to each component curve.[‡]

5.1. *Parameterization method for spliced C^2-curves*

The parameterization method for a spliced C^2-curve γ proceeds essentially by parameterizing each component curve γ^i as in Section 2. The main difference here is that each γ^i is only a subset of the C^2-regular boundary Γ^i; hence, for the parameterization method in Section 2 to be applicable, we need to further request the corners of each γ^i to coincide with some vertices in $\mathscr{T}_h$. Under these condition, we define the domain of the parameterization for component curve γ^i as

$$\gamma_h^i = \{\text{Union of positive edges with respect to } \gamma^i \text{ in } \mathscr{T}_h\}.$$

The parameterization of each component γ^i follows then as ${}^i\pi: \gamma_h^i \to \gamma^i$, for $0 \leqslant i \leqslant m$.

By construction then, a similar result to that in Section 2 applies: Consider a family of quasi-uniform triangulations $\{\mathscr{T}_h\}_h$ with $h \searrow 0$ such that the spliced C^2-curve $\gamma = \bigcup_{i=1}^m \gamma^i$ with component curves $\{\gamma^i\}_i$ is immersed in $\mathscr{T}_h$ for each h, and assume that

- **B**.1 for each h and each $1 \leqslant i \leqslant m$, the conditioning angle of $\mathscr{T}_h$ computed with respect to γ^i is smaller than $\theta_0 < \pi/2$, for some θ_0 independent of h and i,[§]
- **B**.2 for each h, there is a vertex of $\mathscr{T}_h$ coincident with each corner of γ, and
- **B**.3 h is small enough,

then the map ${}^i\pi: \gamma_h^i \to \gamma^i$ is injective and onto, for $1 \leqslant i \leqslant m$.

The pseudocode to identify each set γ_h^i is very similar to that in algorithm 1. The main difference is that in this case it is also necessary to check condition (ii) mentioned above to confirm that a triangle is positively cut by a component curve. For completeness, the pseudocode in this case is shown in Algorithm 2.

For spliced C^2-curves, it is possible to have edges that are positive with respect to more than one component curve, especially near vertices. Hence, there may be one or more edges that belong to $\gamma_h^i \cap \gamma_h^j$ with $i \neq j$. For this reason, we can only claim that ${}^i\pi: \gamma_h^i \to \gamma^i$ is one-to-one for each i.

[‡]Specifically, by replacing Γ by the component curve in the definition of proximal vertex, conditioning angles and positive edge.

[§]This can be equivalently written as $\sup_h \max_{1 \leqslant i \leqslant m} \max_{K \in \mathscr{P}_h^i} \{\text{Conditioning angle of } K\} < \pi/2$, where $\mathscr{P}_h^i$ is the collection of triangles in $\mathscr{T}_h$ positively cut by γ^i.

DOI: 10.1002/nme

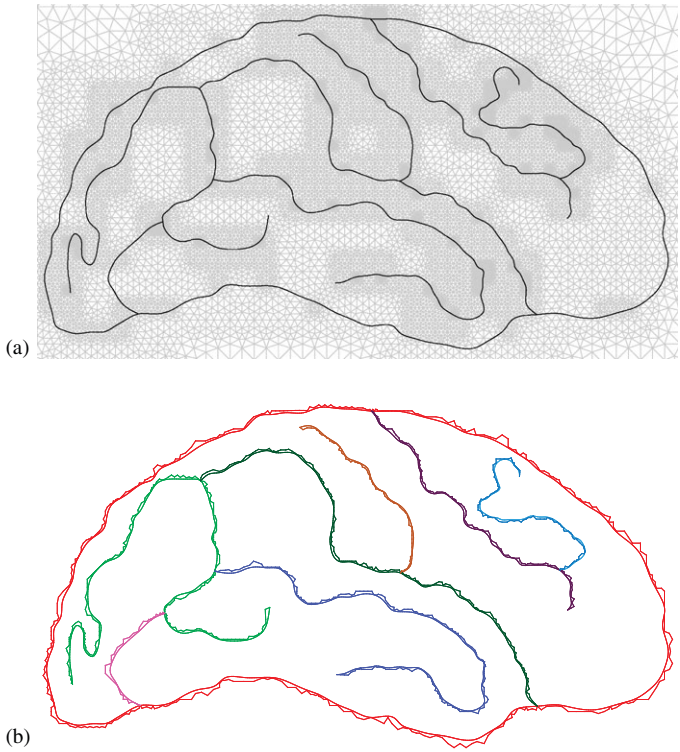


Figure 14. Illustration of the parameterization method for a spliced C^2-curve. (a) Shown in black is a spliced C^2-curve with 8 component curves and 14 corners (8 T-junctions and 6 end points). It is immersed in an acute-angled triangulation that has a vertex coinciding with each corner. (b) Each component curve is shown here in a different color along with its corresponding set of positive edges. Notice that each one of these sets of positive edges always start and end at the end points of its component curve, by construction.

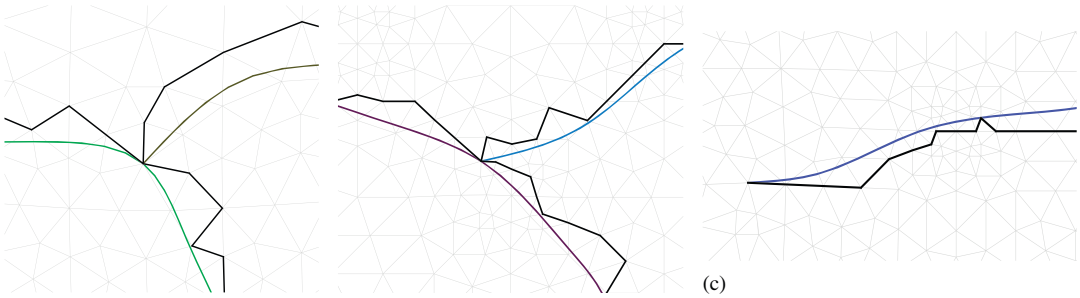


Figure 15. Close up views near three corners in the example shown in Figure 14. The corners in (a) and (b) are T-junctions while the one in (c) is an end point. Positive edges are shown in solid black lines.

This may not hold for the parameterization of γ over the entire collection of edges $\bigcup_i \gamma_h^i$, since some points in $\bigcup_i \gamma_h^i$ may be mapped to two or more points in γ. Further restrictions on the mesh are needed for this.

Figure 14 shows an example in which a spliced C^2-curve is immersed in an acute-angled triangulation. The curve has eight component curves each of which is a cubic spline and only one of which is a C^2-regular boundary. The curve has 14 corners which includes 8 T-junctions (or self-intersections) and 6 end points. The background mesh has a vertex coincident with each of these corners, as shown in the close-up views in Figure 15. Each pair of component curves and

DOI: 10.1002/nme

positive edges is shown in a different color in Figure 14(b). The background mesh is much finer than what is minimally required to obtain an injective and onto parameterization.

5.2. *On triangulations having vertices coincident with corners*

If the curve γ has corners, we require the background mesh $\mathscr{T}_h$ to have a vertex coincident with each corner point. This step is included for convenience, to avoid parameterizations that end at points in the middle of an edge. This does not reduce the generality of the approach, since as we discuss below, it is often not necessary to *a priori* include the corners as vertices of the triangulation.

We suppose that triangles in $\mathscr{T}_h$ in the vicinity of γ are acute angled. Then, it is possible to locally perturb a few vertices around each corner to ensure that the resulting triangulation still has an acute conditioning angle with each component curve of γ. The following is an example of one such perturbation map. Let c be a corner of γ and suppose that $\mathscr{T}_h$ does not have a vertex coincident with c. Identify $v_\star$ to be (one of) the vertex of $\mathscr{T}_h$ closest to c, i.e.,

$$v_\star = \arg\min_{v\in\mathscr{T}_h} d(v,c).$$

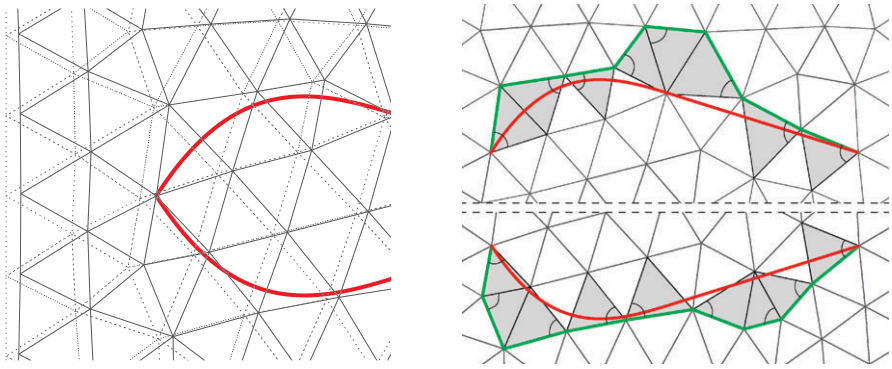
With $r_\star > 0$, consider the following map defined over vertices of $\mathscr{T}_h$:

$$v \mapsto \begin{cases} v + \left(1 - \dfrac{d(v_\star, v)}{r_\star}\right)(c - v_\star) & \text{if } d(v, v_\star) < r_\star, \\ v & \text{otherwise.} \end{cases} \tag{1}$$

Under the action of this map, vertex $v_\star$ is mapped to the corner c. Vertices that are at distances larger than $r_\star$ remain undisturbed. Furthermore, no vertex is moved by a distance larger than $d(c, v_\star)$, which by definition of $v_\star$ is less than or equal to the mesh size h. In this sense, the above map is a *small* and *local* perturbation of vertices of $\mathscr{T}_h$ around the corner c.

The curve in Figure 16 is a spliced C^2-curve with two components ($\gamma^\pm$) and two corners. It is the profile of a member of the NACA five digit airfoil series where

$$\gamma^\pm = \left\{ (x,y) : y = \begin{cases} \pm\dfrac{k}{6}(x^3 - 3mx^2 + m^2(3-m)x), & 0 \leqslant x \leqslant m, \\ \pm\dfrac{km^3}{6}(1-x), & m < x \leqslant 1 \end{cases} \right\} \quad \text{with } k = 15,\ \ m = 0.5.$$



(a) (b)

Figure 16. Example to illustrate the idea of perturbing vertices of the background mesh near corners to satisfy condition **B**.1. The curve shown in red is a symmetric airfoil with two corners and two component curves. The background mesh is the one shown in Figure 3(a). As shown in dotted lines in (a), the background mesh does not have vertices at the corners. This is rectified after perturbing the vertices, as shown in solid lines. For each component curve, figure (b) shows the triangles that are positively cut in gray, positive edges in green and conditioning angles for each component curve.

DOI: 10.1002/nme

When immersed in the background mesh of Figure 3(a), vertices of the mesh do not coincide with the corners, as shown in dotted lines in Figure 16(a). This is remedied by perturbing the mesh locally around the corners with the map in (1), as shown in solid lines in Figure 16(a). Figure 16(b) shows for each component curve, the triangles that are positively cut (shaded in gray), the positive edges (in green) and the conditioning angles that were checked to be acute.

For the map given by (1), vertex perturbations span a distance of $r_\star$ around each corner. The size of the parameter $r_\star$, which is a few edge lengths, is decided based on the requirement that the resulting triangulation have an acute conditioning angle with the component curves of γ. Since the perturbation map is continuous, making $r_\star$ large enough will ensure that changes in interior angles of triangles in $\mathscr{T}_h$ due to vertex adjustments are sufficiently small. Instead of selecting a (conservative) large enough value for $r_\star$, an iterative procedure can be adopted. We determine the smallest $n \in \mathbb{N}$ for which $r_\star = nh$ ensures that the triangulation post vertex adjustments has an acute conditioning angle.

With a large mesh size, vertex perturbations can affect a large fraction of vertices in the mesh. This was the case in the example shown in Figure 16(a), where the mesh was kept coarse for illustration purposes. It is even possible that the same vertex is perturbed by maps corresponding to different corners. With a small enough mesh size, each vertex in the mesh is perturbed by the map of at most one corner. This is the case in the example shown in Figure 21(a).

The perturbation map in (1) is not the only one possible. Similar maps are easy to construct.

5.3. On the need of geometric tolerances

Computationally determining whether a vertex is inside or outside the domain defined by a parent curve often needs the introduction of tolerances when evaluating the sign of the defining function. Fortunately, the method is quite robust with respect to this determination. This is because failing to accurately determine whether a vertex lies inside a domain or not is equivalent to parameterizing a nearby smooth curve, one that is at a distance of the order of machine precision (or tolerance used for computing the defining function) from the actual curve.

However, some additional care is needed for corners. For sufficiently small mesh sizes, the vertex of the background mesh coinciding with a corner necessarily belongs to a positive edge. Therefore, we automatically identify corners as vertices of a positive edge. In this way, we avoid having to discern whether the value of the defining function is strictly negative or not there.

6. APPLICATION TO THE CONSTRUCTION OF FINITE ELEMENT SPACES ON IMMERSED CURVES

Next, we apply the parameterization to the construction of finite element spaces on curves. For simplicity, we describe the construction only for a connected C^2-regular boundary Γ. The extension to spliced C^2-curves follows naturally. We denote the collection of positive edges in the (sufficiently refined) background mesh $\mathscr{T}_h$ by $\mathscr{E}_h$ and the union of these edges by Γ_h. For each $e \in \mathscr{E}_h$, we denote by $u^e \in \mathbb{R}^2$ a unit vector parallel to e, and by $J^e = |\nabla\pi \cdot u^e|$ the Jacobian of the parameterization. Note that for any $f \in L^1(\Gamma)$, we have

$$\int_\Gamma f \, \mathrm{d}\Gamma = \sum_{e \in \mathscr{E}_h} \int_{\xi \in e} (f \circ \pi) J^e \, \mathrm{d}\xi, \tag{2}$$

which can be used to integrate over Γ.

A C^0 finite element space on Γ is obtained by constructing a finite element space over Γ_h and then pushing these functions forward to Γ through a composition with π, the closest-point projection of Γ. Three aspects of our method are crucial for this. First, the domain of parameterization is a collection of edges. Constructing finite-dimensional spaces over Γ_h is straightforward. Second, constructing a C^0 finite element space over Γ_h is simplified precisely because like Γ, Γ_h is a simple connected curve. If it was not, continuity across edges in Γ_h may have to be imposed with constraints. Finally, since $\pi : \Gamma_h \to \Gamma$ is injective and onto, the space of functions over Γ obtained

DOI: 10.1002/nme

by composing finite element functions defined on Γ_h with the map π, is well defined. That is, the resulting finite element space on Γ consists of functions that are defined for all points in Γ and that are single valued.

Let $V_h^e \subset C^2(e, \mathbb{R})$ be a space of finite element functions over $e \in \mathscr{E}_h$. A C^0 space of finite element of functions on Γ_h is given by

$$\hat{V}_h = \{v_h : \Gamma_h \to \mathbb{R} : v_h|_e \in V_h^e \ \ \forall e \in \mathscr{E}_h, \text{ and } v_h \in C^0(\Gamma_h, \mathbb{R})\}.$$

We then have the following C^0 finite element space on Γ:

$$V_h = \{\hat{v}_h \circ \pi^{-1} : \hat{v}_h \in \hat{V}_h\}.$$

In the following, we give specific examples of such spaces by constructing Lagrange and Hermite elements on Γ. The latter consists of functions in $C^1(\Gamma, \mathbb{R})$. To this end, we denote the space of polynomials of degree less than or equal to k on $\hat{I} = [0, 1]$ by $\mathbb{P}_k(\hat{I})$. For each $e \in \mathscr{E}_h$, $M^e : \hat{I} \to e$ is the affine map from $\hat{I}$ to the edge e. We set $P_e = \pi(e)$.

6.1. *Lagrange finite elements*

Following the notation in [11], let $\{\hat{I}, \mathbb{P}_k(\hat{I}), \hat{\Sigma}_k\}$ be the reference Lagrange finite element on $\hat{I}$, where $\hat{\Sigma}_k = \{\hat{\sigma}_i\}_{i=1}^{k+1}$ is the set of degrees of freedom. The points

$$\{0 = z_1 < z_2 < \cdots < z_{k+1} = 1\} \subseteq \hat{I}$$

are such that for each $\hat{v} \in C^0(\hat{I})$,

$$\hat{\sigma}_i(\hat{v}) = \hat{v}(z_i), \quad 1 \leqslant i \leqslant k+1.$$

Let $\{\hat{N}_i\}_{i=1}^{k+1}$ denote the basis for $\mathbb{P}_k(\hat{I})$ dual to $\hat{\Sigma}_k$ so that $\hat{N}_i(z_j) = \delta_{ij}$.

We define a finite element over the patch P_e in Γ as $\{P_e, V_h^k(P_e), \Sigma_k^e\}$, where

$$V_h^k(P_e) = \text{SPAN}\{\hat{N}_i \circ (M^e)^{-1} \circ (\pi)^{-1}\}$$

as sketched in Figure 17, and the degrees of freedom $\Sigma_k^e = \{\sigma_i^e\}_{i=1}^{k+1}$ as

$$\sigma_i^e(v) = \hat{\sigma}_i(v \circ \pi \circ M^e) \quad \forall v \in V_h^k(P_e), \ \ 1 \leqslant i \leqslant k+1.$$

Finally, the Lagrange finite element space on Γ is given by

$$V_h^k = \prod_{e \in \mathscr{E}_h} V_h^k(P_e).$$

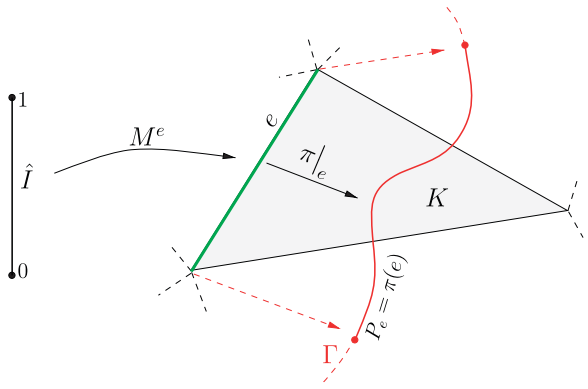


Figure 17. Illustration to explain the construction of finite element spaces on curves. The reference element is the segment $[0, 1]$. With an affine map M^e, the reference element is mapped to an edge e. Then, the closest-point projection π maps the edge e onto the curve Γ.

DOI: 10.1002/nme

As mentioned before, due to the fact that Γ_h is simple and connected, the above definition implies that $V_h^k \subset C^0(\Gamma, \mathbb{R})$.

In order to define an interpolation operator $\mathscr{I}_h : H^1(\Gamma) \to V_h^k$, note that π is C^1 on Γ_h. Therefore, $v \in H^1(P_e) \Longleftrightarrow v \circ \pi \in H^1(e)$, and hence $\sigma_i^e(v)$ is well-defined. The interpolation operator is defined piecewise as

$$\mathscr{I}_h(v)|_{P_e} = \sum_{i=1}^{k+1} \sigma_i^e(v) \hat{N}_i \circ (M^e)^{-1} \circ (\pi)^{-1}.$$

6.2. Cubic Hermite elements

Let $\{\hat{I}, \mathbb{P}_3(\hat{I}), \hat{\Sigma}_H\}$ denote the reference cubic Hermite element where the four degrees of freedom in $\hat{\Sigma}_H$ are

$$\hat{\sigma}_1(v) = v(0), \quad \hat{\sigma}_2(v) = v(1),$$
$$\hat{\sigma}_3(v) = v'(0), \quad \hat{\sigma}_4(v) = v'(1) \quad \forall v \in C^1(\hat{I}).$$

The shape functions $\{\hat{H}_i\}$ dual to $\hat{\Sigma}_H$ are

$$\hat{H}_1(\xi) = 2\xi^3 - 3\xi^2 + 1,$$
$$\hat{H}_2(\xi) = -2\xi^3 + 3\xi^2,$$
$$\hat{H}_3(\xi) = \xi^3 - 2\xi^2 + \xi,$$
$$\hat{H}_4(\xi) = \xi^3 - \xi^2.$$

As before, define the finite element $\{P_e, V_h^H(P_e), \Sigma_H^e\}$ over each patch P_e as

$$V_h^H(P_e) = \text{SPAN}\{\hat{H}_i \circ (M^e)^{-1} \circ (\pi)^{-1}\}.$$

The degrees of freedom Σ_H^e are now defined as

$$\sigma_i^e(v) = \hat{\sigma}_i(v \circ \pi \circ M^e) \quad \text{for } i = 1, 2,$$
$$\sigma_3^e(v) = \frac{1}{|e||J^e(0)|}\, \hat{\sigma}_3(v \circ \pi \circ M^e),$$
$$\sigma_4^e(v) = \frac{1}{|e||J^e(1)|}\, \hat{\sigma}_4(v \circ \pi \circ M^e),$$

where $|e|$ is the length of edge e. The finite element space on Γ can now be defined as

$$V_h^H = \prod_{e \in \mathscr{E}_h} V_h^H(P_e).$$

By scaling the degrees of freedom $\hat{\sigma}_3$ and $\hat{\sigma}_4$ that interpolate the derivatives, we have ensured that $V_h^H \subset C^1(\Gamma, \mathbb{R})$. An interpolation operator $\mathscr{I}_h : H^2(\Gamma) \mapsto V_h^H$ is defined in a natural way assuming that Γ is at least a C^3-regular boundary, namely,

$$\mathscr{I}_h(v)|_{P_e} = \sum_{i=1}^{4} \sigma_i^e(v) \hat{H}_i \circ (M^e)^{-1} \circ (\pi)^{-1}.$$

7. NUMERICAL EXAMPLES WITH FINITE ELEMENTS

With numerical examples, we showcase the performance of the parameterization method and the construction of finite element spaces described above. We first look at the convergence rate of

DOI: 10.1002/nme

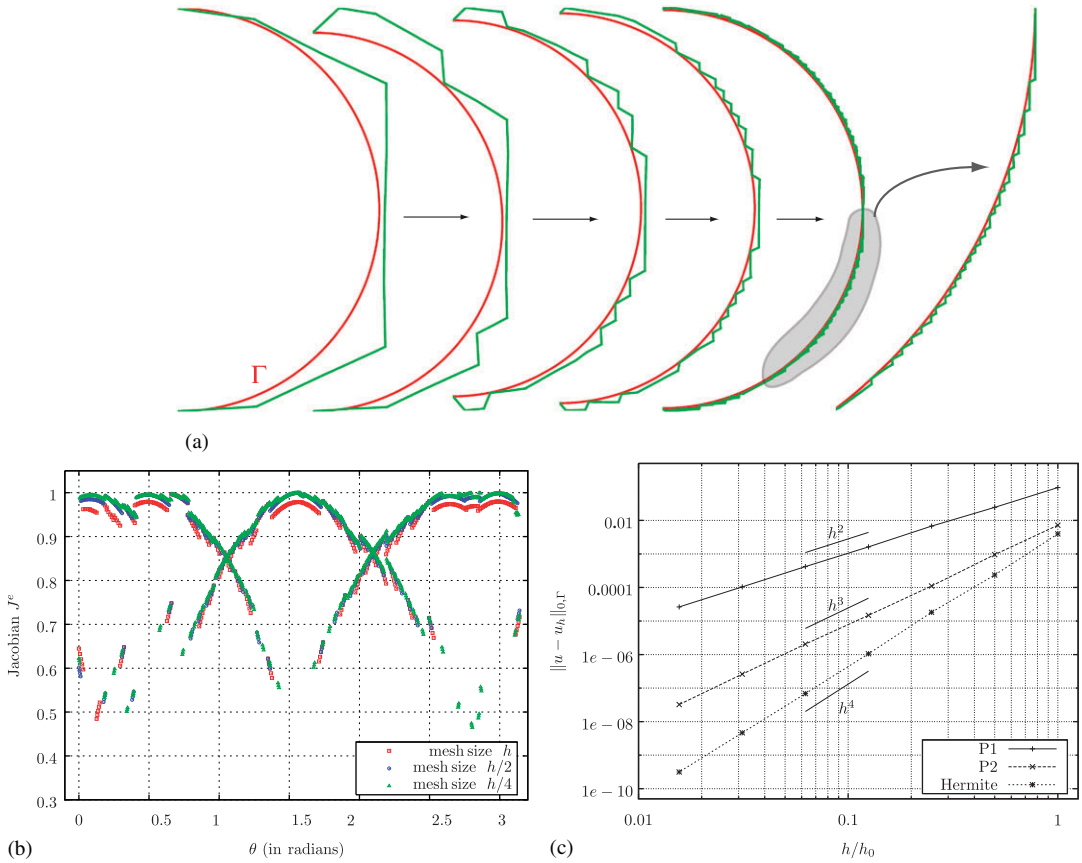


Figure 18. Jacobian of the parameterization and convergence of interpolation error as a function of mesh size: (a) A semi-circle (in red) is immersed in the background mesh shown in Figure 3(a). The triangulation is refined with a self-similar subdivision and vertices near the two corners are perturbed to satisfy **B**.2. The positive edges in successive refinements of the background mesh are shown in green in (a); (b) Jacobian as a function of angle for the parameterization of the semi-circle over positive edges for different mesh sizes. Notice that the Jacobian is bounded and away from zero independent of the mesh size; and (c) $L^2(\gamma)$-norm of the interpolation errors $(u - \mathcal{I}_h u)$ as a function of mesh size for $u(\theta) = \cos\theta\cos 2\theta$ for the interpolation operators $\mathcal{I}_h$ in defined Sections 6.1 and 6.2. Note the optimal convergence for the finite element spaces V_h^1, V_h^2 and V_h^H.

the finite element interpolation operators defined in Sections 6.1 and 6.2. We also verify that the Jacobians of the parameterization remain bounded and away from zero independent of h (as we expect from the acute conditioning angle requirement). We then show numerical examples that approximate the solutions to the steady-state heat equation and the Timoshenko beam model over curved beams.

7.1. Optimal convergence of the finite element interpolants

In the following example, we demonstrate the optimal convergence of the finite element interpolants in the L^2-norm for the spaces defined above. The curve γ is a semi-circle (a one-component spliced C^2-curve) immersed in the triangulation shown in Figure 3(a), and in four successively refined triangulations. Each finer triangulation was obtained by subdividing each triangle in the mesh into four self-similar ones. Figure 18(a) shows γ in red, and the positive edges in green. The background triangulations are not shown.

The Jacobian for the parameterization of γ is plotted for three different values of h in Figure 18(b). We observe that the Jacobian is never close to zero. In fact, it is bounded from below and above independently of h.

DOI: 10.1002/nme

For the function $u=\sin\theta\cos 2\theta, \theta\in[0,\pi]$, the $L^2(\gamma)$-norm of the interpolation error $(u-\mathscr{I}_h u)$ is plotted in Figure 18(c). We note that the interpolation error converges optimally for the finite element spaces V_h^1, V_h^2 and V_h^H on Γ.

7.2. *Steady-state heat equation*

Solving the steady-state heat equation on a curve γ consists of determining $u:\gamma\to\mathbb{R}$ that minimizes the functional

$$I[u]=\int_\gamma\left(\frac{1}{2}\nabla_s u\cdot\nabla_s u-fu\right)\mathrm{d}\gamma, \tag{3}$$

where $f:\gamma\to\mathbb{R}$ is a given function and ∇_s is the tangential gradient operator. The discrete problem on γ is to determine u_h that belongs to a finite element space V_h such that

$$\int_\gamma \nabla_s u_h\cdot\nabla_s v_h\,\mathrm{d}\gamma=\int_\gamma f v_h\,\mathrm{d}\gamma\quad\forall v_h\in V_h^\partial,$$

where $V_h^\partial\subset V_h$ is the set of admissible test functions. By adopting a basis $\{N_i\}$ for the space V_h^∂, we obtain a set of linear equations that needs to be solved, namely

$$\mathbf{K}\mathbf{U}_h=\mathbf{F}\quad\text{where } K_{ij}=\int_\gamma\nabla_s N_i\cdot\nabla_s N_j\,\mathrm{d}\gamma,$$

$$F_i=\int_\gamma f N_i\,\mathrm{d}\gamma.$$

Integrals over γ in the above expressions are performed over the reference element $\hat{I}$ by a change of variables, as in (2).

We show an example with the semi-circle and positive edges depicted in Figure 18(a). The source function is defined as

$$f(\theta)=-\frac{1}{r^2}(4\cos\theta\sin 2\theta+5\sin\theta\cos 2\theta),$$

where r is the radius of the semi-circle and θ is the angular coordinate. The solution of (3) is the function $u(\theta)=\sin\theta\cos 2\theta,\ \theta\in[0,\pi]$, and is plotted in Figure 19(a). Homogeneous Dirichlet boundary conditions were imposed at both end points ($\theta=0,\pi$); hence, in this case $V_h=V_h^\partial$. Finite

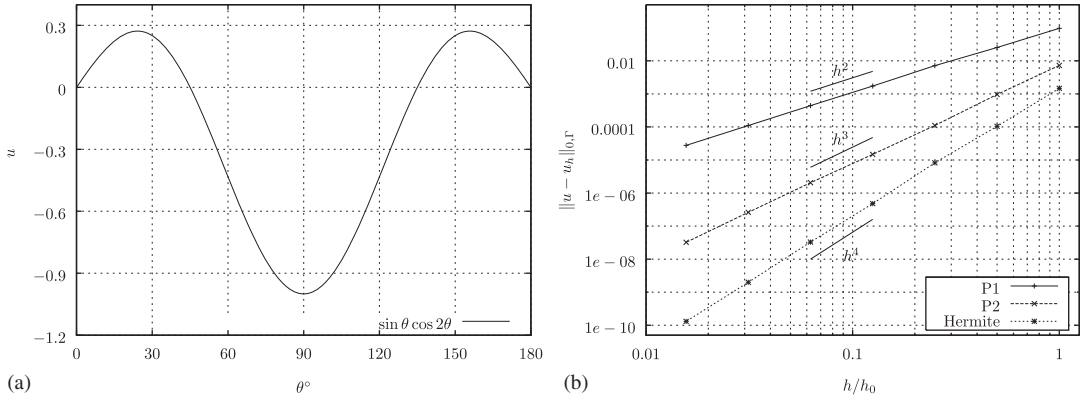


Figure 19. Convergence of the computed finite element solution u_h to the exact solution $u(\theta)=\sin\theta\cos 2\theta$ for the steady heat equation on a semi-circle. The exact solution is plotted in (a). The rate of convergence of the error measured in the $L^2(\gamma)$-norm is seen to be optimal for each one of the finite element spaces V_h^1, V_h^2 and V_h^H, as shown in (b).

DOI: 10.1002/nme

element approximations u_h were computed with the spaces V_h^1, V_h^2 and V_h^H. Figure 19(b) shows the convergence of u_h to u in the $L^2(\gamma)$-norm. The rates of convergence are optimal.

7.3. Timoshenko beam model

Next, we consider a Timoshenko beam model for a thin beam $\mathscr{B}$ in $\mathbb{R}^2$, which is one of the form

$$\mathscr{B} = \{\xi + \zeta\hat{n}(\xi) : \xi \in \gamma, \zeta \in (-H, H)\}.$$

The set $\gamma \subset \mathscr{B}$, a spliced C^3-curve, is the mid-curve of the domain. The thickness of the domain, given by $2H > 0$, is assumed to be small compared with the length of γ. Assuming small rotations of material fibers normal to the mid-curve, the displacement field $\mathbf{u} : \mathscr{B} \to \mathbb{R}^2$ is of the form

$$\mathbf{u}(\xi, \zeta) = \mathbf{u}_m(\xi) + \zeta\Theta(\xi)\hat{t}(\xi), \tag{4}$$

where $\mathbf{u}_m : \gamma \to \mathbb{R}^2$, $\Theta : \gamma \to \mathbb{R}$ and $\hat{t}(\xi)$ is the unit tangent to γ at the point ξ. The infinitesimal strain tensor $\boldsymbol{\varepsilon}(\mathbf{u})$ for the displacement field $\mathbf{u}$ is defined as $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}})$, with components $\boldsymbol{\varepsilon}_{ij}(\mathbf{u})$ in a standard Cartesian basis for $\mathbb{R}^2$. With an isotropic linear elastic material model for $\mathscr{B}$, $\mathbf{u}$ is determined as a stationary point of the functional

$$I[\mathbf{u}] = \int_{\mathscr{B}} (\lambda(\mathrm{tr}[\boldsymbol{\varepsilon}(\mathbf{u})])^2 + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}) - \mathbf{B} \cdot \mathbf{u})\, \mathrm{d}\mathscr{B}, \tag{5}$$

where $\mathrm{tr}[\boldsymbol{\varepsilon}(\mathbf{u})] = \boldsymbol{\varepsilon}_{11}(\mathbf{u}) + \boldsymbol{\varepsilon}_{22}(\mathbf{u})$, $\mathbf{B} : \mathscr{B} \to \mathbb{R}^2$ is a given body force per unit volume of $\mathscr{B}$ and λ, μ are known material parameters called Lamé constants. Since the unknowns $\mathbf{u}_m$ and Θ are both defined over γ, the problem in (5) is in fact one over γ although integration along the thickness coordinate ζ is still required.

The discrete problem is defined just as for the heat equation: we seek $(\mathbf{u}_m^h, \Theta_h) \in [V_h^1]^2 \times V_h^1$ which renders the functional (5) stationary.

We present two examples with this model. In the first, we set the mid-curve of the beam to be the semi-circle of radius r parameterized over the most refined positive edges shown in Figure 18(a). The material constants are set to $\lambda/\mu = 1$, the thickness to $2H = r/15$, and the body force $\mathbf{B}/\mu = -0.01e_y$, where e_y is the unit vector pointing toward the top of the page. The two ends of the semi-circle are rigidly fixed by setting $\mathbf{u}_m = 0$ and $\Theta = 0$ there. The deformed configuration of the mid-curve is shown in Figure 20(a).

To evaluate the accuracy of the solution, we also computed the solution of the linear elasticity problem in which the kinematic constraint (4) is not imposed. In this case, we solve for the displacement field over $\mathscr{B}$ with a body-fitting mesh of triangles, C^0-piecewise affine finite elements, imposing Dirichlet boundary condition $\mathbf{u} = 0$ at the two ends of the strip, and traction-free boundary condition on the rest of the boundary. Of course, in this case the small thickness of $\mathscr{B}$ requires the elements to be rather small throughout the mesh. Figure 20(b) shows the corresponding two-dimensional calculation for the deformation of $\mathscr{B}$.

Figure 20(c) plots the Cartesian components of the displacement along the mid-curve computed with the Timoshenko beam model and with the two-dimensional linear elastic model. Both are essentially on top of each other, indicating that: (a) the kinematic assumption (4) is a good one in this case, and more importantly, (b) the finite element method we created over γ using the parameterization yields an accurate solution.

Figure 21 shows a similar calculation but for a beam whose mid-curve is the letter 'S'. This curve is a spliced C^3-curve consisting of three component curves. Two of the component curves are arcs of a circle of radius 2 units and are drawn in red and orange in Figure 21(a), while the third is a line segment that is drawn in blue. The thickness of the beam normal to the mid-curve is 0.2 units. Figure 21(a) shows the mid-curve immersed in a triangulation that is a refinement of the one shown in Figure 3(a). Also shown are the small vertex adjustments near each of the four corners. The given triangulation is drawn with dotted lines. Around each corner, vertex adjustments span a distance of about five edge lengths. After these adjustments, there is a vertex coincident with each corner. The positive edges for parameterization of the mid-curve are drawn in green in

DOI: 10.1002/nme

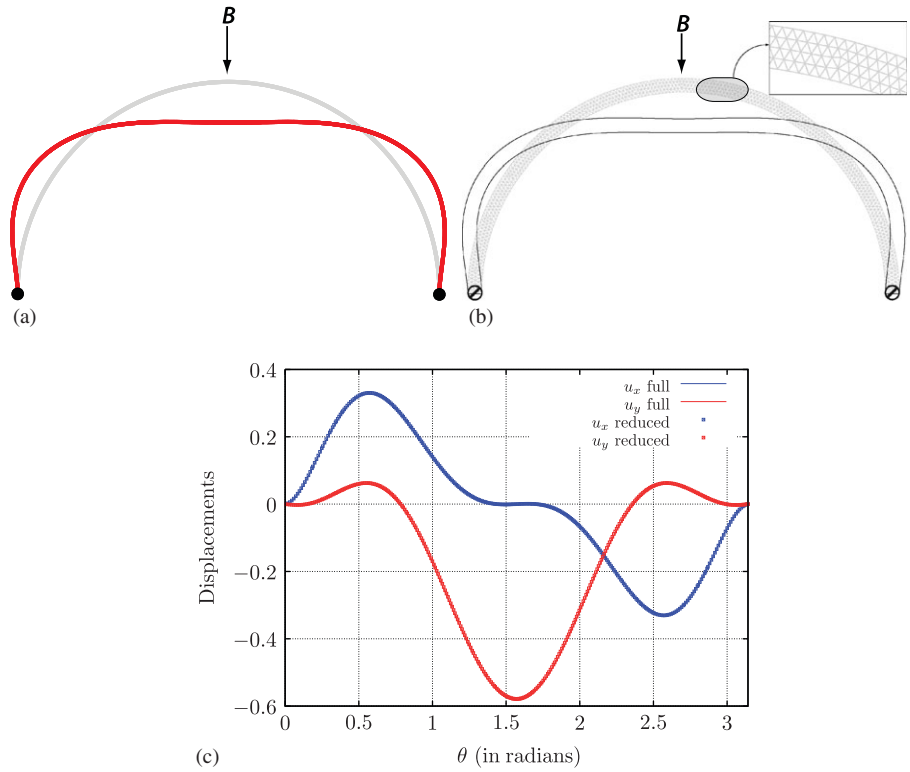


Figure 20. Computing the elastic deformation of a thin Timoshenko beam. The mid-curve of the beam is a semi-circle of radius r and the thickness normal to it is $H = r/30$. Figure (a) shows the deformed mid-curve in red, in the presence of a constant body force acting downward. The deformation was also computed with a linear elastic model on a two-dimensional mesh, as shown in figure (b). The mesh in the reference or undeformed configuration is shown in light gray while the boundary of the deformed configuration is shown in black. Figure (c) compares the components of the displacement field in the vertical and horizontal directions along the mid-curve, computed with both the exact (or full) and the reduced order model. The very good agreement between the two indicates that the finite element method over γ used to solve the reduced order model is returning an accurate solution.

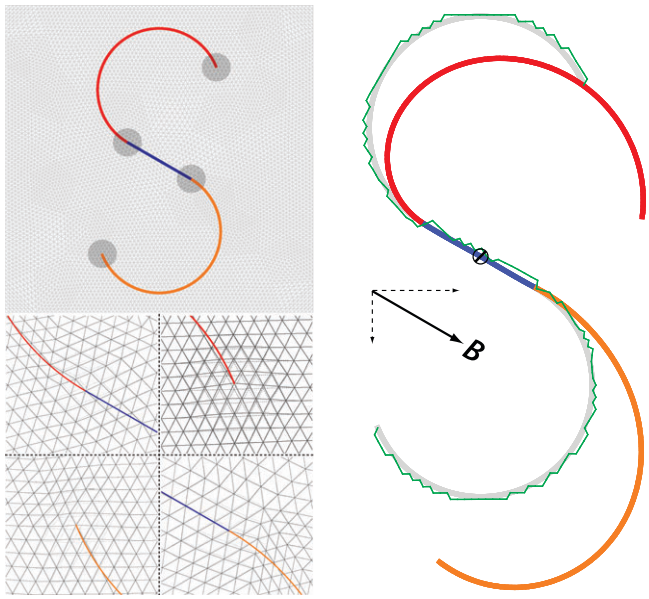
Figure 21(b). A cross-section of the beam is rigidly fixed by setting $\mathbf{u}_m = 0$ and $\Theta = 0$ at a point on the mid-curve. A constant body force per unit volume $\mathbf{B} = 5 \times 10^{-3}(\cos(\pi/6)e_x - \sin(\pi/6)e_y)$ is applied, where e_x and e_y are unit vectors pointing toward the right and top margins of the page, respectively. The equilibrium deformed configuration of the mid-curve is also sketched. The body force has been scaled to produce a large deformation for visual effect.

We conclude this section by referring to [21] for a general discussion on the merits of the Timoshenko beam model and numerical issues with shear locking when the thickness is made small. We highlight that the construction we have for Hermite elements can also be used for the Euler–Bernoulli beam model, because the finite element space constructed consists of C^1 functions.

8. CONCLUSIONS

We have proposed a parameterization method for planar curves over edges of an ambient triangulation. With limited assumptions on the classes of curves and triangulations, we demonstrated through numerous examples some interesting features of the method, including its robustness. These features were vital in the construction of finite elements on curves. In a forthcoming paper, we will provide rigorous arguments to substantiate the observations made here. We intend to explore how such a parameterization is useful in the context of problems with curved domains and

DOI: 10.1002/nme



(a) (b)

Figure 21. Computing the elastic deformation of a Timoshenko beam whose mid-curve is the letter 'S'. Figure (a) shows the mid-curve immersed in a triangulation that is a refinement of the one shown in Figure 3(a). The curve consists of three smooth component curves: two arcs of a circle and a line segment each of which is shown in different colors. As described in Section 5.2, small vertex adjustments are performed near each corner of the curve to ensure that there is a vertex coincident with each corner. The edges identified for parameterization of the curve are shown in green in (b). It also shows the deformed equilibrium configuration of the mid-curve in the presence of a constant body force **B** when the cross-section of the body at the indicated point on the line segment is rigidly fixed. The body force is scaled to produce a visually pronounced deformation.

high-order immersed boundary methods. We can imagine a scenario in which, at least in principle, the same mesh is used for computing solutions on an evolving domain. The added advantage in such a case is that the sparsity structure of some of the matrices involved (e.g., mass, stiffness) can be left unaltered as the domain evolves.

To make the parameterization method more robust, it would be important to know how small the local mesh size needs to be for the method to generate a one-to-one and onto parameterization. One possibility would be to formulate an efficient method to test whether the mesh size is small enough on the fly. Alternatively, it could be possible to obtain easy-to-compute analytical bounds on the local mesh size for a good parameterization. In this case, a non-conservative bound would also avoid overly refined meshes.

We are also investigating whether this method generalizes to one for parameterizing surfaces immersed in tetrahedral meshes. This would be a valuable tool, especially because there is no canonical parameterization for surfaces.

APPENDIX A: COMPUTING THE SIGNED DISTANCE AND THE CLOSEST-POINT PROJECTION

An important question with our parameterization method is how to compute the closest-point projection π for a connected C^2-regular boundary Γ. As discussed in Section 6, for the construction of finite element spaces on Γ, we also need to compute its derivatives. While the parameterization we compute for Γ is independent of how Γ is represented, the way to compute π and its derivatives is particular for each curve representation. We address this question for two commonly encountered

DOI: 10.1002/nme

representations: when Γ is given parametrically (e.g., splines) or implicitly as the zero level set of a function (common in immersed boundary methods).

Throughout this section, we denote the unit tangent and unit normal to Γ by $\hat{T}$ and $\hat{N}$, respectively. The signed distance function to Γ is denoted by ϕ. We assume

(i) an orientation for Γ such that $\hat{N}$ is parallel to $\nabla\phi$ at each point on the curve, and
(ii) that $\{\hat{T}, \hat{N}\}$ is a positively oriented basis for $\mathbb{R}^2$ at each point in Γ.

The signed curvature of Γ is denoted by κ. Note that the sign of κ depends on the orientation assumed for Γ (cf. [22]).

Let $p \in \mathbb{R}^2$ with Cartesian coordinates (x_p, y_p) be a point that is close to Γ. Then, we have (see [10])

$$\pi(p) = p - \phi\nabla\phi(p), \tag{A1}$$

$$\nabla\phi(p) = \hat{N}(\pi(p)). \tag{A2}$$

Equations (A1) and (A2) imply that

$$(p - \pi(p)) \cdot \hat{T}(\pi(p)) = 0. \tag{A3}$$

Differentiating (A1) with respect to p shows that

$$\nabla\pi(p) = \mathbf{I} - \nabla\phi \otimes \nabla\phi(p) - \phi\nabla\nabla\phi(p), \tag{A4}$$

where $\mathbf{I}$ is the identity tensor on $\mathbb{R}^{2\times 2}$ and $\otimes$ is the usual tensor product. Noting from (A4) that

$$\nabla\pi(p) \cdot \hat{N}(\pi(p)) = 0,$$

it is straightforward to show that (see Section A.3)

$$\nabla\pi(p) = \frac{\hat{T}(\pi(p)) \otimes \hat{T}(\pi(p))}{1 - \phi(p)\kappa(\pi(p))}. \tag{A5}$$

The term $(1 - \phi(p)\ \kappa(\pi(p)))$ in the denominator in (A5) explicitly demonstrates that p needs to be close to Γ.

A.1. Parametric representation for Γ

Suppose that Γ is given by a regular parametrization in a set of Cartesian coordinates as

$$\Gamma = \{(x(t), y(t)) : x, y \in C^2(I, \mathbb{R}), I \subset \mathbb{R}\}. \tag{A6}$$

Note that since the parameterization is assumed to be regular,

$$(x')^2 + (y')^2 > 0 \quad \text{on } I.$$

Let the point $\pi(p)$ have coordinates $(x(\xi), y(\xi))$, where ξ is to be determined. From (A6), the unit tangent and normal to Γ at $\pi(p)$ are computed as

$$\hat{T}(\pi(p)) = \frac{(x'(\xi), y'(\xi))}{\sqrt{x'(\xi)^2 + y'(\xi)^2}},$$

$$\hat{N}(\pi(p)) = \frac{(-y'(\xi), x'(\xi))}{\sqrt{x'(\xi)^2 + y'(\xi)^2}},$$

where we have assumed that Γ has been oriented to satisfy (A2). From (A3), we get

$$(x_p - x(\xi)) \cdot x'(\xi) + (y_p - y(\xi)) \cdot y'(\xi) = 0. \tag{A7}$$

DOI: 10.1002/nme

To find the closest-point projection $\pi(p)$, we solve (A7) for the unknown ξ. The solution may not be unique, but if p is sufficiently close to Γ, there will be a unique minimizer of $|\phi(p)|$. We can then compute $\pi(p)=(x(\xi), y(\xi))$ while $\phi(p)$ and $\nabla\phi(p)$ follow as

$$\begin{aligned}\phi(p) &= (p-\pi(p))\cdot\hat{N}(\pi(p)),\\ &= \frac{(y_p-y(\xi))x'(\xi)-(x_p-x(\xi))y'(\xi)}{\sqrt{x'(\xi)^2+y'(\xi)^2}}.\end{aligned} \tag{A8}$$

$$\nabla\phi(p) = \frac{(-y'(\xi), x'(\xi))}{\sqrt{x'(\xi)^2+y'(\xi)^2}}. \tag{A9}$$

Using (A6), the signed curvature of Γ at $\pi(p)$ is computed as

$$\kappa(\pi(p)) = \left.\frac{x'y''-y'x''}{(x'^2+y'^2)^{3/2}}\right|_{\xi}. \tag{A10}$$

Finally, $\nabla\pi(p)$ can be computed from (A5). Similar calculations also apply to the case when Γ is locally given by parametric equations.

An important example of parametric curves is splines, which are ubiquitous in computer graphics and engineering applications. References [14, 15, 23] and the ones therein investigate optimizations specific to these curves for efficient and robust computation of $\pi(p)$.

A.2. Implicit representation for Γ

Next, suppose that Γ is given as the zero level set of a function $\psi\in C^2(\mathbb{R}^2, \mathbb{R})$, i.e. $\Gamma=\psi^{-1}(0)$. We assume that

$$\begin{aligned}&|\nabla\psi| > 0 \quad \text{on } \psi^{-1}(0),\\ \text{and} \quad &\operatorname{sign}(\psi) = \operatorname{sign}(\phi) \quad \text{on } \mathbb{R}^2.\end{aligned} \tag{A11}$$

Let $\pi(p)$ have Cartesian coordinates $(x_\star, y_\star)$. We can compute the unit tangent and normal to Γ at the point $\pi(p)$ as

$$\hat{T}(\pi(p)) = \left.\frac{(\psi_y, -\psi_x)}{\sqrt{\psi_x^2+\psi_y^2}}\right|_{(x_\star, y_\star)},$$

$$\hat{N}(\pi(p)) = \left.\frac{(\psi_x, \psi_y)}{\sqrt{\psi_x^2+\psi_y^2}}\right|_{(x_\star, y_\star)},$$

where we have denoted partial derivatives by using subscripts. Since ψ has the same sign as ϕ, $\hat{N}$ is parallel to $\nabla\psi$ at each point on Γ. Using the above expression for the unit normal in (A1), we get

$$x_\star = x_p - \phi(p)\left.\frac{\psi_x}{\sqrt{\psi_x^2+\psi_y^2}}\right|_{(x_\star, y_\star)}, \tag{A12}$$

$$y_\star = y_p - \phi(p)\left.\frac{\psi_y}{\sqrt{\psi_x^2+\psi_y^2}}\right|_{(x_\star, y_\star)}. \tag{A13}$$

DOI: 10.1002/nme

Along with the fact that $\psi(x_\star, y_\star)=0$, (A12) and (A13) yield a system of three coupled equations which can be solved to compute $x_\star$, $y_\star$ and $\phi(p)$. From (A11), the signed curvature of Γ at the point $\pi(p)$ is computed as

$$\kappa(\pi(p))=\left.\frac{2\psi_{xy}\psi_x\psi_y-\psi_{xx}\psi_y^2-\psi_{yy}\psi_x^2}{(\psi_x^2+\psi_y^2)^{3/2}}\right|_{(x_\star,y_\star)}. \tag{A14}$$

Finally, $\nabla\pi(p)$ can be computed from (A5).

A.3. Computation of the gradient of the closest-point projection

For completeness, we prove (A5) next. This is precisely stated in the following proposition; see also [24, lemmas 14.16, 14.17]).

Proposition
Let $\Gamma\subset\mathbb{R}^2$ be a C^2-regular boundary with unit tangent $\hat{T}$, signed curvature κ, signed distance ϕ and closest-point projection π. If $p\in\mathbb{R}^2$ is sufficiently close to Γ, then

$$\nabla\pi(p)=\frac{\hat{T}(\pi(p))\otimes\hat{T}(\pi(p))}{1-\phi(p)\kappa(\pi(p))}. \tag{A15}$$

Proof
We know that (see [10])

$$\nabla\phi(p)=\hat{N}(\pi(p))\Rightarrow\nabla\phi\cdot\nabla\phi(p)=1. \tag{A16}$$

Differentiating (A16) with respect to p, we get

$$\nabla\nabla\phi\cdot\nabla\phi(p)=0. \tag{A17}$$

For convenience, let $\hat{t}=\hat{T}(\pi(p))$. The symmetry of $\nabla\nabla\phi$ and (A17) implies that

$$\nabla\nabla\phi(p)=(\hat{t}\cdot\nabla\nabla\phi(p)\cdot\hat{t})\hat{t}\otimes\hat{t}. \tag{A18}$$

Using (A16) and denoting the arc-length parameter of Γ by s, we get

$$\begin{aligned}\hat{t}\cdot\nabla\nabla\phi(p)\cdot\hat{t}&=\left.\frac{\mathrm{d}\hat{N}}{\mathrm{d}s}\right|_{\pi(p)}\cdot\nabla\pi(p)\cdot\hat{t}\\&=-\kappa(\pi(p))\hat{t}\cdot\nabla\pi(p)\cdot\hat{t}.\end{aligned} \tag{A19}$$

Since p is close to Γ, we have

$$\pi(p)=p-\phi\nabla\phi(p). \tag{A20}$$

Differentiating (A20) with respect to p, we get

$$\nabla\pi(p)=\mathbf{I}-\nabla\phi\otimes\nabla\phi(p)-\phi\nabla\nabla\phi(p), \tag{A21}$$

where $\mathbf{I}$ is the unit tensor in $\mathbb{R}^{2\times2}$. Using (A16) and (A17) in (A21) shows that

$$\nabla\pi(p)\cdot\nabla\phi(p)=0. \tag{A22}$$

Since $\nabla\pi$ is a symmetric tensor (see (A21)), (A22) implies that

$$\nabla\pi(p)=(\hat{t}\cdot\nabla\pi(p)\cdot\hat{t})\hat{t}\otimes\hat{t}. \tag{A23}$$

It remains to compute $\hat{t}\cdot\nabla\pi(p)\cdot\hat{t}$. From (A16), (A19) and (A21), we get

$$\hat{t}\cdot\nabla\pi(p)\cdot\hat{t}=1+\phi(p)\kappa(\pi(p))\hat{t}\cdot\nabla\pi(p)\cdot\hat{t}. \tag{A24}$$

DOI: 10.1002/nme

Since $|\phi(p)\kappa(\pi(p))| < 1$ (from p being close to Γ), (A24) shows that

$$\hat{t} \cdot \nabla \pi(p) \cdot \hat{t} = \frac{1}{1 - \phi(p)\ \kappa(\pi(p))}. \tag{A25}$$

Equation (A15) follows from (A23) and (A25). □

ACKNOWLEDGEMENTS

R. R. is grateful for the support provided by a Stanford Graduate Fellowship at the Stanford University. We thank the anonymous reviewers, whose comments helped to improve the manuscript. This work is supported by ONR Young Investigator Award under grant number N000140810852, NSF Career Award under grant CMMI-0747089 and Department of the Army Research under grant W911NF-07-2-0027.

REFERENCES

1. Lew A, Buscaglia G. A discontinuous-Galerkin-based immersed boundary method. *International Journal for Numerical Methods in Engineering* 2008; **76**(4):427–454.
2. Rangarajan R, Lew A, Buscaglia G. A discontinuous-Galerkin-based immersed boundary method with non-homogeneous boundary conditions and its application to elasticity. *Computer Methods in Applied Mechanics and Engineering* 2009; **198**(17–20):1513–1534.
3. Strang G, Berger A. The change in solution due to change in domain. In *Proceedings of Symposia in Pure Mathematics*, *XXIII*, Spencer DC (ed.), vol. 23. AMS: Providence, RI, 1973; 199–205.
4. Thomeé V. Polygonal approximation of the domain in Dirichlet's problem. *Journal of the Institute of Mathematics and its Applications* 1973; **11**:33–34.
5. Ruuth S, Merriman B. A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics* 2008; **227**(3):1943–1961.
6. Macdonald C, Ruuth S. The implicit Closest Point Method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing* 2009; **31**:4330–4350.
7. Mémoli F, Sapiro G, Osher S. Solving variational problems and partial differential equations mapping into general target manifolds. *Journal of Computational Physics* 2004; **195**(1):263–292.
8. Bertalmío M, Mémoli F, Cheng L, Sapiro G, Osher S. Variational problems and partial differential equations on implicit surfaces: bye bye triangulated surfaces? *Geometric Level Set Methods in Imaging*, *Vision*, *and Graphics*. Springer: New York, 2003; 381–397.
9. Gonzalez M, Goldschmit M. Inverse geometry heat transfer problem based on a radial basis functions geometry representation. *International Journal for Numerical Methods in Engineering* 2005; **65**(8):1243–1268.
10. Henry D, Hale J, Pereira A. *Perturbation of the Boundary in Boundary-value Problems of Partial Differential Equations*. Cambridge University Press: Cambridge, 2005.
11. Brenner S, Scott L. *The Mathematical Theory of Finite Element Methods*. Springer: Berlin, 2002.
12. Krantz S, Parks H. *The Geometry of Domains in Space*. Birkhäuser: Basel, 1999.
13. Krantz S, Parks H. *The Implicit Function Theorem*. Birkhäuser: Basel, 2003.
14. Wang H, Kearney J, Atkinson K. Robust and efficient computation of the closest point on a spline curve. *Proceedings of the Fifth International Conference on Curves and Surfaces*, Saint-Malo, 2002; 397–406.
15. Ma Y, Hewitt W. Point inversion and projection for NURBS curve and surface: control polygon approach. *Computer Aided Geometric Design* 2003; **20**(2):79–99.
16. Hu S, Wallner J. A second order algorithm for orthogonal projection onto curves and surfaces. *Computer Aided Geometric Design* 2005; **22**(3):251–260.
17. De Berg M, Cheong O, Van Kreveld M, Overmars M. *Computational Geometry*: *Algorithms and Applications*. Springer: New York, 2008.
18. Frey P, George P. *Mesh Generation*: *Application to Finite Elements*. ISTE: London, 2008.
19. Bern M, Eppstein D, Gilbert J. Provably good mesh generation. *Journal of Computer and System Sciences* 1994; **48**(3):384–409.
20. Guillemin V, Pollack A. *Differential Topology*. Prentice-Hall: Englewood Cliffs, NJ, 1974.
21. Chapelle D, Bathe K. *The Finite Element Analysis of Shells*: *Fundamentals*. Springer: Berlin, 2003.
22. Do Carmo M. *Differential Geometry of Curves and Surfaces*. Prentice-Hall: Englewood Cliffs, NJ, 1976.
23. Selimovic I. Improved algorithms for the projection of points on NURBS curves and surfaces. *Computer Aided Geometric Design* 2006; **23**(5):439–445.
24. Gilbarg D, Trudinger N. *Elliptic Partial Differential Equations of Second Order*. Springer: Berlin, 2001.

DOI: 10.1002/nme