

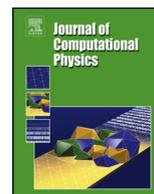


ELSEVIER

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp



Shape optimization of self-avoiding curves



Shawn W. Walker

Department of Mathematics, Center for Computation and Technology (CCT), Louisiana State University, Baton Rouge, LA 70803, USA

ARTICLE INFO

Article history:

Received 1 June 2015

Received in revised form 22 December 2015

Accepted 3 February 2016

Available online 6 February 2016

Keywords:

Self-avoiding curve

Self-contact

Proximity

Shape optimization

Semi-flexible polymers

Knots

ABSTRACT

This paper presents a softened notion of proximity (or self-avoidance) for curves. We then derive a sensitivity result, based on shape differential calculus, for the proximity. This is combined with a gradient-based optimization approach to compute three-dimensional, parameterized curves that minimize the sum of an elastic (bending) energy and a proximity energy that maintains self-avoidance by a penalization technique. Minimizers are computed by a sequential-quadratic-programming (SQP) method where the bending energy and proximity energy are approximated by a finite element method.

We then apply this method to two problems. First, we simulate adsorbed polymer strands that are constrained to be bound to a surface and be (locally) inextensible. This is a basic model of semi-flexible polymers adsorbed onto a surface (a current topic in material science). Several examples of minimizing curve shapes on a variety of surfaces are shown. An advantage of the method is that it can be much faster than using molecular dynamics for simulating polymer strands on surfaces. Second, we apply our proximity penalization to the computation of ideal knots. We present a heuristic scheme, utilizing the SQP method above, for minimizing rope-length and apply it in the case of the trefoil knot. Applications of this method could be for generating good initial guesses to a more accurate (but expensive) knot-tightening algorithm.

© 2016 Elsevier Inc. All rights reserved.

1. Introduction

Self-assembly of micro structures is a growing research area with applications in basic science and material design [22, 37,32]. In particular, polymers confined to curved surfaces have great scientific interest. The packing of strands of DNA onto a protein complex, as well as packaging DNA into a small volume to fit inside a cell, is of fundamental importance in biology [35,41,52]. In the design of materials, micro-scale patterned surfaces can be used to create material components with novel optical, electronic and magnetic properties [43]. Moreover, elastic polymer chains can be forced into regular patterns when they are energetically bound to a deformable membrane [33,61,62,53]. Therefore, modeling and simulating the equilibrium configurations of semi-flexible polymer strands on surfaces is important for basic physics understanding and developing technological applications [12,1,46].

A crucial driving force of these patterns is self-contact [19,20], i.e. an object's matter cannot overlap itself. A long polymer chain that is forced to reside on a closed, bounded surface must interact with *itself* when obtaining its equilibrium configuration. Hence, the presence of self-contact (or self-avoidance) in these systems plays a fundamental role in their equilibrium states and dynamic evolutions [50], so we must account for it when modeling these systems. Multiple characterizations of self-contact have been developed in the literature [29,30,19] (all in the context of knots).

E-mail address: walker@math.lsu.edu.

This paper presents a smoothed notion of self-avoidance for curves that is suitable for gradient-based optimization methods. Our method is based on a modified version of the *global radius of curvature* [19,20,31,42,47]. This concept possesses some very nice analytical properties [42,47] and was used to prove the existence of optimal shapes of “thick” knots [19,20,24]. But it is certainly not limited to this one application. We demonstrate our method by applying it to two problems.

The first problem is on the adsorption of semi-flexible polymers onto surfaces, which has been done in several research works [35,41,52,59,61,62,53] mostly using molecular dynamics and Lennard-Jones potentials. We propose to simulate the equilibrium configuration of inextensible elastic curves that are bound to a surface and satisfy the no-penetration condition (i.e. the elastic curve is self-avoiding). By no-penetration, we assume the curve is surrounded by a *tubular neighborhood* that does not self-intersect (see Section 2.1.5), which is completely inline with modeling “thick” polymer strands.

We use an energetic continuum approach, i.e. the curves are parameterized 1-D manifolds (not beads on a string) that minimize the sum of an elastic energy and a “proximity” (penalized) energy that enforces the no-penetration condition. The proximity functional (see Section 2.3) softens the effect of self-contact, and as far as we know is new. Gradient information is computed via shape differentiation. We then discretize the problem and give a discrete sequential-quadratic-programming (SQP) method for finding local minimizers of the energy. To the best of our knowledge, this is a novel scheme, but we point out another constrained gradient scheme in [3]. As for efficiency, our method takes a few hundred to about a thousand iteration steps. A molecular dynamics simulation typically takes on the order of a few million steps. Hence, our method should be faster than a direct molecular dynamics simulation.

The second problem is on computing ideal knots. We present a heuristic algorithm for minimizing rope-length of closed curves (i.e. for approximating ideal knots), that utilizes the SQP method above, and apply it in the case of the trefoil knot. One application of the method could be for generating good initial guesses to a more accurate (but expensive) knot-tightening algorithm. Furthermore, it may be useful for probing optimal configurations of more complicated knots.

The paper is organized as follows. Section 2 reviews the concept of global radius of curvature, then we present our penalized approach for preventing self-contact. Section 3 shows how we model the equilibrium shape of semi-flexible curves as minimizers of an energy functional which includes a term to penalize self-intersection; Section 4 describes a finite element method and optimization algorithm for computing minimizers, including a heuristic method for minimizing rope-length (Section 4.7). Several numerical results are shown in Section 5, followed with some concluding remarks.

2. Measuring proximity

In order to model and simulate curves that do not self-intersect, we must quantify self-contact for “thick” curves (see Definition 1). We start by reviewing the concept of global radius of curvature [19,20,31]. Then, we present a softened version of the global radius of curvature (i.e. the “proximity”) that is useful for numerics and optimization. Throughout the paper, we make the following assumption.

Assumption 1. Let $\Sigma \subset \mathbb{R}^3$ be a smooth (open or closed) curve in \mathbb{R}^3 , with length L , and parameterized by $\alpha : [0, 1] \rightarrow \mathbb{R}^3$ (unless otherwise stated). Moreover, assume that Σ is *simple*, i.e. it has no self-intersections.

2.1. Global radius of curvature

2.1.1. Circumradius

Let \mathcal{R} be the circumradius of three non-collinear points $\mathbf{x}, \mathbf{y}, \mathbf{z}$ in \mathbb{R}^3 :

$$\mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \frac{|\mathbf{x} - \mathbf{y}||\mathbf{y} - \mathbf{z}||\mathbf{z} - \mathbf{x}|}{4\mathcal{A}(\mathbf{x}, \mathbf{y}, \mathbf{z})}, \quad (1)$$

where $|\cdot|$ is the Euclidean norm and $\mathcal{A}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is the area of a triangle with vertices $\mathbf{x}, \mathbf{y}, \mathbf{z}$. Thus, $\mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \geq 0$ and $\mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}) := \infty$ if $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are collinear. Any three non-collinear points define a unique circumcircle and a unique circumsphere that contains the circumcircle as a great circle.

If Σ is a simple smooth curve, then the domain of \mathcal{R} can be extended to any triple of points on Σ by continuous limits. Suppose that $\mathbf{x} = \alpha(s)$, $\mathbf{y} = \alpha(t)$, $\mathbf{z} = \alpha(u)$ are three distinct points on Σ . Then one can show that [19]

$$\mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{y}) := \lim_{u \rightarrow t} \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \frac{|\mathbf{x} - \mathbf{y}|}{2|\sin \theta_{\mathbf{xy}}|}, \quad (2)$$

where $\theta_{\mathbf{xy}}$ is the angle between the vector $\mathbf{x} - \mathbf{y} \neq \mathbf{0}$ and the tangent vector $\boldsymbol{\tau}(\mathbf{y})$ to Σ at \mathbf{y} . Thus, the limit circumcircle is tangent to Σ at \mathbf{y} and passes through \mathbf{x} . We also have that

$$\mathcal{R}(\mathbf{x}, \mathbf{x}, \mathbf{x}) := \lim_{t, u \rightarrow s} \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \rho(\mathbf{x}), \quad (3)$$

where $\rho(\mathbf{x})$ is the standard local radius of curvature of Σ at \mathbf{x} , i.e. the limit circumcircle is the *osculating* circle of Σ at \mathbf{x} .

For any curve Σ , we define the *global radius of curvature* $\rho_G(\mathbf{x})$ at each point \mathbf{x} in Σ by [19,20]

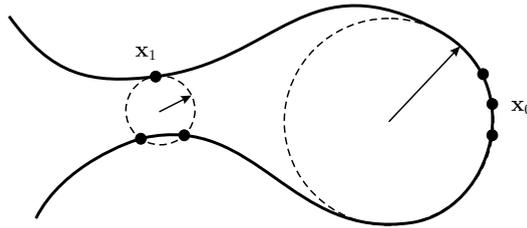


Fig. 1. Illustration of the global radius of curvature. We compute $\rho_G(\mathbf{x}_0)$, $\rho_G(\mathbf{x}_1)$ at two points \mathbf{x}_0 , \mathbf{x}_1 on the thick black curve Σ by doing the minimization in (5). Two circumcircles are shown for different values of \mathbf{y} , \mathbf{z} . For this particular curve, $\rho_G(\mathbf{x}_0)$ is achieved by the local radius of curvature, i.e. $\rho_G(\mathbf{x}_0) = \rho(\mathbf{x}_0)$. But $\rho_G(\mathbf{x}_1) < \rho(\mathbf{x}_1)$ where the curve comes near to itself.

$$\rho_G(\mathbf{x}) = \inf_{\substack{\mathbf{y}, \mathbf{z} \in \Sigma \setminus \{\mathbf{x}\} \\ \mathbf{y} \neq \mathbf{z}}} \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}). \tag{4}$$

When Σ is simple and smooth, the infimum can be replaced by a minimum, i.e.

$$\rho_G(\mathbf{x}) = \min_{\mathbf{y}, \mathbf{z} \in \Sigma} \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}). \tag{5}$$

Moreover, $\rho_G : \Sigma \rightarrow \mathbb{R}_+$ is a continuous function on Σ . This follows by continuity of $\mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z})$ and (2), (3). See Fig. 1 for an illustration.

The function ρ_G generalizes curvature to a global concept. Clearly, we have

$$0 \leq \rho_G(\mathbf{x}) \leq \rho(\mathbf{x}), \quad \forall \mathbf{x} \in \Sigma. \tag{6}$$

Note that ρ_G is infinite when Σ is a straight line. Conversely, if there is a point \mathbf{x}_0 in Σ for which $\rho_G(\mathbf{x}_0) = \infty$, then $\rho_G(\mathbf{x}) = \infty$ for all \mathbf{x} in Σ and Σ is a straight line (because ρ_G is a continuous function).

We can characterize the global radius of curvature geometrically. Depending on the point \mathbf{x} in Σ , the number $\rho_G(\mathbf{x})$ may be the local radius of curvature, or the strictly smaller radius of a circle containing \mathbf{x} and another distinct point \mathbf{y} in Σ where the circle is tangent. Thus, the minimization in (5) can be done with the restriction $\mathbf{y} = \mathbf{z}$. See [19] for more details.

2.1.2. Point tangent radius

Let $\boldsymbol{\tau}(\mathbf{y})$ be the unit tangent vector of Σ at \mathbf{y} in Σ . Then, by taking appropriate limits, one can show that the point-tangent radius is given by

$$\lim_{\mathbf{z} \rightarrow \mathbf{y}} \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{y}) = \frac{|\mathbf{x} - \mathbf{y}|}{2 \left| \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \times \boldsymbol{\tau}(\mathbf{y}) \right|} = \frac{|\mathbf{x} - \mathbf{y}|}{2 \left[1 - \left(\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right)^2 \right]^{1/2}},$$

i.e. it is the radius of a circle that passes through \mathbf{x} and \mathbf{y} and is tangent to Σ at \mathbf{y} . With this, we define the point tangent function

$$\mathcal{R}_{\text{pt}}(\mathbf{a}, \boldsymbol{\tau}) := \frac{|\mathbf{a}|}{2 \left[1 - \left(\frac{\mathbf{a}}{|\mathbf{a}|} \cdot \boldsymbol{\tau} \right)^2 \right]^{1/2}}. \tag{7}$$

Thus, the point tangent radius is related to the circumradius by

$$\mathcal{R}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = \mathcal{R}(\mathbf{x}, \mathbf{y}, \mathbf{y}). \tag{8}$$

Note that $\mathcal{R}_{\text{pt}}(\mathbf{a}, \boldsymbol{\tau}(\mathbf{y}))$ is an even function with respect to the argument \mathbf{a} .

Hence, we can compute the global radius of curvature (at a point \mathbf{x} in Σ) by computing

$$\rho_G(\mathbf{x}) = \min_{\mathbf{y} \in \Sigma} \mathcal{R}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})), \tag{9}$$

instead of the double minimization in (5). Alternative radius functions have been proposed for defining other global radius of curvature functions. In [44], they show that the point tangent radius gives the minimal value amongst all obvious choices of radius function.

2.1.3. Basic properties

We now list properties of the point tangent radius and the global radius of curvature (see [19,31,44] for details). Basic limiting arguments show that

$$\lim_{\mathbf{y} \rightarrow \mathbf{x}} \mathcal{R}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = \frac{1}{|\boldsymbol{\kappa}(\mathbf{x})|} = \rho(\mathbf{x}), \tag{10}$$

where $\boldsymbol{\kappa}$ is the vector curvature of Σ at \mathbf{x} . For a fixed \mathbf{x} in Σ , the associated minimizer in (9) satisfies one of the following two conditions [31,44]:

$$\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) = 0, \text{ (type 1) OR } |\boldsymbol{\kappa}(\mathbf{y})| \mathcal{R}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = 1, \text{ (type 2)}, \tag{11}$$

i.e. either $\mathbf{x} - \mathbf{y}$ is orthogonal to Σ at \mathbf{y} or the point tangent circle has the same radius as $\rho(\mathbf{y})$. For example, $\rho_G(\mathbf{x})$ may be realized by the local radius of curvature. This result follows by basic calculus and limits [44].

2.1.4. Minimum global radius of curvature

For any curve Σ , we associate the number

$$\underline{\rho}_G(\Sigma) = \inf_{\mathbf{x} \in \Sigma} \rho_G(\mathbf{x}). \tag{12}$$

If Σ is simple and smooth, then ρ_G is continuous, so we can replace the infimum by a min:

$$\underline{\rho}_G(\Sigma) = \min_{\mathbf{x} \in \Sigma} \rho_G(\mathbf{x}). \tag{13}$$

The following geometric interpretation of $\underline{\rho}_G(\Sigma)$ is given in [19]. Any spherical shell of radius less than $\underline{\rho}_G(\Sigma)$ cannot intersect Σ in three or more points (counting tangency points twice). Basically, a ball of radius less than $\underline{\rho}_G(\Sigma)$ cannot become “stuck” on Σ , because there is always “room” for it to pass through the interstices of the curve Σ .

Put differently, $\underline{\rho}_G(\Sigma)$ yields information about the *closest proximity* of Σ with itself. Either $\underline{\rho}_G(\Sigma)$ is the minimum local radius of curvature or the strictly smaller radius of a sphere, that contains no portion of the curve Σ in its interior, which is tangent to the curve at two *diametrically opposite* points \mathbf{x} and \mathbf{y} [19,31,44]. At these points we have the symmetry property

$$\underline{\rho}_G(\Sigma) = \rho_G(\mathbf{x}) = \mathcal{R}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = \mathcal{R}_{\text{pt}}(\mathbf{y} - \mathbf{x}, \boldsymbol{\tau}(\mathbf{x})) = \rho_G(\mathbf{y}). \tag{14}$$

Let \mathcal{U}_{dc} be the set of pairs of points (\mathbf{x}, \mathbf{y}) in Σ such that $\mathbf{x} \neq \mathbf{y}$ and $\mathbf{x} - \mathbf{y}$ is orthogonal to $\boldsymbol{\tau}(\mathbf{x})$ and $\boldsymbol{\tau}(\mathbf{y})$ (i.e. the set of double critical points). One way to characterize the minimum global radius of curvature is [9,34,19]

$$\underline{\rho}_G(\Sigma) = \min \left\{ \rho(\mathbf{x}), \min_{(\mathbf{x}, \mathbf{y}) \in \mathcal{U}_{\text{dc}}} \frac{|\mathbf{x} - \mathbf{y}|}{2} \right\}. \tag{15}$$

We also have that $\underline{\rho}_G(\Sigma)$ is continuous on the space of $C^{1,1}$ curves, i.e. $\underline{\rho}_G(\Sigma)$ is continuous at any simple curve Σ that is not a straight line segment (otherwise it is infinite). Moreover, $\underline{\rho}_G(\Sigma_k)$ tends to zero for any sequence of simple smooth curves $\{\Sigma_k\}$ that approaches a self-intersecting curve [20,47].

2.1.5. Curve thickness

Let $\mathcal{T}(\Sigma)$ be the “tube” surrounding Σ defined as

$$\mathcal{T}(\Sigma) = \bigcup_{\mathbf{x} \in \Sigma} \mathcal{D}(\mathbf{x}, r), \tag{16}$$

where $\mathcal{D}(\mathbf{x}, r)$ is the open disk of radius $r > 0$ centered at \mathbf{x} such that the tangent vector $\boldsymbol{\tau}(\mathbf{x})$ of Σ is orthogonal to $\mathcal{D}(\mathbf{x}, r)$. The maximum value of r such that all of the disks $\{\mathcal{D}(\mathbf{x}, r)\}$ remain pairwise disjoint is precisely $\underline{\rho}_G(\Sigma)$ [19]. Thus, $\underline{\rho}_G(\Sigma)$ is the radius of the thickest (open) smooth tube that can be centered on Σ , without self-intersections. Along these lines, we define the notion of a “thick” curve.

Definition 1 (Curve thickness). Let Σ be a $C^{1,1}$ curve. We say that Σ is a **thick** curve if there exists a tubular neighborhood $\mathcal{T}(\Sigma)$, with positive thickness, such that $\mathcal{T}(\Sigma)$ does not self-intersect. Moreover, we say Σ has **thickness** $d > 0$ if d has the maximal value possible such that $\mathcal{T}(\Sigma)$ does not self-intersect when the disks in (16) have radius less than or equal to d .

In our application, the curve will have a prescribed thickness (e.g. the thickness of the polymer) that our algorithm will maintain in a penalized sense (Section 4.6).

Previous researchers have introduced and studied the concept of an ideal knot [29,30,19]. Let \mathcal{K} be a set of simple, smooth, closed curves of a specified topological knot class with fixed length L . Now consider the maximization: find Σ^* in \mathcal{K} such that

$$\underline{\rho}_G(\Sigma^*) = \max_{\Sigma \in \mathcal{K}} \underline{\rho}_G(\Sigma). \tag{17}$$

The optimizer Σ^* in \mathcal{K} is called an *ideal knot*, i.e. Σ^* is the thickest curve of fixed length that can be tied into a given knot. Existence of $C^{1,1}$ optimal solutions is described in [20,42]. It can be shown [19] that a necessary condition of the optimal solution Σ^* is that

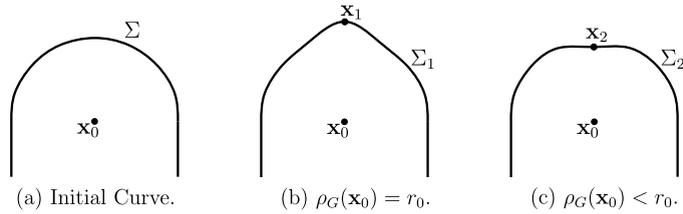


Fig. 2. Illustration of the non-smoothness of ρ_G . (a) A portion of a curve $\Sigma \subset \mathbb{R}^3$ is shown, where the point \mathbf{x}_0 lies on a straight part of the curve that is coming out of the page. Note: $\rho_G(\mathbf{x}_0) = r_0$. (b) The curve is deformed slightly but does not change $\rho_G(\mathbf{x}_0)$. (c) A different deformation does change $\rho_G(\mathbf{x}_0)$.

$$\rho_G(\mathbf{x}) = \underline{\rho}_G(\Sigma^*), \quad \forall \mathbf{x} \in \Sigma^*, \tag{18}$$

i.e. the global radius of curvature is *constant* and minimal over the entire curve.

2.2. Proximity

It is clear that the global radius of curvature ρ_G provides a notion of proximity (or closeness) for a thick curve (Definition 1) with respect to itself. However, ρ_G is not a particularly smooth function for the following reasons. First, computing a minimum (for a fixed \mathbf{x}) in (9) is not a smooth process. For example, let $g(t) = \min(f_1(t), f_2(t))$ for all t where f_1 and f_2 are C^∞ functions, and note that g can be rewritten as

$$g(t) = \frac{f_1(t) + f_2(t)}{2} - \frac{|f_1(t) - f_2(t)|}{2}.$$

The presence of the absolute value means g cannot be better than continuous. Second, one can construct a smooth curve Σ such that an infinite number of points \mathbf{y} realize the minimum in (9) for a fixed \mathbf{x} . This can lead to abrupt changes in ρ_G when smoothly deforming Σ (Fig. 2 shows an example).

In Fig. 2(a), part of a smooth closed curve Σ (embedded in three dimensions) is shown. It consists of a circular arc connected to two straight pieces with another part of the curve “coming out of the page” along the axis of the arc (i.e. the dot at \mathbf{x}_0). We consider \mathbf{x}_0 to be a point on the vertical piece that is in the plane of the circular arc. Computing $\rho_G(\mathbf{x}_0)$, we see that every point \mathbf{y} on the circular arc realizes the minimum in (9). Since it is a circular arc, we see that $\rho_G(\mathbf{x}_0) = r_0$ where r_0 is the radius of the arc. In Fig. 2(b), part of the circular arc has been deformed into a new shape Σ_1 . In this case, the points on Σ_1 that achieve the minimum in (9) are the points on the circular arc that are unaffected by the deformation. In fact, $\mathcal{R}_{pt}(\mathbf{x}_0 - \mathbf{x}_1, \boldsymbol{\tau}(\mathbf{x}_1)) > r_0$ so \mathbf{x}_1 does not achieve the minimum. Fig. 2(c), shows a different deformation taking Σ into Σ_2 . Now we have that $\rho_G(\mathbf{x}_0) = r_2 < r_0$ because $\mathcal{R}_{pt}(\mathbf{x}_0 - \mathbf{x}_2, \boldsymbol{\tau}(\mathbf{x}_2)) = r_2 < r_0$. Thus, ρ_G can change abruptly depending on the particular deformation.

Indeed, let $\Sigma \subset \mathbb{R}^3$ be a given curve and suppose $\mathbf{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a smooth vector-valued function. We can define a parameterized family of curves by $\Sigma_t = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \mathbf{y} + t\mathbf{V}(\mathbf{y}), \forall \mathbf{y} \in \Sigma\}$. Let \mathbf{x}_0 be a fixed point in Σ and let \mathbf{x}_t be the corresponding point in Σ_t . Now consider the function $f(t) = \rho_G(\mathbf{x}_t, \Sigma_t)$, where we include the second argument Σ_t to emphasize that ρ_G is computed on the curve Σ_t . Even for a smooth \mathbf{V} , $f(t)$ may not even be differentiable (in the classical sense). Because of this non-smoothness, and the fact that ρ_G is a non-local function, computing the shapes of ideal knots is extremely difficult. Several researchers [39,31,37,44,3,40,2] attacked this problem by a brute-force optimization approach, e.g. a Monte-Carlo/molecular dynamics type of procedure was used. However, this is computationally expensive especially when the object being optimized is a smooth 1-D curve. It is well-known that ideal knots are regular $C^{1,1}$ curves, and a molecular dynamics approach does *not* take advantage of this.

On the other hand, ensuring that a curve is self-avoiding (i.e. does not come close to a self-intersection) is not as extreme as ideal knots. In the following sections, we describe how to soften ρ_G and define a penalty function that averages the effect of self-contact in a way that allows us to compute shape derivatives [15,45,55]. Moreover, we incorporate this into a constrained shape optimization method for computing optimal configurations of curves that minimize their bending energy with an additional penalty term to prevent self-intersections.

2.2.1. Weighted distance

We introduce a different geometric interpretation of the point tangent radius. Let us define the **normal distance**

$$|\mathbf{a}|_{\boldsymbol{\tau}(\mathbf{z})} := \left[\mathbf{a}^T (\mathbf{I} - \boldsymbol{\tau}(\mathbf{z}) \otimes \boldsymbol{\tau}(\mathbf{z})) \mathbf{a} \right]^{1/2}, \tag{19}$$

where \mathbf{a} is an arbitrary vector and \mathbf{z} is in Σ . In other words, it is the length of the vector \mathbf{a} when projected onto the subspace orthogonal to the tangent space of Σ at \mathbf{z} ; so, $|\cdot|_{\boldsymbol{\tau}(\mathbf{z})}$ is a semi-norm. Now define the **normal distance ratio**:

$$\mathcal{N}(\mathbf{a}, \boldsymbol{\tau}(\mathbf{z})) := \frac{|\mathbf{a}|_{\boldsymbol{\tau}(\mathbf{z})}}{|\mathbf{a}|} = \frac{\text{normal length}}{\text{standard length}} = \left[1 - \left(\frac{\mathbf{a}}{|\mathbf{a}|} \cdot \boldsymbol{\tau}(\mathbf{z}) \right)^2 \right]^{1/2}. \tag{20}$$

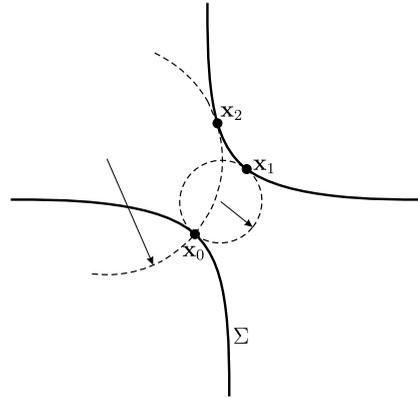


Fig. 3. Diagram of the behavior of the point tangent radius \mathcal{R}_{pt} . Part of a curve Σ is shown with three points identified. Direct computation gives $|\mathbf{x}_0 - \mathbf{x}_1|/2 = \mathcal{R}_{pt}(\mathbf{x}_0 - \mathbf{x}_1, \boldsymbol{\tau}(\mathbf{x}_1)) = \mathcal{R}_{pt}(\mathbf{x}_1 - \mathbf{x}_0, \boldsymbol{\tau}(\mathbf{x}_0)) \approx 11.87$, $\mathcal{R}_{pt}(\mathbf{x}_0 - \mathbf{x}_2, \boldsymbol{\tau}(\mathbf{x}_2)) \approx 33.74$, and $|\mathbf{x}_0 - \mathbf{x}_2|/2 \approx 16.44$. Clearly, \mathcal{R}_{pt} inflates distances for point pairs that are not points of closest approach (i.e. type 1 critical points).

Clearly, $\mathcal{N}(\cdot, \boldsymbol{\tau}(\cdot))$ is dimensionless and takes values in $[0, 1]$. If \mathbf{a} is close to being normal to Σ at \mathbf{z} , then $\mathcal{N}(\mathbf{a}, \boldsymbol{\tau}(\mathbf{z}))$ is close to 1; if it is close to being tangent, then it is close to 0. Note that $\mathcal{N}(\mathbf{a}, \boldsymbol{\tau}(\mathbf{z}))$ is an even function with respect to the argument \mathbf{a} .

In lieu of the above, we can rewrite (8) in the form

$$\mathcal{R}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = \frac{|\mathbf{x} - \mathbf{y}|}{2} \frac{1}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))}. \tag{21}$$

If $\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = 0$ and $\mathbf{x} \neq \mathbf{y}$, then $\mathcal{R}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) := +\infty$. The first term, $|\mathbf{x} - \mathbf{y}|/2$, in (21) is simply the “half-distance” associated with two points. The second term, $\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^{-1}$, accounts for the geometry of the curve Σ . In fact, when $\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) < 1$, the second term inflates the half distance between \mathbf{x} and \mathbf{y} making them seem farther away; see Fig. 3 for an illustration.

Remark 1. One of the main issues in dealing with self-contact is how to distinguish between *nearly adjacent* points on a curve and points that are clearly disjoint but close in space. Obviously, points that are close with respect to the parameterization variable t in $\boldsymbol{\alpha}(t)$ should not play a role in determining self-contact; nearby points along a curve are supposed to be close.

A common trick to enforce self-avoidance of curves is to attach point charges along the curve with a pairwise Coulomb repulsive potential energy term defined for any two charges. For this to work, a “cut-off” distance is usually defined so that nearby charges along the curve ignore each other if they are within the cut-off distance. This is an ad-hoc approach because the cut-off distance is arbitrary and it is not clear what a “good” cut-off distance should be. For instance, what if the curve loops back quickly on itself? If the radius of curvature is less than the cut-off distance, then any type 1 critical points will be incorrectly ignored.

Hence, the normal distance ratio allows for a convenient way to ignore points on the curve that are nearly adjacent to the fixed query point \mathbf{x} , but without specifying any arbitrary cut-off distance. So $\mathcal{R}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ is a good measure for how close the points \mathbf{x} and \mathbf{y} really are with respect to self-contact.

2.2.2. Modified point tangent radius

The point tangent radius $\mathcal{R}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ quantifies self-contact, but is difficult to compute with numerically when \mathbf{y} is close to \mathbf{x} . The ratio in (21) involves small quantities for a smooth curve as \mathbf{y} approaches \mathbf{x} . Moreover, obtaining derivative information from \mathcal{R}_{pt} is even more delicate (when \mathbf{y} is close to \mathbf{x}). In order to utilize a gradient based optimization method to enforce no self-intersection (of thick curves), we need another option to compute with.

Hence, we consider a modified version of the point tangent radius:

$$\widetilde{\mathcal{R}}_{pt}(\mathbf{a}, \boldsymbol{\tau}) := \frac{|\mathbf{a}|}{2} \frac{1}{\mathcal{N}^2(\mathbf{a}, \boldsymbol{\tau})}, \tag{22}$$

i.e. we square the denominator in (21). Note that $\mathcal{R}_{pt} \leq \widetilde{\mathcal{R}}_{pt}$. The same interpretation holds, as before, except $\lim_{\mathbf{x} \rightarrow \mathbf{y}} \widetilde{\mathcal{R}}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = \infty$ (follows by standard limits). So nearby points on a smooth curve are always considered “far away” regardless of the local radius of curvature, i.e. $\widetilde{\mathcal{R}}_{pt}$ ignores the (pointwise) local radius of curvature of the curve. This is convenient in computing derivatives.

Remark 2 (Effect of $\widetilde{\mathcal{R}}_{pt}$). Using $\widetilde{\mathcal{R}}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ yields the same set of double critical (type 1) points as $\mathcal{R}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$; recall (15). This is because $\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = 1$ at double critical points. Using $\widetilde{\mathcal{R}}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ effectively removes the local

self-interaction of \mathcal{R}_{pt} . Thus, we can ignore the *sensitivity* of the local curvature when computing derivative information (also see Remark 4).

2.2.3. Sensitivities

Sections 3 and 4 present a gradient based optimization method for optimizing thick curves that minimize an energy consisting of a bending energy term and a penalized self-intersection energy. For this, we must compute the sensitivity of $\widetilde{\mathcal{R}}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ when the curve Σ is perturbed. This section states some basic formulas that we will need later.

Consider a reference curve $\Sigma_0 \subset \mathbb{R}^3$ and let $\mathbf{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be a smooth vector-valued function. We define a perturbed curve Σ by the *perturbation of the identity* approach [55,15]:

$$\Sigma_\epsilon = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{x} = \mathbf{y} + \epsilon \mathbf{V}(\mathbf{y}), \forall \mathbf{y} \in \Sigma_0\}. \tag{23}$$

Note: we could consider \mathbf{V} as only defined on Σ_0 . By straightforward calculus of variations, we have the material derivative of the distance:

$$\delta_{\mathbf{V}}|\mathbf{x} - \mathbf{y}| = \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot (\mathbf{V}(\mathbf{x}) - \mathbf{V}(\mathbf{y})). \tag{24}$$

Moreover, the material derivative of $\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ gives

$$\begin{aligned} \delta_{\mathbf{V}}\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = & - \left[1 - \left(\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right)^2 \right]^{-1/2} \left(\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right) \cdot \\ & \cdot \left\{ \frac{\mathbf{V}(\mathbf{x}) - \mathbf{V}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) - \left[\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \frac{\mathbf{V}(\mathbf{x}) - \mathbf{V}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \right] \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right. \\ & \left. + \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \dot{\boldsymbol{\tau}}(\mathbf{y}) \right\}, \end{aligned}$$

where $\dot{\boldsymbol{\tau}} = [\mathbf{I} - \boldsymbol{\tau} \otimes \boldsymbol{\tau}] \partial_s \mathbf{V}$ and ∂_s is the derivative with respect to arc-length. Rewriting, we have

$$\begin{aligned} \delta_{\mathbf{V}}\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) = & - \frac{1}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))} \left(\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right) \cdot \\ & \cdot \left\{ \boldsymbol{\tau}(\mathbf{y}) \cdot \left[\mathbf{I} - \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \otimes \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \right] \frac{\mathbf{V}(\mathbf{x}) - \mathbf{V}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \right. \\ & \left. + \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot [\mathbf{I} - \boldsymbol{\tau}(\mathbf{y}) \otimes \boldsymbol{\tau}(\mathbf{y})] \partial_s \mathbf{V}(\mathbf{y}) \right\}. \end{aligned} \tag{25}$$

2.3. Penalty approach

2.3.1. Proximity penalty

We quantify self-contact of a curve, with uniform thickness, in the following way. Let d be the cross-sectional radius of a thick curve, i.e. the radius of the cylindrical tubular neighborhood (recall (16) and Definition 1). Define the two-point self-contact function

$$\mathcal{S}_c(\mathbf{x}, \mathbf{y}) = 2\widetilde{\mathcal{R}}_{\text{pt}}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) - 2d = \frac{|\mathbf{x} - \mathbf{y}|}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^2} - 2d. \tag{26}$$

We can interpret the values of \mathcal{S}_c in the following way. If \mathbf{x}, \mathbf{y} are double-critical points (recall (15)) such that $\mathbf{x} \neq \mathbf{y}$, then $\mathcal{S}_c \geq 0$ implies that the tubular neighborhoods near \mathbf{x} and \mathbf{y} do not overlap (note that $\widetilde{\mathcal{R}}_{\text{pt}} = \mathcal{R}_{\text{pt}}$ for double-critical points). If $\mathcal{S}_c < 0$, then they do overlap.

This implies the following. If $\mathcal{S}_c(\mathbf{x}, \mathbf{y}) \geq 0$ for all \mathbf{x}, \mathbf{y} in Σ and $\rho(\mathbf{x}) \geq d$ for all \mathbf{x} in Σ , then by (15) it must be that $\underline{\rho}_G(\Sigma) \geq d$. Thus, for a curve with uniform thickness d , it will not overlap itself (no self-penetration).

Remark 3 (Penalize self-intersection). In the case of optimizing the shape of a thick curve (with thickness d) such that it never self-intersects, we could enforce that $\mathcal{S}_c \geq 0$ and the local radius of curvature be larger than d . Another option is to soften the constraint by merely *penalizing* self-intersection of the tubular neighborhood of the curve by incorporating \mathcal{S}_c into an appropriate barrier function. This is done in Section 3 which utilizes the bending energy of a curve to penalize the local radius of curvature.

Therefore, we use the following penalization approach for enforcing $S_c(\mathbf{x}, \mathbf{y}) \geq 0$. Let $\psi : \mathbb{R} \rightarrow \mathbb{R}_+$ be a C^1 cut-off function defined by

$$\psi(t) = \begin{cases} 0, & \text{if } t \geq 0, \\ (1/2)t^2, & \text{if } t < 0, \end{cases} \tag{27}$$

and define the *proximity penalty function* at \mathbf{x} in Σ :

$$Q_p(\mathbf{x}) = \int_{\Sigma} \psi(S_c(\mathbf{x}, \mathbf{y})) ds(\mathbf{y}). \tag{28}$$

So $Q_p(\mathbf{x})$ vanishes when no “points of closest approach” are sufficiently close to \mathbf{x} . Alternatively, it is positive if a point of closest approach is close to \mathbf{x} . Note that $Q_p(\mathbf{x})$ also depends on Σ .

The main point of (28) is that it averages the effect of self-contact for points of closest approach, instead of the “hard” constraint imposed by the minimization in (9). Hence, we refer to (28) as a “softened” version of self-contact for points of closest approach (also see Remark 5).

The set of type 1 critical points (11) and the set of double critical points \mathcal{U}_{dc} (see (14)) for a curve Σ can vary erratically as Σ smoothly deforms. In other words, recall Σ_ϵ in (23) and let $\rho_G^\epsilon(\mathbf{x}_\epsilon)$ be the global radius of curvature for Σ_ϵ at the point

$$\mathbf{x}_\epsilon = \mathbf{x} + \epsilon \mathbf{V}(\mathbf{x}), \tag{29}$$

for some fixed \mathbf{x} in Σ_0 . In general, the minimizing point in $\rho_G^\epsilon(\mathbf{x}_\epsilon)$ does not depend smoothly on ϵ (recall Section 2.2 and Fig. 2). Therefore, building a gradient based optimization method by tracking points of closest approach is not practical. Instead, we use (28) for its averaging effect.

2.3.2. Sensitivity of the proximity

The gradient of $Q_p(\mathbf{x})$ with respect to \mathbf{x} is needed for the shape optimization method in Section 3. But $Q_p(\mathbf{x}) = Q_p(\mathbf{x}, \Sigma)$ also depends on Σ and it is not clear if we can vary \mathbf{x} independently of Σ because $\mathbf{x} \in \Sigma$. First, we establish this fact.

Recall Σ_ϵ in (23), i.e.

$$\mathbf{y}_\epsilon = \Phi_\epsilon(\mathbf{y}) := \mathbf{y} + \epsilon \mathbf{V}(\mathbf{y}), \text{ for some } \mathbf{y} \in \Sigma \iff \mathbf{y}_\epsilon \in \Sigma_\epsilon, \tag{30}$$

for some arbitrary $\mathbf{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ in $C^{1,1}$. Let \mathbf{x} be a fixed point in $\Sigma \equiv \Sigma_0$. Then \mathbf{x}_ϵ is the corresponding point in Σ_ϵ through the map (30).

Note that the function $\widetilde{\mathcal{R}}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ is continuous whenever $\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) > 0$ and infinite otherwise. In fact, $\widetilde{\mathcal{R}}_{pt}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))$ is large for all \mathbf{y} in Σ sufficiently close to \mathbf{x} . Let $B_r(\mathbf{x})$ be the ball of radius $r > 0$ centered at \mathbf{x} . Then there exists $\varrho > 0$ such that

$$S_c(\mathbf{x}, \mathbf{y}) \geq 2, \text{ for all } \mathbf{y} \in B_\varrho(\mathbf{x}) \cap \Sigma. \tag{31}$$

Let $\widehat{\Sigma}_\epsilon = \Phi_\epsilon(B_\varrho(\mathbf{x}) \cap \Sigma)$. By continuity, there exists $\epsilon' > 0$ such that

$$S_{c,\epsilon}(\mathbf{x}_\epsilon, \mathbf{y}_\epsilon) \geq 1, \text{ for all } \mathbf{y}_\epsilon \in \widehat{\Sigma}_\epsilon, \text{ for all } \epsilon \in [0, \epsilon'], \tag{32}$$

where $S_{c,\epsilon}$ is the self-contact function on Σ_ϵ . Therefore,

$$\psi(S_{c,\epsilon}(\mathbf{x}_\epsilon, \mathbf{y}_\epsilon)) = \psi'(S_{c,\epsilon}(\mathbf{x}_\epsilon, \mathbf{y}_\epsilon)) = 0, \text{ for all } \mathbf{y}_\epsilon \in \widehat{\Sigma}_\epsilon, \text{ for all } \epsilon \in [0, \epsilon']. \tag{33}$$

Define $\widetilde{\Sigma}_\epsilon = \Phi_\epsilon(\Sigma \setminus B_\varrho(\mathbf{x})) = \Sigma_\epsilon \setminus \widehat{\Sigma}_\epsilon$. From (33), we see that

$$Q_p(\mathbf{x}_\epsilon, \widetilde{\Sigma}_\epsilon) = \int_{\widetilde{\Sigma}_\epsilon} \psi(S_{c,\epsilon}(\mathbf{x}_\epsilon, \mathbf{y}_\epsilon)) ds(\mathbf{y}_\epsilon), \text{ for all } \epsilon \in [0, \epsilon']. \tag{34}$$

Clearly, $\widetilde{\Sigma}_\epsilon$ does not depend on \mathbf{V} evaluated at \mathbf{x} . In particular, \mathbf{x}_ϵ and $\widetilde{\Sigma}_\epsilon$ are independent, which means they can be varied independently; this is the *main reason* for introducing (22). Therefore, by (34), \mathbf{x} is simply a parameter in $Q_p(\mathbf{x})$, which means we can differentiate $Q_p(\mathbf{x})$ with respect to \mathbf{x} in the standard way.

Computing the gradient, we find

$$\nabla_{\mathbf{x}} Q_p(\mathbf{x}) = \int_{\Sigma} \psi'(S_c(\mathbf{x}, \mathbf{y})) \nabla_{\mathbf{x}} S_c(\mathbf{x}, \mathbf{y}) ds(\mathbf{y}), \tag{35}$$

where

$$\begin{aligned} \nabla_{\mathbf{x}} \mathcal{S}_c(\mathbf{x}, \mathbf{y}) &= \nabla_{\mathbf{x}} \left(\frac{|\mathbf{x} - \mathbf{y}|}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^2} \right) \\ &= \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \frac{1}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^2} - 2 \frac{|\mathbf{x} - \mathbf{y}|}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^3} \nabla_{\mathbf{x}} \mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})). \end{aligned}$$

The last term leads to

$$\begin{aligned} \nabla_{\mathbf{x}} \mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y})) &= - \frac{1}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))} \frac{1}{|\mathbf{x} - \mathbf{y}|} \left(\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right) \\ &\quad \cdot \left[\mathbf{I} - \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \otimes \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \right] \boldsymbol{\tau}(\mathbf{y}), \end{aligned}$$

which follows by calculations similar to (25). Altogether, we obtain

$$\begin{aligned} \nabla_{\mathbf{x}} \mathcal{S}_c(\mathbf{x}, \mathbf{y}) &= \frac{1}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^2} \left\{ \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \right. \\ &\quad \left. + 2 \frac{1}{\mathcal{N}(\mathbf{x} - \mathbf{y}, \boldsymbol{\tau}(\mathbf{y}))^2} \left(\frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \cdot \boldsymbol{\tau}(\mathbf{y}) \right) \left[\mathbf{I} - \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \otimes \frac{\mathbf{x} - \mathbf{y}}{|\mathbf{x} - \mathbf{y}|} \right] \boldsymbol{\tau}(\mathbf{y}) \right\}. \end{aligned} \tag{36}$$

Remark 4. The presence of $\widetilde{\mathcal{R}}_{pt}$ in $\mathcal{S}_c(\mathbf{x}, \mathbf{y})$ makes computing the gradient of $Q_p(\mathbf{x})$ with respect to \mathbf{x} straightforward. If we had used \mathcal{R}_{pt} instead, then the calculation would be much more involved because of the self-interaction of the local curvature.

3. Minimizing energy

3.1. Curve energy

Given a thick curve $\Sigma \subset \mathbb{R}^3$, with thickness d (see Definition 1), we define its energy to be

$$\mathcal{A}(\Sigma) = J_b(\Sigma) + \eta J_p(\Sigma), \tag{37}$$

$$J_b(\Sigma) = \frac{1}{2} \int_{\Sigma} |\boldsymbol{\kappa}(\mathbf{x})|^2 ds(\mathbf{x}), \tag{38}$$

$$J_p(\Sigma) = \int_{\Sigma} Q_p(\mathbf{x}) ds(\mathbf{x}), \tag{39}$$

where (38) is the so-called bending energy [60], (39) is the proximity penalty functional, and $\eta > 0$ is a penalization parameter. The bending energy is used to control the local radius of curvature. We use J_p as an energy penalization on the proximity of the curve to itself (recall Section 2.3). In other words, J_b and J_p penalize the terms in the characterization (15).

Our goal is to find a curve Σ^* that minimizes (37) subject to constraints discussed in the following sections. Choosing η large enough will ensure minimal overlap of $T(\Sigma, d) = \cup_{\mathbf{x} \in \Sigma} \mathcal{D}(\mathbf{x}, d)$, where $\mathcal{D}(\mathbf{x}, d)$ is the open disk centered at \mathbf{x} , perpendicular to $\boldsymbol{\tau}(\mathbf{x})$, with radius d , i.e. the region occupied by the “thick” curve (recall Section 2.1.5).

Remark 5. Many kinds of knot energies and self-avoidance energies have been proposed (for example, see [9,19,39,36,40,48,49]). But they are more global in nature. For instance, [48,49] define an L^p integral of $1/\rho_G(\cdot)$ as a self-avoidance energy, which puts an L^p bound on the global curvature. As $p \rightarrow \infty$, their energy recovers the minimum global radius of curvature (13) [19]. Thus, their proposed energy gives a means to interpolate between a purely “soft” repulsive potential and the “hard” potential corresponding to exact no self-penetration. The energy we propose is more localized and is only active when points of closest approach become too close. This is consistent with our goal of modeling polymer strands adsorbed onto surfaces.

3.1.1. Sensitivity of the bending energy

Consider a reference curve $\Sigma_0 \subset \mathbb{R}^3$ and let $\mathbf{V} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ be a smooth vector-valued function. Recall the perturbed curve Σ_ϵ in (23). With this, we consider the functional $J_b(\epsilon) \equiv J_b(\Sigma_\epsilon)$ as a function of ϵ . Differentiating $J_b(\epsilon)$ gives the so-called shape perturbation of J_b with respect to \mathbf{V} , i.e.

$$\delta J_b(\Sigma; \mathbf{V}) := \left. \frac{d}{d\epsilon} J_b(\epsilon) \right|_{\epsilon=0}. \tag{40}$$

This is a classic calculation that can be found in [5,6,17,60,55] among others. For our purposes, we refer to the result in [6] for the shape sensitivity of the bending energy of a surface. Their result reduces to the following in the case of a curve:

$$\delta J_b(\Sigma; \mathbf{V}) = \int_{\Sigma} \nabla_{\Sigma} \boldsymbol{\kappa} : \nabla_{\Sigma} \mathbf{V} - 2 \int_{\Sigma} [(\nabla_{\Sigma} \text{id}_{\Sigma})(\nabla_{\Sigma} \mathbf{V})] : [\nabla_{\Sigma} \boldsymbol{\kappa}] + \frac{1}{2} \int_{\Sigma} (\nabla_{\Sigma} \cdot \boldsymbol{\kappa})(\nabla_{\Sigma} \cdot \mathbf{V}) - (\partial_s \mathbf{V}) \cdot \boldsymbol{\kappa} \Big|_{\partial \Sigma} - \frac{1}{2} |\boldsymbol{\kappa}|^2 \mathbf{V} \cdot \boldsymbol{\tau} \Big|_{\partial \Sigma}, \tag{41}$$

where $\Big|_{\partial \Sigma}$ denotes evaluation at the end points of the open curve Σ , $\nabla_{\Sigma} := \boldsymbol{\tau} \partial_s$, and ∂_s is the derivative with respect to arc-length. The boundary terms are a minor modification of the derivation in [6] which can be found in [55].

Because Σ is a curve, we can further simplify (41) to

$$\delta J_b(\Sigma; \mathbf{V}) = \int_{\Sigma} \partial_s \boldsymbol{\kappa} \cdot \partial_s \mathbf{V} - \frac{3}{2} \int_{\Sigma} (\boldsymbol{\tau} \cdot \partial_s \boldsymbol{\kappa})(\boldsymbol{\tau} \cdot \partial_s \mathbf{V}) - (\partial_s \mathbf{V}) \cdot \boldsymbol{\kappa} \Big|_{\partial \Sigma} - \frac{1}{2} |\boldsymbol{\kappa}|^2 \mathbf{V} \cdot \boldsymbol{\tau} \Big|_{\partial \Sigma}. \tag{42}$$

3.1.2. Sensitivity of the proximity

Applying standard shape differentiation formulas [15,23,55] to (39), we get

$$\delta J_p(\Sigma; \mathbf{V}) = \int_{\Sigma} Q_p(\mathbf{x})(\boldsymbol{\tau} \cdot \partial_s \mathbf{V}) ds(\mathbf{x}) + \int_{\Sigma} \mathbf{V} \cdot \nabla_{\mathbf{x}} Q_p(\mathbf{x}) ds(\mathbf{x}), \tag{43}$$

where the formula for $\nabla_{\mathbf{x}} Q_p(\mathbf{x})$ is given in (35).

3.2. Constraints

3.2.1. Local inextensibility constraint

Let $\boldsymbol{\alpha} : [0, 1] \rightarrow \Sigma$ be a parameterization of Σ ; similarly, let $\boldsymbol{\alpha}_0$ parameterize a fixed reference curve Σ_0 . Define the local length constraint functional as

$$B(q, \Sigma) = \int_{\Sigma} q(\boldsymbol{\alpha}_0 \circ \boldsymbol{\alpha}^{-1}(\mathbf{x})) ds(\mathbf{x}) - \int_{\Sigma_0} q(\mathbf{x}) ds(\mathbf{x}), \quad \text{for all } q : \Sigma_0 \rightarrow \mathbb{R}, \tag{44}$$

where $\boldsymbol{\alpha}^{-1}$ is the inverse map of $\boldsymbol{\alpha}$, and q is any scalar valued $L^2(\Sigma_0)$ integrable function. So the inextensibility constraint is simply $B(q, \Sigma) = 0$ for all q , i.e. the parts of the curve Σ and Σ_0 that correspond (through the map $\boldsymbol{\alpha}_0 \circ \boldsymbol{\alpha}^{-1}$) must have equal length. Imposing a global length constraint may not be adequate in all cases. If the thickness d varies along the curve, then a local length constraint is necessary to preserve the desired thickness distribution. Also, in the numerics, it may be desirable to use an adaptive mesh for representing Σ ; again, a local length constraint is needed for maintaining the local mesh size.

The sensitivity of (44) can be computed by standard methods in the calculus of variations [16,26,55] and/or standard shape differentiation formulas. We write this as a bilinear form

$$b(p, \mathbf{V}) := \delta_{\Sigma} B(p, \Sigma; \mathbf{V}) = \int_{\Sigma} p \boldsymbol{\tau} \cdot \partial_s \mathbf{V}, \tag{45}$$

for all smooth functions \mathbf{V} and $L^2(\Sigma)$ functions p , where p is a Lagrange multiplier (see [57] for a similar approach). We use δ_{Σ} to emphasize that we are perturbing with respect to Σ .

3.2.2. Surface attachment constraint

Recalling the application, i.e. that a thick curve Σ models a polymer strand being adsorbed onto a surface $\Gamma \subset \mathbb{R}^3$, we must also impose this condition into the minimization of (37). This could be done by adding an additional energy penalization to (37); this would yield an additional force (similar to the other sensitivities) that pushes Σ onto Γ . Alternatively, we can constrain Σ to lie on Γ . In this paper, we take the latter approach.

Let Γ be a smooth surface without boundary and represent it as the zero set of a level set function $\phi : \mathbb{R}^3 \rightarrow \mathbb{R}$:

$$\Gamma = \{\mathbf{x} \in \mathbb{R}^3 : \phi(\mathbf{x}) = 0\}. \tag{46}$$

For convenience, we assume that ϕ is, in fact, the distance function to Γ . Now define the surface constraint functional

$$C(\mu, \Sigma) = \int_{\Sigma} \mu(\mathbf{x}) \phi(\mathbf{x}) ds(\mathbf{x}), \quad \text{for all } \mu : \Sigma \rightarrow \mathbb{R}. \tag{47}$$

So the surface constraint is simply $C(\mu, \Sigma) = 0$ for all μ in $L^2(\Sigma)$, i.e. Σ should lie on the zero level set of ϕ . The sensitivity of (47) follows by basic shape differentiation, which we write as a bilinear form

$$c(\lambda, \mathbf{V}) := \delta_{\Sigma} C(\lambda, \Sigma; \mathbf{V}) = \int_{\Sigma} \lambda (\mathbf{V} \cdot \nabla \phi) + \int_{\Sigma} \lambda \phi \boldsymbol{\tau} \cdot \partial_{\mathbf{S}} \mathbf{V}, \tag{48}$$

for all smooth functions \mathbf{V} and $L^2(\Sigma)$ functions λ , where λ is a Lagrange multiplier.

3.3. Minimization problem

We define the admissible set of curves to be

$$\mathcal{U} = \{ \Sigma \in C^{1,1} : \Sigma \subset \Gamma, B(q, \Sigma) = 0, \forall q \in L^2(\Sigma_0), \text{ and } C(\mu, \Sigma) = 0, \forall \mu \in L^2(\Sigma) \}, \tag{49}$$

where Σ_0 is a fixed reference curve (recall Section 3.2.1). The minimization problem is then to find a curve Σ^* in \mathcal{U} such that

$$\mathcal{A}(\Sigma^*) = \min_{\Sigma \in \mathcal{U}} \mathcal{A}(\Sigma). \tag{50}$$

Similar optimization problems have been considered before in the context of non-linear rods and knots. In [20,42], they showed the existence of $C^{1,1}$ minimizing curves that satisfied a thickness constraint. A similar problem was also analyzed for optimizing the shape of magnetic micro-swimmers, with a local length constraint, in [27,57].

We will not delve into the theoretical intricacies of the continuous optimization problem (50), because most of the theoretical issues have been established in other research. The main goal in the present paper is to develop a computational algorithm for finding numerical realizations of Σ^* in (50) and to illustrate our method in a variety of cases (Section 5). Thus, we will keep the discussion of the continuous optimization problem at a formal level.

In order to solve this non-linear constrained optimization problem, we first reformulate it using Lagrange multipliers. Hence, consider the Lagrangian

$$\mathcal{L}(\Sigma, p, \lambda) = \mathcal{A}(\Sigma) - B(p, \Sigma) - C(\lambda, \Sigma), \tag{51}$$

defined for all smooth curves Σ and $L^2(\Sigma)$ functions p, λ . In other words, the local inextensibility constraint and surface constraint are enforced through Lagrange multipliers.

The first order optimality conditions for (51) follow from shape differentiation and differentiating the multipliers, i.e. find $(\Sigma^*, p^*, \lambda^*)$ such that

$$\begin{aligned} \delta_{\Sigma} \mathcal{L}(\Sigma^*, p^*, \lambda^*; \mathbf{Y}) &\equiv \delta_{\Sigma} \mathcal{A}(\Sigma^*; \mathbf{Y}) - \delta_{\Sigma} B(p^*, \Sigma^*; \mathbf{Y}) - \delta_{\Sigma} C(\lambda^*, \Sigma^*; \mathbf{Y}) = 0, \\ \delta_p \mathcal{L}(\Sigma^*, p^*, \lambda^*; q) &\equiv B(q, \Sigma^*) = 0, \\ \delta_{\lambda} \mathcal{L}(\Sigma^*, p^*, \lambda^*; \mu) &\equiv C(\mu, \Sigma^*) = 0, \end{aligned} \tag{52}$$

for all possible variations \mathbf{Y}, q , and μ . We postpone the details of solving this system to Section 4.2.

4. Discrete SQP method

We now discretize the problem using finite elements. We introduce discrete function spaces to represent Σ , as well as the Lagrange multipliers (p, λ) . We then describe a gradient descent type of scheme for minimizing $\mathcal{A}(\Sigma)$ under the local inextensibility and surface constraints described in Section 3.2.

4.1. Finite element spaces

We begin by partitioning the interval $[0, 1]$ into $N - 1$ sub-intervals: $\mathcal{I} := \{I_j\}_{j=1}^{N-1}$, i.e. the mesh (N is the number of points or vertices). Define a continuous piecewise polynomial space on \mathcal{I} :

$$\begin{aligned} \mathbb{X}_k &:= \{v \in C^0([0, 1]) : v|_{I_j} \in \mathcal{P}_k(I_j), 1 \leq j \leq N - 1\}, k \geq 1, \\ \mathbb{X}_0 &:= \{v \in L^2([0, 1]) : v|_{I_j} \in \mathcal{P}_0(I_j), 1 \leq j \leq N - 1\}, \end{aligned} \tag{53}$$

where $\mathcal{P}_k(I)$ is the space of polynomials of degree k on the interval I . Note that if \mathbf{v} is a vector valued function, then \mathbf{v} in \mathbb{X}_k means that each component of \mathbf{v} is in \mathbb{X}_k (the same notation holds for all the other spaces).

For a general open curve Σ , we parameterize it by $\boldsymbol{\alpha} : [0, 1] \rightarrow \mathbb{R}^3$. For the numerics, we replace Σ with Σ_h which is parameterized by $\boldsymbol{\alpha}_h : [0, 1] \rightarrow \mathbb{R}^3$ (h is a discretization parameter). In other words, $\Sigma_h = \boldsymbol{\alpha}_h([0, 1])$. Moreover, we take $\boldsymbol{\alpha}_h$ in \mathbb{X}_2 (i.e. continuous piecewise quadratic over \mathcal{I}). Thus, the map $\boldsymbol{\alpha}_h$ induces a partition of edge segments $\mathcal{E} = \{E_j\}$ on Σ_h , i.e. $E_j = \boldsymbol{\alpha}_h(I_j)$ for $1 \leq j \leq N - 1$. With this, we define the local mesh size $h_j = |E_j|$ and smallest mesh size $h = \min_j h_j$ for the mesh \mathcal{E} on Σ_h .

Algorithm 1 SQP method.

Given a fixed reference curve Σ_h^0 , with parameterization $\alpha^0 : [0, 1] \rightarrow \mathbb{R}^3$, let α_h^0 be the standard Lagrange interpolant of α^0 over $[0, 1]$. Initialize merit function penalty $\zeta^0 := 0$.
for $k = 0, 1, 2, \dots$ **do**
 Perform line-search with merit function (see Algorithm 2) to obtain β^{k+1} .
 Update curve: $\alpha_h^{k+1}(t) := \alpha_h^k(t) + \beta^{k+1} \mathbf{V}_h^{k+1} \circ \alpha_h^k(t)$, for all t . This yields Σ_h^{k+1} .
end for
 Note: convergence is obtained when β^{k+1} becomes too small.

Now define a finite element space on Σ_h [7,8,13] for the curve perturbation:

$$\mathbb{Y}_h := \{v \in C^0(\Sigma_h) : (v \circ \alpha_h) \in \mathbb{X}_2\}, \tag{54}$$

and piecewise constant spaces for the multipliers

$$\mathbb{Q}_h := \{q \in L^2(\Sigma_h) : (q \circ \alpha_h) \in \mathbb{X}_0\}, \quad \mathbb{M}_h := \{\mu \in L^2(\Sigma_h) : (\mu \circ \alpha_h) \in \mathbb{X}_0\}. \tag{55}$$

4.2. SQP method

The discrete version of the first order optimality conditions (52) comes about by replacing $(\Sigma^*, p^*, \lambda^*)$ with $(\Sigma_h^*, p_h^*, \lambda_h^*)$ in $\mathbb{Y}_h \times \mathbb{Q}_h \times \mathbb{M}_h$ and taking variations (\mathbf{Y}, q, μ) in $\mathbb{Y}_h \times \mathbb{Q}_h \times \mathbb{M}_h$.

Applying a quasi-Newton method to (52) yields an iterative method for obtaining a solution. Each iteration of the method requires solving a saddle-point system to obtain a descent direction \mathbf{V}_h which is used to update the curve Σ_h . Thus, given the current guess Σ_h^k at iteration $k \geq 0$, we obtain \mathbf{V}_h^{k+1} by solving the following problem: find $(\mathbf{V}_h^{k+1}, p_h^{k+1}, \lambda_h^{k+1})$ in $\mathbb{Y}_h \times \mathbb{Q}_h \times \mathbb{M}_h$ such that

$$\begin{aligned} a(\mathbf{V}_h^{k+1}, \mathbf{Y}) + b(p_h^{k+1}, \mathbf{Y}) + c(\lambda_h^{k+1}, \mathbf{Y}) &= -\delta_{\Sigma} \mathcal{A}(\Sigma_h^k, \mathbf{Y}), \quad \forall \mathbf{Y} \in \mathbb{Y}_h, \\ b(q, \mathbf{V}_h^{k+1}) &= -B(q, \Sigma_h^k), \quad \forall q \in \mathbb{Q}_h, \\ c(\mu, \mathbf{V}_h^{k+1}) &= -C(\mu, \Sigma_h^k), \quad \forall \mu \in \mathbb{M}_h, \end{aligned} \tag{56}$$

where $\mathbb{Y}_h, \mathbb{Q}_h, \mathbb{M}_h$ are defined on Σ_h^k . The bilinear form $a(\cdot, \cdot)$ is a chosen inner product on \mathbb{Y}_h . It replaces the usual Hessian term in Newton’s method with a positive definite matrix, which makes the method a quasi-Newton method. Iterating (56) yields a sequential quadratic programming (SQP) method for optimizing Σ (see Algorithm 1 for a precise description). For further details on SQP methods, see [10,21,38,25,51]. Similar approaches can be found in [58,57,27].

4.3. Merit function

The presence of the constraints affects the choice of step-size when updating the curve, i.e. the cost functional $\mathcal{A}(\Sigma_h)$ may decrease but at the expense of violating the constraints. So we must balance reducing $\mathcal{A}(\Sigma_h)$ versus violating the constraints. Therefore, the line-search in Algorithm 1 is performed using a merit function [38], which is defined as

$$\theta(\Sigma_h, \zeta) = \mathcal{A}(\Sigma_h) + \zeta (\|B(\Sigma_h)\|_{l^1} + \|C(\Sigma_h)\|_{l^1}), \tag{57}$$

where $\zeta \geq 0$ is a penalty parameter for the constraint violation and is updated at each optimization step. The l^1 norms are defined as

$$\|B(\Sigma_h)\|_{l^1} = \max_{\substack{q \in \mathbb{Q}_h \\ \|q\|_{l^\infty} = 1}} B(q, \Sigma_h) = \sum_{j=1}^{N-1} \left| |E_j| - |E_j^0| \right|, \tag{58}$$

where $E_j \subset \Sigma_h, E_j^0 \subset \Sigma_h^0$ are corresponding edges through the map $\alpha_0 \circ \alpha^{-1}$, and

$$\|C(\Sigma_h)\|_{l^1} = \max_{\substack{\mu \in \mathbb{M}_h \\ \|\mu\|_{l^\infty} = 1}} C(\mu, \Sigma_h) = \sum_{j=1}^{N-1} \left| \int_{E_j} \phi(\mathbf{x}) ds(\mathbf{x}) \right|. \tag{59}$$

The directional derivative, at step k of the optimization, is given by [38]

$$\delta_{\Sigma} \theta(\Sigma_h^k, \zeta; \mathbf{V}_h^{k+1}) = \delta_{\Sigma} \mathcal{A}(\Sigma_h^k, \mathbf{V}_h^{k+1}) - \zeta (\|B(\Sigma_h^k)\|_{l^1} + \|C(\Sigma_h^k)\|_{l^1}), \tag{60}$$

where $(\mathbf{V}_h^{k+1}, p_h^{k+1}, \lambda_h^{k+1})$ solves (56). In fact, the derivative can be estimated by [38]

Algorithm 2 Line search with merit function.

- 1: Set parameter $\xi \in (0, 1)$ and initial step size $\beta_0 > 0$.
- 2: Input: Σ_h^k and ζ^k .
- 3: Initialize: $\beta^{k+1} := \beta_0$, ACCEPT := FALSE.
- 4: **while** ACCEPT == FALSE **do**
- 5: Compute $(\mathbf{V}_h^{k+1}, p_h^{k+1}, \lambda_h^{k+1})$ by solving (66) with β^{k+1} .
- 6: Define penalty: $\zeta^{k+1} := \max(\zeta^k, \|p_h^{k+1}\|_{L^\infty}, \|\lambda_h^{k+1}\|_{L^\infty})$.
- 7: Compute $\alpha_h^{k+1}(t) := \alpha_h^k(t) + \beta^{k+1} \mathbf{V}_h^{k+1} \circ \alpha_h^k(t)$, for all t . This defines Σ_h^{k+1} .
- 8: Evaluate $\theta(\Sigma_h^k, \zeta^{k+1})$ and $\theta(\Sigma_h^{k+1}, \zeta^{k+1})$ by (57), and $\delta\theta(\Sigma_h^k, \zeta^{k+1}; \mathbf{V}_h^{k+1})$ by (60).
- 9: **if** $\theta(\Sigma_h^{k+1}, \zeta^{k+1}) \leq \theta(\Sigma_h^k, \zeta^{k+1}) + \xi \beta^{k+1} \delta\theta(\Sigma_h^k, \zeta^{k+1}; \mathbf{V}_h^{k+1})$ **then**
- 10: ACCEPT := TRUE
- 11: **else**
- 12: $\beta^{k+1} := \beta^{k+1}/2$.
- 13: **end if**
- 14: **end while**
- 15: **return** β^{k+1} and ζ^{k+1} .
- 16: Note: if β^{k+1} becomes too small, we break the loop.

$$\delta_\Sigma \theta(\Sigma_h^k, \zeta; \mathbf{V}_h^{k+1}) \leq -a(\mathbf{V}_h^{k+1}, \mathbf{V}_h^{k+1}) - \left(\zeta - \max(\|p_h^{k+1}\|_{L^\infty}, \|\lambda_h^{k+1}\|_{L^\infty}) \right) \left(\|B(\Sigma_h^k)\|_{L^1} + \|C(\Sigma_h^k)\|_{L^1} \right). \tag{61}$$

Therefore, at each step of the optimization, we update the constraint penalty by $\zeta^{k+1} := \max(\zeta^k, \|p_h^{k+1}\|_{L^\infty}, \|\lambda_h^{k+1}\|_{L^\infty})$; this ensures that $\delta\theta(\Sigma_h^k, \zeta; \mathbf{V}_h^{k+1}) \leq 0$. Thus, at the k th optimization step, we define a step size β^{k+1} to be acceptable if

$$\theta(\Sigma_h^{k+1}, \zeta^{k+1}) \leq \theta(\Sigma_h^k, \zeta^{k+1}) + \xi \beta^{k+1} \delta_\Sigma \theta(\Sigma_h^k, \zeta^{k+1}; \mathbf{V}_h^{k+1}), \quad \text{for some } \xi \in (0, 1), \tag{62}$$

where Σ_h^{k+1} is parameterized by $\alpha_h^{k+1}(t) := \alpha_h^k(t) + \beta^{k+1} \mathbf{V}_h^{k+1} \circ \alpha_h^k(t)$, for all t in $[0, 1]$. The precise line search method is described in Algorithm 2. See [38] for more details and [58] for an example of using a line search with merit function.

4.4. Discrete weak formulation

The sensitivity of the bending energy $\delta J_b(\Sigma_h)$ involves derivatives of the curvature κ (see Section 3.1.1). This means that directly computing $\delta J_b(\Sigma_h)$ requires a sufficiently smooth curve. One way to circumvent this is to adopt the approach in [6] and introduce κ as an unknown and append another equation to (56). To motivate this, consider the following integration by parts relation when Σ is an open smooth curve [18,14],

$$\int_\Sigma \partial_s \text{id}_\Sigma \cdot \partial_s \mathbf{W} = \boldsymbol{\tau} \cdot \mathbf{W} \Big|_{\partial \Sigma} - \int_\Sigma \partial_s^2 \text{id}_\Sigma \cdot \mathbf{W} = \boldsymbol{\tau} \cdot \mathbf{W} \Big|_{\partial \Sigma} + \int_\Sigma \kappa \cdot \mathbf{W}, \quad \text{for all smooth } \mathbf{W}, \tag{63}$$

where $\text{id}_\Sigma : \Sigma \rightarrow \Sigma$ is the identity map on Σ , $\boldsymbol{\tau} = \partial_s \text{id}_\Sigma$, and $\kappa = -\partial_s^2 \text{id}_\Sigma$ [16]. Given Σ , we can view (63) as a weak form of the curvature κ .

Therefore, at each step of the optimization, we impose (63) at the discrete level. This can be done by writing (63) explicitly in terms of Σ_h^k , implicitly in terms of Σ_h^{k+1} , or semi-implicitly. In order to avoid taking small steps during the optimization process, and for efficiency, we follow [56,6] and use the semi-implicit approach. This gives

$$\int_{\Sigma_h^k} \partial_s (\alpha_h^{k+1} \circ (\alpha_h^k)^{-1}) \cdot \partial_s \mathbf{W} = \int_{\Sigma_h^k} \kappa_h^{k+1} \cdot \mathbf{W}, \quad \text{for all } \mathbf{W} \in \mathbb{W}_{h,0},$$

where κ_h^{k+1} has zero boundary values and lies in the space

$$\mathbb{W}_{h,0} := \{v \in C^0(\Sigma_h^k) : (v \circ \alpha_h^k) \in \mathbb{X}_2 \text{ and } v = 0, \text{ at } \partial \Sigma_h^k\}. \tag{64}$$

Choosing zero boundary conditions for curvature is a natural boundary condition when modeling elastic beams. It also has the added benefit of simplifying the formulation. Furthermore, we can use the relation $\alpha_h^{k+1} \circ (\alpha_h^k)^{-1} := \text{id}_{\Sigma_h^k} + \beta^{k+1} \mathbf{V}_h^{k+1}$ and the fact that $\partial_s \text{id}_{\Sigma_h^k} = \boldsymbol{\tau}_h^k$ to arrive at

$$\int_{\Sigma_h^k} \kappa_h^{k+1} \cdot \mathbf{W} - \beta^{k+1} \int_{\Sigma_h^k} \partial_s \mathbf{V}_h^{k+1} \cdot \partial_s \mathbf{W} = \int_{\Sigma_h^k} \boldsymbol{\tau}_h^k \cdot \partial_s \mathbf{W}, \quad \text{for all } \mathbf{W} \in \mathbb{W}_{h,0}, \tag{65}$$

where $\boldsymbol{\tau}_h^k$ is the explicit tangent vector of Σ_h^k . Similar approaches can be found in [5,4].

Hence, given a trial step size β^{k+1} , we obtain the corresponding descent direction \mathbf{V}_h^{k+1} by solving the following problem. Find $(\mathbf{V}_h^{k+1}, \boldsymbol{\kappa}_h^{k+1}, p_h^{k+1}, \lambda_h^{k+1})$ in $\mathbb{Y}_h \times \mathbb{W}_{h,0} \times \mathbb{Q}_h \times \mathbb{M}_h$ such that

$$\begin{aligned} m(\partial_s \boldsymbol{\kappa}_h^{k+1}, \partial_s \mathbf{Y}) - (3/2)m(\boldsymbol{\tau}_h^k \cdot \partial_s \boldsymbol{\kappa}_h^{k+1}, \boldsymbol{\tau}_h^k \cdot \partial_s \mathbf{Y}) \\ + a(\mathbf{V}_h^{k+1}, \mathbf{Y}) + b(p_h^{k+1}, \mathbf{Y}) + c(\lambda_h^{k+1}, \mathbf{Y}) = -\eta \delta_{\Sigma_h^k} J_p(\Sigma_h^k, \mathbf{Y}), \quad \forall \mathbf{Y} \in \mathbb{Y}_h, \\ m(\boldsymbol{\kappa}_h^{k+1}, \mathbf{W}) - \beta^{k+1} m(\partial_s \mathbf{V}_h^{k+1}, \partial_s \mathbf{W}) = m(\boldsymbol{\tau}_h^k, \partial_s \mathbf{W}), \quad \forall \mathbf{W} \in \mathbb{W}_{h,0}, \\ b(q, \mathbf{V}_h^{k+1}) = -B(q, \Sigma_h^k), \quad \forall q \in \mathbb{Q}_h, \\ c(\mu, \mathbf{V}_h^{k+1}) = -C(\mu, \Sigma_h^k), \quad \forall \mu \in \mathbb{M}_h, \end{aligned} \tag{66}$$

where $m(\mathbf{u}, \mathbf{v}) = \int_{\Sigma_h^k} \mathbf{u} \cdot \mathbf{v}$.

Remark 6. Algorithm 2 (for the line search) must solve (66) for each trial step size. The alternative would be to use an explicit calculation of $\boldsymbol{\kappa}_h^{k+1}$ and solve for \mathbf{V}_h^{k+1} once before doing the line search. However, we found that this leads to extremely small step sizes, so it is cheaper (overall) to include (66) in the line search.

The reason is due to the bending energy. Assuming $a(\cdot, \cdot)$ is an L^2 inner product, (66) is a modified version of the Willmore flow [5,4,6] for curves with additional constraints and the proximity penalty functional as an extra driving force. The L^2 -gradient flow of the Willmore functional is a non-linear PDE with one derivative in time and four derivatives in space. So using an explicit method for the gradient flow of the Willmore functional would yield a step size restriction on the order of $\beta = O(h^4)$. Hence, the semi-implicit approach in (66) is crucial for improving the convergence rate to a minimizer.

4.5. Computation of the proximity penalty

We evaluate J_p and δJ_p by first computing an interpolant of Q_p in the space \mathbb{Y}_h . In other words, at each nodal variable \mathbf{x}_i on Σ_h , we compute $Q_p(\mathbf{x}_i)$ using numerical quadrature on Σ_h (similarly for δJ_p). The nodal variables consist of the endpoints and midpoints of each edge in \mathcal{E} .

Hence, for each point \mathbf{x}_i , we must compute an integral over Σ_h . However, we do not need to compute the integral over the entire curve. For instance, let \mathbf{x}_0 be any point in Σ_h and $E \in \mathcal{E}$ be an edge of Σ_h . Then,

$$\text{dist}(\mathbf{x}_0, E) \geq 2d, \quad \Rightarrow \quad \mathcal{S}_c(\mathbf{x}_0, \mathbf{y}) \geq 0, \quad \text{for all } \mathbf{y} \in E. \tag{67}$$

This implies that the contribution to $Q_p(\mathbf{x}_0)$ from edge E is zero (same for $\nabla Q_p(\mathbf{x}_0)$). This is a manifestation of the lower interaction property in [19].

Therefore, we only need to evaluate quadrature formulas on edges that are less than a distance of $2d$ from \mathbf{x}_0 to compute $Q_p(\mathbf{x}_0)$ and $\nabla Q_p(\mathbf{x}_0)$. We can take advantage of this by finding the nearest neighbor edges to each nodal point, which can be done efficiently using an octree data structure.

4.6. Optimization schedule for adsorbed polymers

We use the following scheme in all of the examples in Section 5.1 to compute a constrained minimizer.

1. Generate a curve that is close to the surface Γ before running Algorithm 1. This is done by minimizing the following energy:

$$\tilde{\mathcal{A}}(\Sigma) = J_b(\Sigma) + \eta J_p(\Sigma) + \eta_\phi J_\phi(\Sigma), \quad J_\phi(\Sigma) := \int_{\Sigma} \sqrt{\epsilon_0^2 + \phi^2(\mathbf{x})} ds(\mathbf{x}), \tag{68}$$

where $J_\phi(\Sigma)$ is a regularization of $\int_{\Sigma} |\phi|$ with $\epsilon_0 > 0$. The minimization is done by a gradient descent method, where the descent direction is obtained by solving a modified version of (66). This is done by setting $\lambda_h^{k+1} = 0$, omitting the last equation in (66), and changing the right-hand-side of the first equation in (66) accordingly. We then use a standard back-tracking line search with the energy (68) (no merit function) to update the curve. This first step is necessary to give a good initial curve for Algorithm 1. Once the curve is sufficiently close to Γ (we use ϕ to evaluate this), or the step-size is too small, we stop the descent scheme.

2. Run the merit function based Algorithm 1 with the curve produced by step (1) as the initial curve. During this run, we reset ζ to zero every 100 iterations. This is useful if ζ becomes artificially large during a single iteration due to one of the Lagrange multipliers having a large maximum. If ζ is extremely large, then the optimization can stall. Algorithm 1 terminates when the step size becomes less than the minimum step size allowed.

Algorithm 3 Inflate knot.

```

Define  $f^0 := 1$ . Set  $\gamma_0 = 0.1$ ,  $\text{TOL} = 10^{-5}$ .
Let  $\Sigma_h^0$  minimize  $\tilde{\mathcal{A}}(\Sigma_h, f^0)$  for all  $\Sigma_h$  of fixed length. Compute  $\rho_G^0(\mathbf{x})$  from  $\Sigma_h^0$ .
for  $k = 0, 1, 2, \dots$  do
  Compute  $\delta f^{k+1} = \omega(\rho_G^k)$ .
  Initialize:  $\gamma^{k+1} := \gamma_0$ .
  while  $\gamma^{k+1} \geq \text{TOL}$  do
    Update:  $\tilde{f} = f^k + \gamma^{k+1} \delta f^{k+1}$  (inflate).
    Let  $\tilde{\Sigma}_h$  minimize  $\tilde{\mathcal{A}}(\Sigma_h, \tilde{f})$  for all  $\Sigma_h$  of fixed length. Compute  $\tilde{\rho}_G(\mathbf{x})$  from  $\tilde{\Sigma}_h$ .
    if  $\min(\tilde{\rho}_G) > \min(\rho_G^k)$  then
      Accept the curve  $\tilde{\Sigma}_h$ :
       $\Sigma_h^{k+1} := \tilde{\Sigma}_h$ ,  $\rho_G^{k+1} := \tilde{\rho}_G$ ,  $f^{k+1} = \tilde{f}$ .
      break while.
    else
       $\gamma^{k+1} := \gamma^{k+1}/2$ .
    end if
  end while
end for
The FOR loop terminates when  $\gamma^{k+1}$  becomes less than TOL.

```

4.7. Inflating a knot

4.7.1. Overview

We present a heuristic method, using the proximity penalty approach and the SQP method in Section 4.2, for inflating mathematical knots. Recalling Section 2.1.5 and (17), a curve Σ is called an ideal knot if it maximizes $\underline{\rho}_G(\Sigma)$ amongst all curves in a specified topological knot class of fixed length.

Suppose we are given a curve Σ that minimizes (37) subject to the local length constraint (i.e. we remove the surface constraint). A necessary condition for Σ to be an ideal knot is that $\rho_G(\mathbf{x})$ is constant for all \mathbf{x} in Σ . Thus, if $\rho_G(\mathbf{x})$ is not constant, then increasing the proximity penalty where $\rho_G(\mathbf{x})$ is smallest should (hopefully) push those parts of the curve away from itself, effectively increasing $\underline{\rho}_G(\Sigma)$.

4.7.2. Algorithm

We start by considering a modified proximity penalty functional:

$$\tilde{J}_p(\Sigma, f) = \int_{\Sigma} f(\mathbf{x}) Q_p(\mathbf{x}) ds(\mathbf{x}), \tag{69}$$

where $f : \Sigma \rightarrow \mathbb{R}_+$ is a function that augments the proximity penalty locally. Computing the shape perturbation (while holding f fixed) gives

$$\begin{aligned} \delta \tilde{J}_p(\Sigma, f; \mathbf{V}) &= \int_{\Sigma} f(\mathbf{x}) Q_p(\mathbf{x})(\boldsymbol{\tau} \cdot \partial_s \mathbf{V}) ds(\mathbf{x}) \\ &\quad + \int_{\Sigma} f(\mathbf{x}) \mathbf{V} \cdot \nabla Q_p(\mathbf{x}) ds(\mathbf{x}) + \int_{\Sigma} Q_p(\mathbf{x}) \mathbf{V} \cdot \nabla_{\Sigma} f(\mathbf{x}) ds(\mathbf{x}), \end{aligned} \tag{70}$$

where ∇_{Σ} appears because f is defined only on Σ . So, given f defined on a fixed reference curve Σ^0 , we wish to find a closed curve Σ that minimizes

$$\tilde{\mathcal{A}}(\Sigma, f) = J_b(\Sigma) + \eta \tilde{J}_p(\Sigma, f), \tag{71}$$

subject to a local length constraint. This can be done by a gradient descent method analogous to step (1) in Section 4.6.

The issue now is to find f such that the minimizer of (71) is one that has $\rho_G(\mathbf{x})$ as close as possible to a constant. Algorithm 3 does this.

In Algorithm 3, we apply a map $\omega : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ to determine an incremental change in f . The map is chosen to be C^1 , positive when ρ_G is near $\underline{\rho}_G$, and zero elsewhere. The exact form of $y = \omega(x)$ is a piecewise cubic hermite polynomial defined by the following data:

$$\{x_i\} = \{m, (0.9m + 0.1M), (0.7m + 0.3M), M\}, \quad \{y_i\} = \{1, 1, 0, 0\}, \quad \{s_i\} = \{0, 0, 0, 0\}, \tag{72}$$

where $m = \min(\rho_G)$, $M = \max(\rho_G)$, and $\{s_i\}$ are the slope values corresponding to the $\{(x_i, y_i)\}$ coordinate pairs. Thus, when updating f with $\delta f = \omega(\rho_G)$, the value of $f(\mathbf{x})$ at a point \mathbf{x} is unchanged when $\rho_G(\mathbf{x})$ is much greater than $\underline{\rho}_G$. Otherwise, $f(\mathbf{x})$ increases slightly.

Table 1

Parameter values. For each example, the maximum and minimum step size is given. The value of η used in step (1) and step (2) of the optimization schedule (Section 4.6) is given. The number of iterations used in steps (1) and (2) is also given.

Example	β (max/min)	η	η_ϕ	ϵ_0	ξ	Iterations
Section 5.1.1	$0.03/10^{-6}$	10.0/100.0	1.0	0.01	0.2	434/587
Section 5.1.2	$0.03/10^{-10}$	100.0/200.0	1.0	0.01	0.2	21/356
Section 5.1.3	$0.03/10^{-8}$	10.0/100.0	20.0	0.01	0.02	61/266
Section 5.1.4	$0.03/10^{-8}$	10.0/100.0	10.0	0.05	0.2	28/348
Section 5.1.5	$0.03/10^{-6}$	10.0/100.0	50.0	0.05	0.2	38/342

Remark 7. The heuristic approach of Algorithm 3 avoids differentiating ρ_G . Indeed, in order to maximize ρ_G , a natural approach would be to apply a gradient based optimization method. However, ρ_G is not differentiable with respect to the curve in the usual sense (a more complex notion of differentiability for this problem was recently described in [11]). Our goal here is to present a simple algorithm that gives a decent approximation of an ideal knot.

Remark 8. The computation of ρ_G is approximated, at a nodal point \mathbf{x}_i in Σ_h , by

$$\rho_G(\mathbf{x}_i) = \min \left\{ \min_{\mathbf{y}_i \in \Sigma_h \setminus \{\mathbf{x}_i\}} \mathcal{R}_{pt}(\mathbf{x}_i - \mathbf{y}_i, \boldsymbol{\tau}_h(\mathbf{y}_i)), \frac{1}{|\boldsymbol{\kappa}_h(\mathbf{x}_i)|} \right\}, \tag{73}$$

where the minimization in \mathbf{y}_i was performed over all nodal points in Σ_h such that $\mathbf{y}_i \neq \mathbf{x}_i$. Compare this with (9), (15). Note that $\boldsymbol{\kappa}_h(\mathbf{x}_i)$ is the discrete curvature vector obtained from solving the weak formulation (66). The value of $\boldsymbol{\tau}_h(\mathbf{y}_i)$ is computed directly using the piecewise quadratic curve Σ_h .

5. Numerical results

5.1. Polymers on surfaces

We present several examples of the optimization schedule in Section 4.6. In each case, the parameterization of the initial curve is given as well as the surface on which the curve is to be adsorbed. Moreover, we give the amount by which the constraints are violated, i.e. we give the values of

$$\|B(\Sigma_h)\|_{l^\infty} = \max_{1 \leq j \leq N-1} \left| |E_j| - |E_j^0| \right|, \quad \|C(\Sigma_h)\|_{l^\infty} = \max_{1 \leq j \leq N-1} \left| \int_{E_j} \phi(\mathbf{x}) ds(\mathbf{x}) \right|,$$

in the figure caption for the final iteration. A list of parameters used for all the examples is given in Table 1. The curve thickness was set to $d = 0.2$.

All numerical results were performed in MATLAB using the FELICITY toolbox [54]. The computational time for most of the examples was between 1 and 3 minutes on a Dell desktop computer with an Intel Core i7-4770 CPU at 3.4 GHz, with 32 GB of RAM. The example in Section 5.1.3 took about 4 minutes because a finer discretization was used. The initial step size to try in each iteration of Algorithm 1 was defined to be either twice the step size from the previous iteration or the maximum step size (whichever is smaller). This helped avoid unnecessary initial trial steps.

5.1.1. Coiling onto a sphere

The initial (open) curve Σ_h^0 is parameterized by

$$\boldsymbol{\alpha}_h^0(t) = (2, 0.1 \sin(10\pi t), 45t - 2), \text{ for } t \in [0, 1], \tag{74}$$

with total length of 45.055, and is discretized with 200 edge segments using \mathbb{X}_2 . The surface Γ is a sphere of radius 1.3 centered at the origin.

In order to prevent the curve from getting tangled up as it moves toward the surface in step (1) of the optimization schedule in Section 4.6, we set a Dirichlet boundary condition for the velocity \mathbf{V}_h at the end of the curve farthest from the surface, i.e. we set

$$\mathbf{V}_h|_{\mathbf{x}_0} = -\text{sgn}(\phi(\mathbf{x}_0))\nabla\phi(\mathbf{x}_0),$$

where \mathbf{x}_0 is the point on $\partial\Sigma$ farthest from Γ , and ϕ is the level set function for the surface Γ . In step (2), no Dirichlet condition is applied.

Figs. 4 and 5 show the results of the curve optimization. The surface Γ is shown in light blue. Each edge E_j of Σ_h is a parametric quadratic curve (by the definition of \mathbb{X}_2) defined by the nodal values of $\boldsymbol{\alpha}_h$ restricted to the interval I_j . We plot Σ_h by a thin black curve, with black dots denoting the end points of every edge of Σ_h . The magenta dots represent the nodal values of $\boldsymbol{\alpha}_h$ corresponding to the “midpoint” of the edge. The same format is used in all subsequent figures.

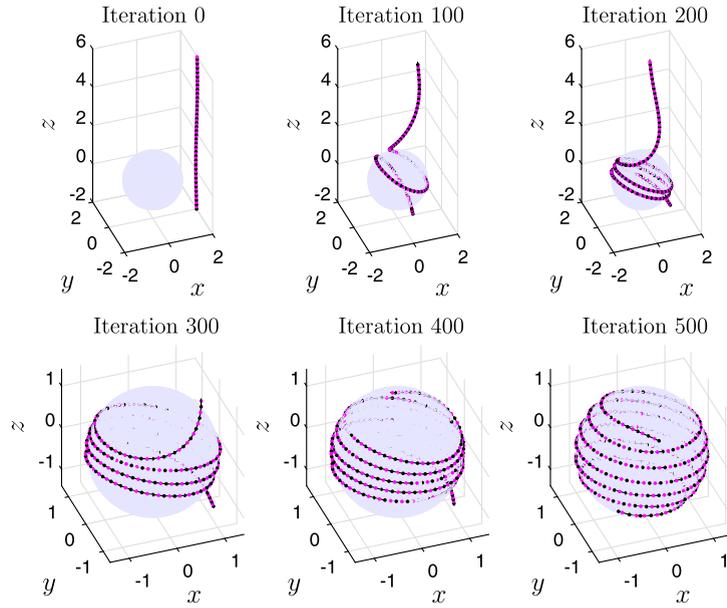


Fig. 4. Optimization of a curve adsorbing onto a sphere (Section 5.1.1). The entire curve is not visible in the first four plots. Without the proximity penalty term, the curve would converge to a great circle with multiple overlaps. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

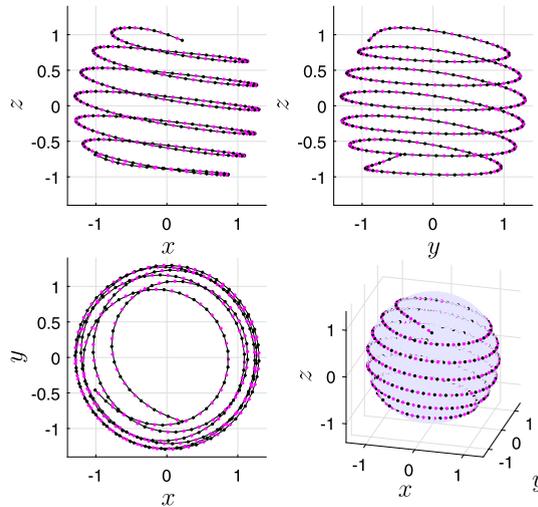


Fig. 5. Final shape for a curve adsorbed onto a sphere (Section 5.1.1). The surface is plotted in the lower right hand figure only. At iteration 1021, the constraints are satisfied to within: $\|B(\Sigma_h)\|_{\infty} = 2.756 \cdot 10^{-7}$, $\|C(\Sigma_h)\|_{\infty} = 6.711 \cdot 10^{-6}$. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

5.1.2. Tennis ball pattern on a sphere

The initial (closed) curve Σ_h^0 is parameterized by

$$\alpha_h^0(t) = (1.2 \cos(2\pi t), 1.2 \sin(2\pi t), 1.8 \sin(6\pi t)), \text{ for } t \in [0, 1], \tag{75}$$

with total length of 23.401, and is discretized with 200 edge segments using \mathbb{X}_2 . The surface Γ is a sphere of radius 1.3 centered at the origin.

Figs. 6 and 7 show the results of the curve optimization.

5.1.3. Tennis ball pattern on a rounded cube

The initial (closed) curve Σ_h^0 is parameterized by

$$\alpha_h^0(t) = (1.0 \cos(2\pi t), 1.0 \sin(2\pi t), 1.2 \sin(8\pi t)), \text{ for } t \in [0, 1], \tag{76}$$

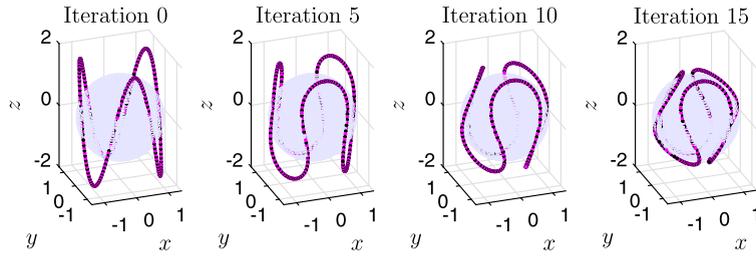


Fig. 6. Optimization of a closed curve adsorbing onto a sphere (Section 5.1.2). Without the proximity penalty term, the curve would converge to a great circle with multiple overlaps.

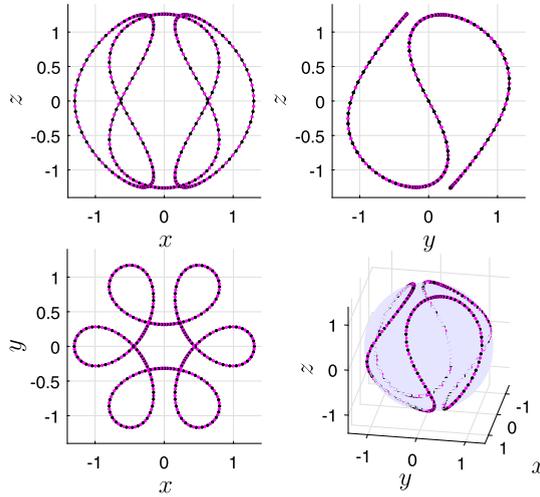


Fig. 7. Final shape for a closed curve adsorbed onto a sphere (Section 5.1.2). The surface is plotted in the lower right hand figure only. Note the symmetric “tennis-ball” pattern. At iteration 377, the constraints are satisfied to within: $\|B(\Sigma_h)\|_{l^\infty} = 3.549 \cdot 10^{-8}$, $\|C(\Sigma_h)\|_{l^\infty} = 5.722 \cdot 10^{-7}$.

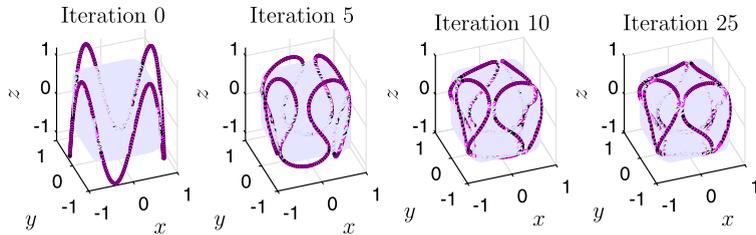


Fig. 8. Optimization of a closed curve adsorbing onto a rounded cube (Section 5.1.3). Without the proximity penalty term, the curve would converge to a loop about a diagonal section of the surface with multiple overlaps.

with total length of 20.635, and is discretized with 400 edge segments using \mathbb{X}_2 . The surface Γ is described by the set of points (x, y, z) that satisfy

$$x^8 + y^8 + z^8 = (0.8)^8. \tag{77}$$

Figs. 8 and 9 show the results of the curve optimization.

5.1.4. Closed curve on an equipotential surface

The initial (closed) curve Σ_h^0 is parameterized by

$$\alpha_h^0(t) = (1.8 \cos(2\pi t), 1.8 \sin(2\pi t), 1.5 \sin(8\pi t)), \text{ for } t \in [0, 1], \tag{78}$$

with total length of 27.316, and is discretized with 200 edge segments using \mathbb{X}_2 . The surface Γ is described by the set of points $\mathbf{x} = (x, y, z)$ that satisfy

$$\sum_{i=1}^4 \frac{1}{\|\mathbf{x} - \mathbf{x}_i\|} = 2.8, \tag{79}$$

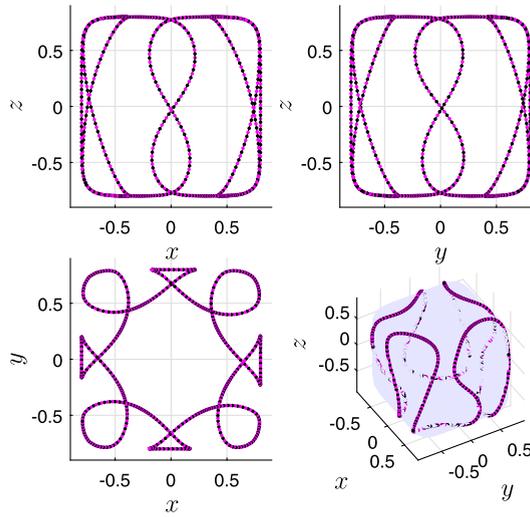


Fig. 9. Final shape for a closed curve adsorbed onto a rounded cube (Section 5.1.3). The surface is plotted in the lower right hand figure only. Note the symmetric pattern analogous to the “tennis-ball” pattern in Section 5.1.2. At iteration 327, the constraints are satisfied to within: $\|B(\Sigma_h)\|_{l^\infty} = 3.351 \cdot 10^{-5}$, $\|C(\Sigma_h)\|_{l^\infty} = 5.055 \cdot 10^{-4}$.

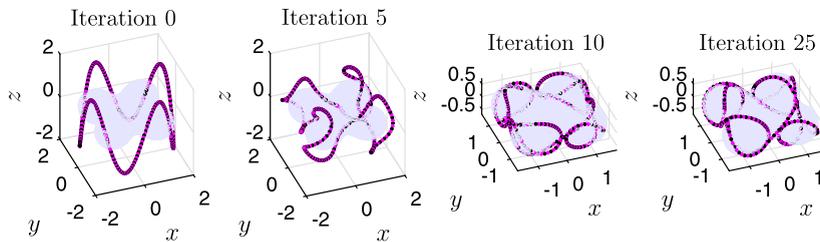


Fig. 10. Optimization of a closed curve adsorbing onto an equipotential surface (Section 5.1.4). Without the proximity penalty term, the curve would converge to a loop about the horizontal mid-plane with multiple overlaps.

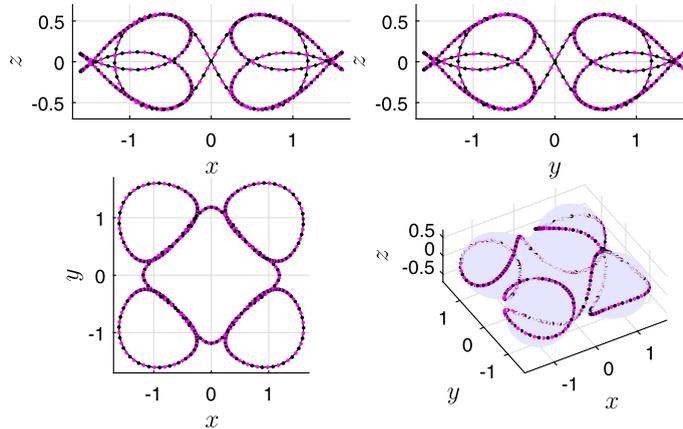


Fig. 11. Final shape for a closed curve adsorbed onto an equipotential surface (Section 5.1.4). The surface is plotted in the lower right hand figure only. The curve achieves a symmetric pattern because the surface is symmetric. At iteration 376, the constraints are satisfied to within: $\|B(\Sigma_h)\|_{l^\infty} = 2.232 \cdot 10^{-6}$, $\|C(\Sigma_h)\|_{l^\infty} = 1.416 \cdot 10^{-4}$.

where

$$\mathbf{x}_1 = (1, 1, 0), \mathbf{x}_2 = (-1, 1, 0), \mathbf{x}_3 = (-1, -1, 0), \mathbf{x}_4 = (1, -1, 0).$$

Figs. 10 and 11 show the results of the curve optimization.

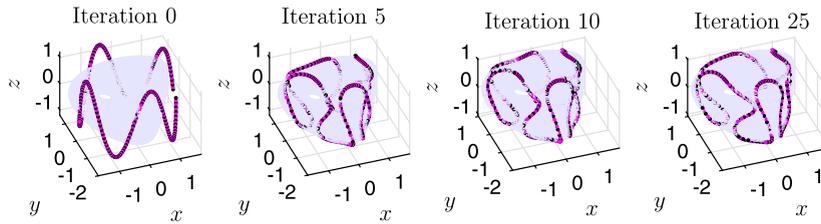


Fig. 12. Optimization of an open curve adsorbing onto a genus-2 surface (Section 5.1.5).

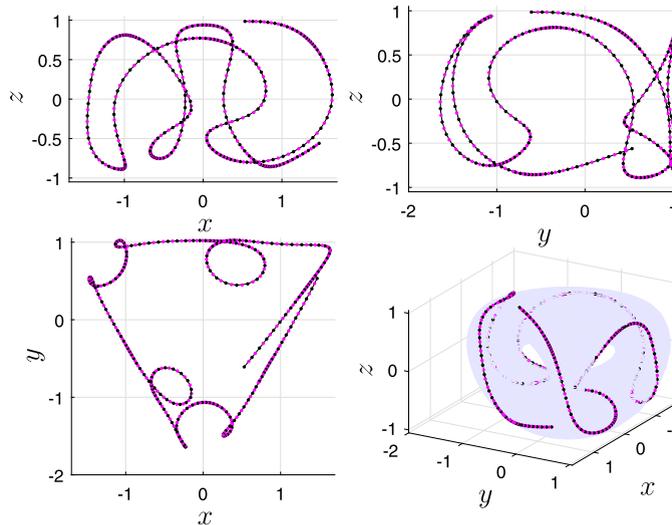


Fig. 13. Final shape for an open curve adsorbed onto a genus-2 surface (Section 5.1.5). The surface is plotted in the lower right hand figure only. The shape is not symmetric because the initial curve was not symmetric. At iteration 380, the constraints are satisfied to within: $\|B(\Sigma_h)\|_{l^\infty} = 1.426 \cdot 10^{-4}$, $\|C(\Sigma_h)\|_{l^\infty} = 1.090 \cdot 10^{-3}$.

5.1.5. Open curve on a genus-2 surface

The initial (open) curve Σ_h^0 is parameterized by

$$\alpha_h^0(t) = (1.5 \cos(1.9\pi t), 1.5 \sin(1.9\pi t), 1.2 \sin(8\pi t)), \text{ for } t \in [0, 1], \tag{80}$$

with total length of 21.807, and is discretized with 200 edge segments using \mathbb{X}_2 . The surface Γ has genus equal to 2 and is described by the set of points $\mathbf{x} = (x, y, z)$ that satisfy

$$2y(y^2 - 3x^2)(1 - z^2) + (x^2 + y^2)^2 - (9z^2 - 1)(1 - z^2) = 0.2. \tag{81}$$

Figs. 12 and 13 show the results of the curve optimization.

5.1.6. Effect of initial condition and penalty parameter

The particular initial condition can affect the equilibrium solution produced by Algorithm 1. For instance, in Section 5.1.1, one can simply rotate the initial condition, and the equilibrium solution will obviously rotate as well (because of spherical symmetry). Moreover, the optimization problem is non-convex so it is expected that minimizers are not unique. For example, in Section 5.1.5, rotating the initial curve by 90° (about the z-axis) leads to a very different minimizing shape.

The effect of η appears to be rather mild. For the case in Section 5.1.4, we tried three different values: $\eta = 50, 100, 200$. The final equilibrium shape was the same for all three values, but the number of iterations in step (2) of the optimization schedule (Section 4.6) varied: 737, 348, 389. However, it is still conceivable that changing η can lead to different minimizing shapes, especially if the surface has lots of folds and twists.

5.2. Inflating the trefoil knot

We demonstrate Algorithm 3 on the trefoil knot. The initial curve used for Σ_h^0 in the algorithm is the trefoil knot parameterized by

$$\alpha_h^0(t) = 0.5((\sin(2\pi t) + 2 \sin(4\pi t)), (\cos(2\pi t) - 2 \cos(4\pi t)), -\sin(6\pi t)), \text{ for } t \in [0, 1], \tag{82}$$

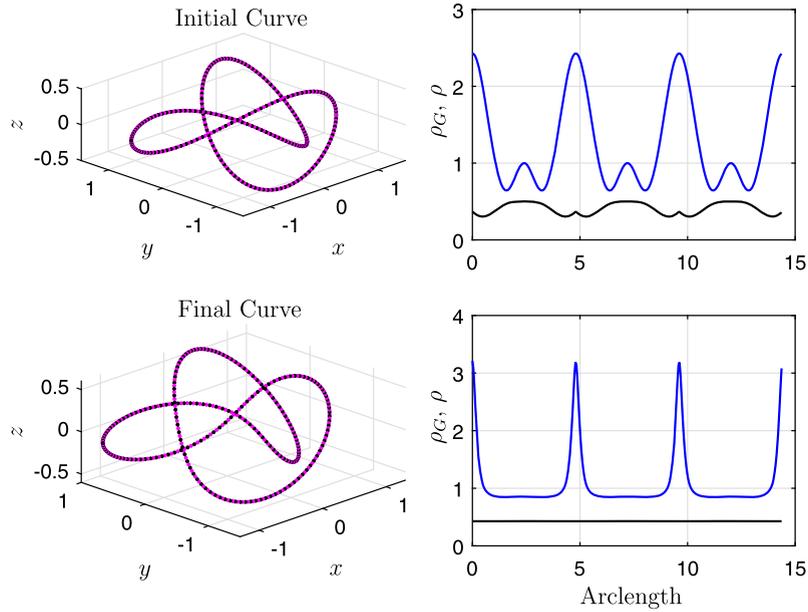


Fig. 14. Inflating the trefoil knot (Section 5.2). The initial curve and final curve are shown. The local radius of curvature ρ is plotted in blue; the global radius of curvature ρ_G is in black. Note: $\rho_G < \rho$ over the entire curve. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

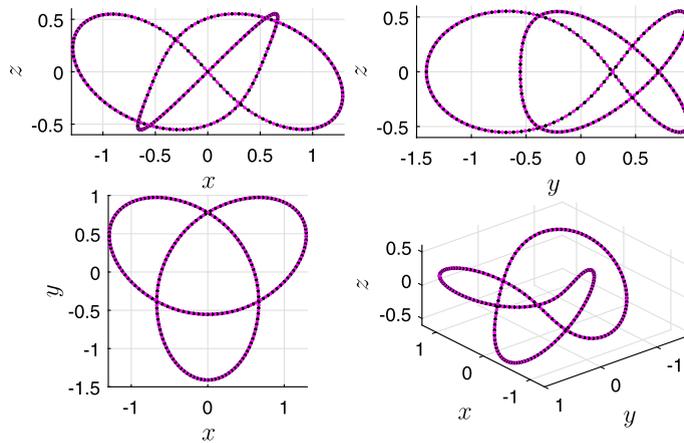


Fig. 15. Final shape of the trefoil knot.

with total length of 14.413, and is discretized with 200 edge segments using \mathbb{X}_2 . We ran Algorithm 3 for a total of 56 iterations until the step size dropped below the minimum allowed. During this process, we must repeatedly find a minimizer of (71). For this, we set $d = 0.5$ and $\eta = 10.0$. The total number of gradient descent iterations used to minimize (71) was 1126. The overall simulation time was 5.66 minutes.

Fig. 14 shows the initial curve and the final curve produced by Algorithm 3, as well as plots of ρ_G and ρ for both curves. Fig. 15 shows different views of the final curve.

Fig. 16 shows the results of the knot “inflation” algorithm. An ideal knot is characterized by its so-called *rope-length*:

$$\text{rope-length of } \Sigma = \frac{|\Sigma|}{\rho_G(\Sigma)}.$$

An ideal knot has minimal rope-length. This is plotted in Fig. 16, which clearly shows the rope-length being reduced. The initial rope-length is 37.4971539641; the final value is 33.7484406674. A very good estimate of the true minimum value is provided in [31,44]: 32.744459376, which is accurate to the number of digits given. Choosing different weights for the $\{x_i\}$ data in (72) does not significantly change the final rope-length we compute; worst case was 34.168787 corresponding to $\{x_i\} = \{m, (0.7m + 0.3M), (0.4m + 0.6M), M\}$.

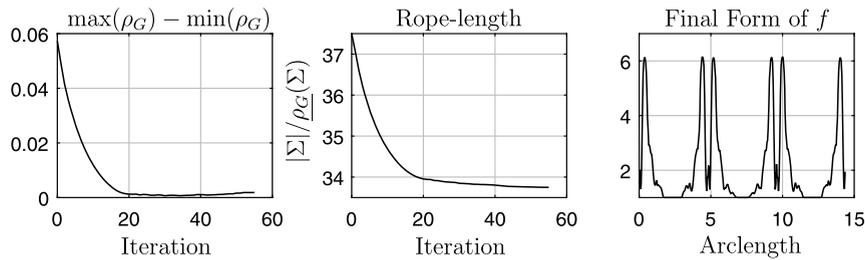


Fig. 16. Knot inflation data.

We also plot the difference $\max(\rho_G) - \min(\rho_G)$ in Fig. 16 to show how the algorithm pushes ρ_G closer to a constant. At the final iteration, $\max(\rho_G) - \min(\rho_G) = 1.8394 \cdot 10^{-3}$. The final form of the function f is also shown.

Remark 9 (Effect of η). The penalty parameter does have some effect on the final solution. The following table lists the values of the rope-length depending on η .

η	1	5	10	20	50
rope-length	66.8283	33.7285	33.7484	34.1993	34.6857

If η is too small, then self-intersection is not sufficiently penalized, i.e. self-intersection is practically ignored. This is why the rope-length is so large when $\eta = 1$. We found that when η is large, the algorithm may end prematurely (i.e. takes too few iterations). This was the case for $\eta = 50$. Thus, if η is large, then the minimum step size should be reduced.

5.3. Conclusion

The main result of this paper is a softened notion of proximity presented in Sections 2.2 and 2.3. We then computed equilibrium curve shapes that minimize an energy consisting of a bending energy and a penalized proximity functional (to enforce no-overlap of the curves). Our algorithm, based on an SQP approach, is able to compute complex curve shapes on a variety of surfaces. Many of the shapes are reminiscent of Euler's elastica (see Figs. 7 and 9).

5.3.1. Self-contacting bodies

We emphasize that minimizing the bending energy of self-avoiding curves, restricted to arbitrary surfaces, is a non-convex problem. There may be many local, constrained minimizers, which our algorithm picks out only one. The same issue is present when using molecular dynamics (MD) with “beads on a string”, such as in [62]. However, the number of time-steps used in MD simulations is on the order of 1 to 10 million, whereas our method takes on the order of hundreds to about a thousand for number of gradient descent steps. Thus, our method should be useful for finding interesting minimizing configurations for semi-flexible polymers adsorbed onto surfaces. One observation we noticed in the simulations was that η_ϕ (in step (1) of Section 4.6) should be chosen larger when the surface has regions of high curvature. This is to overcome the curve's bending stiffness in order to force the curve onto the surface. One possible extension of our method is to allow the surface to deform (e.g. treat the surface as a bio-membrane).

One extension of the method is to replace the penalization approach with an inequality constraint on (28) via a Lagrange multiplier. This would require combining a variational inequality [28] on (39) with the bending energy functional. In addition, it would be interesting if the “softening” of self-contact in (28) could be modified in order to better approximate the “hard” limit (see Section 2.3.1 and Remark 5).

The method could be easily modified for surfaces, i.e. to enforce that a bendable surface does not intersect itself; applications here could be in bio-membranes and cloth simulation. Furthermore, this method could be combined with simulating highly deformable elastic bodies, such as swimming organisms and locomoting snakes, to ensure they maintain the constraint of no self-intersection.

5.3.2. Approximating ideal knots

The heuristic approach in Section 4.7 for finding ideal knot shapes may be a useful sub-component of an overall method. The final shape found is not an ideal knot, but it is close (see Section 5.2 for the actual data). Many highly accurate methods exist for computing minimal rope-length (see [31,44] and [3] for instance). However, they are very expensive methods because the ideal knot optimization problem is not smooth. For example, [3] mentions that a hundred thousand iterations is a standard run. Thus, our scheme is a good option for providing a decent initial guess, at low computational cost, to a more accurate method.

It is not clear how to formally justify the heuristic algorithm in Section 4.7.2. Most likely, a different approach would be necessary. One option is to first adopt the variational inequality approach mentioned in Section 5.3.1 because this eliminates

the penalization parameter. Moreover, it avoids situations where self-contact may be *over-penalized* when the curve has a large set of “points of closest approach” that are equal distance to a single point (see Fig. 2(a) for an example); this is a common situation for ideal knots, such as the trefoil knot.

Using an inequality approach for (39) implies that modifying the proximity penalty functional in (69) would no longer work; recall that we used f in (69) to “inflate” the knot. Instead, one could introduce a variable bending rigidity coefficient into (38) that would become the control variable to optimize to drive the curve to an ideal knot shape. Hence, in order to justify this new algorithm, we need to better understand the effect of the bending energy on the shape of the knot. A key step in this direction would be to derive a sensitivity result connecting the (variable) bending coefficient to the equilibrium shape of a curve that minimizes its bending energy under an inequality constraint on (28). This could potentially provide a new means for computing ideal knots.

Acknowledgements

The author acknowledges financial support by the National Science Foundation through the following grant: DMS-1418994.

References

- [1] D.G. Angelescu, P. Linse, T.T. Nguyen, R.F. Bruinsma, Structural transitions of encapsidated polyelectrolytes, *Eur. Phys. J. E* 25 (3) (2008) 323–334.
- [2] T. Ashton, J. Cantarella, A fast octree-based algorithm for computing ropelength, in: *Physical and Numerical Models in Knot Theory and Their Application to the Life Sciences*, World Scientific Press, 2005, pp. 323–341.
- [3] T. Ashton, J. Cantarella, M. Piatek, E.J. Rawdon, Knot tightening by constrained gradient descent, *Exp. Math.* 20 (1) (2011) 57–90.
- [4] J.W. Barrett, H. Garcke, R. Nürnberg, A parametric finite element method for fourth order geometric evolution equations, *J. Comput. Phys.* 222 (1) (2007) 441–467.
- [5] J.W. Barrett, H. Garcke, R. Nürnberg, Parametric approximation of Willmore flow and related geometric evolution equations, *SIAM J. Sci. Comput.* 31 (1) (2008) 225–253.
- [6] A. Bonito, R.H. Nochetto, M. Sebastian Pauletti, Parametric fem for geometric biomembranes, *J. Comput. Phys.* 229 (9) (May 2010) 3171–3188.
- [7] D. Braess, *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*, 2nd ed., Cambridge University Press, 2001.
- [8] S.C. Brenner, L.R. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Texts in Applied Mathematics, vol. 15, Springer, New York, NY, 2008.
- [9] G. Buck, J. Simon, Energy and length of knots, in: S. Suzuki (Ed.), *Lectures at Knots '96*, Lectures Delivered to the International Conference on Knot Theory, Tokyo, World Scientific, 1997, pp. 219–235.
- [10] M. Burger, A framework for the construction of level set methods for shape optimization and reconstruction, *Interfaces Free Bound.* 5 (2002) 301–329.
- [11] J. Cantarella, J.H.G. Fu, R.B. Kusner, J.M. Sullivan, Ropelength criticality, *Geom. Topol.* 18 (4) (2014) 1973–2043.
- [12] J.J. Cerdà, T. Sintès, A. Chakrabarti, Excluded volume effects on polymer chains confined to spherical surfaces, *Macromolecules* 38 (4) (2005) 1469–1477.
- [13] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, 2nd ed., Classics in Applied Mathematics, SIAM, Philadelphia, PA, ISBN 978-0898715149, 2002.
- [14] K. Deckelnick, G. Dziuk, C.M. Elliott, Computation of geometric partial differential equations and mean curvature flow, *Acta Numer.* 14 (1) (2005) 139–232.
- [15] M.C. Delfour, J.-P. Zolésio, *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*, 2nd ed., Advances in Design and Control, vol. 4, SIAM, 2011.
- [16] M.P. do Carmo, *Differential Geometry of Curves and Surfaces*, Prentice Hall, Upper Saddle River, New Jersey, 1976.
- [17] Q. Du, C. Liu, R. Ryham, X. Wang, A phase field formulation of the Willmore problem, *Nonlinearity* 18 (3) (2005) 1249.
- [18] G. Dziuk, An algorithm for evolutionary surfaces, *Numer. Math.* 58 (1) (1990) 603–611.
- [19] O. Gonzalez, J.H. Maddocks, Global curvature, thickness and the ideal shapes of knots, *Proc. Natl. Acad. Sci. USA* 96 (1999) 4769–4773.
- [20] O. Gonzalez, J.H. Maddocks, F. Schuricht, H. von der Mosel, Global curvature and self-contact of nonlinearly elastic curves and rods, *Calc. Var.* 14 (2002) 29–68.
- [21] M.D. Gunzburger, *Perspectives in Flow Control and Optimization*, SIAM, 2003.
- [22] I.W. Hamley, Nanotechnology with soft materials, *Angew. Chem., Int. Ed. Engl.* 42 (15) (2003) 1692–1712.
- [23] J. Haslinger, R.A.E. Mäkinen, *Introduction to Shape Optimization: Theory, Approximation, and Computation*, Advances in Design and Control, vol. 7, SIAM, 2003.
- [24] K.J. Huerter, Non-uniform thickness and weighted global radius of curvature of smooth curves, PhD thesis, University of Iowa, Dec 2009.
- [25] K. Ito, K. Kunisch, *Lagrange Multiplier Approach to Variational Problems and Applications*, Advances in Design and Control, SIAM, 2008.
- [26] J. Jost, X. Li-Jost, *Calculus of Variations*, Cambridge, 1998.
- [27] E.E. Keaveny, S.W. Walker, M.J. Shelley, Optimization of chiral structures for microscale propulsion, *Nano Lett.* 13 (2) (2013) 531–537.
- [28] D. Kinderlehrer, G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, SIAM, Philadelphia, PA, 1987.
- [29] O. Krötenheerdt, S. Veit, Zur theorie massiver knoten (on the theory of knots with thickness), *Beitr. Algebra Geom.* 5 (1976) 61–74.
- [30] R.B. Kusner, J.M. Sullivan, On distortion and thickness of knots, in: *Topology and Geometry in Polymer Science*, Springer, 1998, pp. 67–78.
- [31] B. Laurie, J. Smutny, M. Carlen, J.H. Maddocks, Biarcs, Global Radius of Curvature, and the Computation of Ideal Knot Shapes, *Series on Knots and Everything*, vol. 36, World Scientific Publishing, Sept 2005, pp. 75–108, chapter 5.
- [32] Z. Lin, *Evaporative Self-Assembly of Ordered Complex Structures*, World Scientific Publishing Company, 2012.
- [33] R. Lipowsky, H.-G. Döbereiner, C. Hiergeist, V. Indrani, Membrane curvature induced by polymers and colloids, *Phys. A, Stat. Mech. Appl.* 249 (1–4) (1998) 536–543.
- [34] R. Litherland, J. Simon, O. Durumeric, E. Rawdon, Thickness of knots, *Topol. Appl.* 91 (3) (1999) 233–244.
- [35] K. Luger, A.W. Mader, R.K. Richmond, D.F. Sargent, T.J. Richmond, Crystal structure of the nucleosome core particle at 2.8 Å resolution, *Nature* 389 (Sept 1997) 251–260.
- [36] K.C. Millett, E.J. Rawdon, Energy, ropelength, and other physical aspects of equilateral knots, *J. Comput. Phys.* 186 (2) (Apr 2003) 426–456.
- [37] T. Neuhaus, O. Zimmermann, U.H.E. Hansmann, Ring polymer simulations with global radius of curvature, *Phys. Rev. E* 75 (May 2007) 051803.
- [38] J. Nocedal, S.J. Wright, *Numerical Optimization*, 2nd ed., Springer Series in Operations Research, Springer, 2006.
- [39] P. Pierański, *Series on Knots and Everything*, Search of Ideal Knots, vol. 19, World Scientific Publishing, River Edge, NJ, Dec 1998, pp. 20–41, chapter 2.

- [40] E.J. Rawdon, J.K. Simon, Polygonal approximation and energy of smooth knots, *J. Knot Theory Ramif.* 15 (4) (2006) 429–451.
- [41] T.J. Richmond, C.A. Davey, The structure of dna in the nucleosome core, *Nature* 423 (May 2003) 145–150.
- [42] F. Schuricht, H.v.d. Mosel, Euler–Lagrange equations for nonlinearly elastic rods with self-contact, *Arch. Ration. Mech. Anal.* 168 (1) (2003) 35–82.
- [43] A. Schweikart, A. Fortini, A. Wittemann, M. Schmidt, A. Fery, Nanoparticle assembly by confinement in wrinkles: experiment and simulations, *Soft Matter* 6 (2010) 5860–5863.
- [44] J. Smutny, Global radii of curvature, and the biarc approximation of space curves: in pursuit of ideal knot shapes, PhD thesis, Thèse École Polytechnique Fédérale de Lausanne (EPFL), 2004.
- [45] J. Sokolowski, J.-P. Zolésio, Introduction to Shape Optimization, Springer Series in Computational Mathematics, Springer-Verlag, 1992.
- [46] A.J. Spakowitz, Z.-G. Wang, Semiflexible polymer confined to a spherical surface, *Phys. Rev. Lett.* 91 (Oct 2003) 166102.
- [47] P. Strzelecki, H. von der Mosel, Global curvature for surfaces and area minimization under a thickness constraint, *Calc. Var. Partial Differ. Equ.* 25 (4) (2006) 431–467.
- [48] P. Strzelecki, H. von der Mosel, On rectifiable curves with L^p -bounds on global curvature: self-avoidance, regularity, and minimizing knots, *Math. Z.* 257 (1) (2007) 107–130.
- [49] P. Strzelecki, H. von der Mosel, Tangent-point repulsive potentials for a class of non-smooth m -dimensional sets in \mathbb{R}^n . Part I: smoothing and self-avoidance effects, *J. Geom. Anal.* 23 (3) (2013) 1085–1139.
- [50] G.M. Sumida, S. Yamada, Self-contact elimination by membrane fusion, *Proc. Natl. Acad. Sci.* 110 (47) (2013) 18958–18963.
- [51] F. Tröltzsch, Optimal Control of Partial Differential Equations, Graduate Studies in Mathematics, American Mathematical Society, April 2010.
- [52] K. van Holde, J. Zlatanova, Chromatin fiber structure: where is the problem now?, *Semin. Cell Dev. Biol.* 18 (5) (2007) 651–658, Membrane Lipid Microdomains: Roles in Signalling and Disease and 3D Chromatin Structure Inside the Cell Nucleus.
- [53] A. Šarić, Self-assembly of nanoparticles on fluid and elastic membranes, PhD thesis, Columbia University, 2013.
- [54] S.W. Walker, FELICITY: Finite Element Implementation and Computational Interface Tool for You, <http://www.mathworks.com/matlabcentral/fileexchange/31141-felicity>.
- [55] S.W. Walker, The Shapes of Things: A Practical Guide to Differential Geometry and the Shape Derivative, 1st ed., Advances in Design and Control, vol. 28, SIAM, 2015.
- [56] S.W. Walker, A. Bonito, R.H. Nochetto, Mixed finite element method for electrowetting on dielectric with contact line pinning, *Interfaces Free Bound.* 12 (1) (March 2010) 85–119.
- [57] S.W. Walker, E.E. Keaveny, Analysis of shape optimization for magnetic microswimmers, *SIAM J. Control Optim.* 51 (4) (2013) 3093–3126.
- [58] S.W. Walker, M.J. Shelley, Shape optimization of peristaltic pumping, *J. Comput. Phys.* 229 (4) (Feb 2010) 1260–1291.
- [59] J. Wang, H. Gao, Brownian dynamics simulations of charged semiflexible polymers confined to curved surfaces, *J. Mech. Behav. Biomed. Mater.* 4 (2) (2011) 174–179, Multiscale Mechanics.
- [60] T.J. Willmore, Total mean curvature squared of surfaces, in: *Geometry and Topology of Submanifolds, VIII*, Brussels, 1995/Nordfjordeid, 1995, 1996, pp. 383–391.
- [61] M.J. Zakhary, P. Sharma, A. Ward, S. Yardimici, Z. Dogic, Geometrical edgeactants control interfacial bending rigidity of colloidal membranes, *Soft Matter* 9 (2013) 8306–8313.
- [62] D. Zhang, A. Chai, X. Wen, L. He, L. Zhang, H. Liang, Ordered regular pentagons for semiflexible polymers on soft elastic shells, *Soft Matter* 8 (2012) 2152–2158.