

Visualizing the tightening of knots

Jason Cantarella*
University of Georgia

Michael Piatek†
University of Washington

Eric Rawdon‡
Duquesne University

ABSTRACT

The study of physical models for knots has recently received much interest in the mathematics community. In this paper, we consider the ropelength model, which considers knots tied in an idealized rope. This model is interesting in pure mathematics, and has been applied to the study of a variety of problems in the natural sciences as well.

Modeling and visualizing the tightening of knots in this idealized rope poses some interesting challenges in computer graphics. In particular, self-contact in a deformable rope model is a difficult problem which cannot be handled by standard techniques. In this paper, we describe a solution based on reformulating the contact problem and using constrained-gradient techniques from nonlinear optimization. The resulting animations reveal new properties of the tightening flow and provide new insights into the geometric structure of tight knots and links.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: collision detection, contact, flexible models, tight knots, ideal knots, ropelength, nonlinear optimization, constrained least squares

1 INTRODUCTION AND BACKGROUND

Over the past decade, there has been considerable interest in mathematical models of physical knots. One of the more popular models proposed has been the *ropelength* model, which considers knots tied in perfectly flexible tubes of constant circular cross-section. The diameter of the cross-section is known as the *thickness* of the knot, and knots of minimum length for fixed thickness are called *tight* or *ideal* knots. This minimum ropelength is a numerical invariant describing the complexity of different types of knots. The model has proved useful in scientific applications ranging from the study of DNA strands [39] to knot-like subatomic structures known as “glueballs” [6]. Although the exact value of the minimum ropelength is known only for a few links, good upper bounds on minimum ropelength can be obtained by constructing approximately tight knots and links by computer methods. As a consequence, virtually all of these scientific applications demand high-quality computations of tight shapes for hundreds or thousands of knot types.

An exciting recent development in this field of mathematics has been the formulation of a kind of Euler-Lagrange equation describing length-critical knots in terms of the set of self-contacts of their tubes [7, 16, 38]. This theory has allowed us to make some conjectures about the tightening process, but these conjectures have

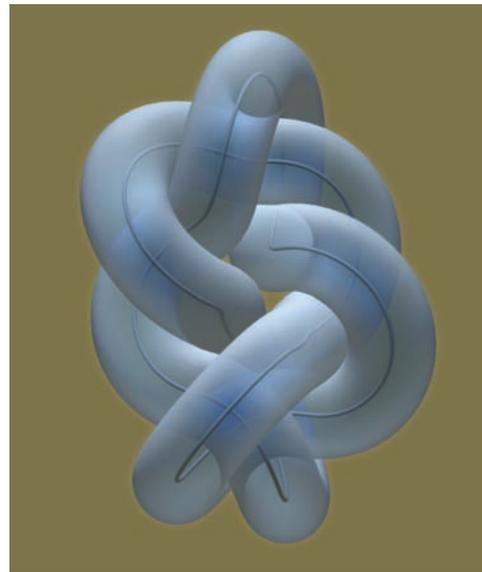


Figure 1: A tight 7.2 knot as computed using our method.

turned out to be difficult to prove without an intuitive understanding of knot tightening. These developments motivated us to pursue the simulation and visualization of the (conjectured) knot tightening flow. We see two main scientific benefits from our effort. First, the resulting software package, *RidgeRunner*, provides a virtual laboratory for mathematicians interested in developing and testing their understanding of tightening, along with new visualizations of the tightening process. In this way, we hope it will aid in proving theorems about tight knots. Second, the final results of the tightening simulation have expanded the scope and quality of existing computations of tight configurations of knots and links [10, 19, 30]. We hope that our data set will enable new applications of the theory of tight knots in the sciences.

Since we are interested in visualizing the behavior of a particular mathematical model for rope, we cannot reuse existing codes with their own formulations of the rope problem. Thus the real-time rope simulators of [4] and [28] are a source of inspiration, but do not solve the problem at hand.

Instead, our simulation applies a formulation of contact as a linear complementarity problem (cf. [3, 24, 22]) to a deformable model for rope. This appears to be a new use of an LCP to model contact. Rigid-body contact has also been formulated as an LCP, and [27] used a different LCP to model contact between slightly deformable or “quasi-rigid” objects. In all of these cases (rigid, quasi-rigid, and fully deformable), the contact problem is modeled by a system of inequality constraints and the velocity of the system is modified to respect these constraints to first order. While our work is similar in spirit to these previous efforts, it is not directly comparable: the mathematical tightening process we animate is not a model of the (second-order) dynamics of a rope accelerating and decelerating under the influence of applied forces. Instead, our tightening flow, like the curve-shortening flow [12], is a first-order ODE which approximates traditional dynamics only for a very lightweight or

*e-mail: cantarel@math.uga.edu, supported by NSF DMS #0204862

†e-mail: piatek@cs.washington.edu, supported by NSF DMS #0311010

‡e-mail: rawdon@mathcs.duq.edu, supported by NSF DMS #0311010

slow-moving rope.

The primary attraction of these strategies is that they provide an extremely robust framework for handling the difficult problems that arise when many contacts happen at the same time. In our application, the correction of a slightly interpenetrating position of a tight knot can often require a coordinated global motion of the entire model. We generate these motions using a Newton's method strategy which (like that of [23]) is inspired by the methods of mathematical optimization. The result is very effective numerical control over rope interpenetration in challenging circumstances.

The main mathematical problem in applying an LCP formulation of contact to our model is in deriving an appropriate set of inequality constraints for self-contact. Our paper starts with a new approach to this problem for the ropelength model. Having done so, our conjectured tightening motion is the trajectory of the method of constrained gradient-descent [32, 14] from numerical optimization. We briefly review the relevant material, including some mathematical contributions, and turn to our implementation. This includes the development of `tsnnls` [9], a new solver for the sparse non-negative least squares problem using the block-principal pivoting algorithm of [31]. Finally, we present an overview of the results of our simulations and a discussion of the animations which accompany our paper.

2 THE ROPELENGTH MODEL

Mathematically, we can think of a tube around a C^2 curve γ as the union of the collection of disks of fixed diameter ε that are centered on the points of γ and normal to the corresponding tangent vectors of γ . If we choose ε very large, these disks could overlap with one another, but for sufficiently small ε , the disks are disjoint. The maximum diameter for which the interiors of these disks are disjoint from one another is called the *thickness* $\text{Thi}(\gamma)$ of the core curve γ . This idea of thickness was first proposed in [18], and was rediscovered more recently by [25] and [5].

In the mathematical study of tight knots, a "rope" is a curve whose thickness is bounded below by some τ . The knot-tightening problem is then to find minimal length curves that obey this bound. It is known that such minimizing curves exist for every kind of knot and link, but their exact shapes are currently the subject of active mathematical research (cf. [8, 15, 16]). Interestingly, the curves of minimum ropelength in a given knot type are not always congruent: some links are known to have a family of tight realizations with different shapes.

It is clear that a thickness bound places constraints on the curvature of the core curve, since a core that turns too fast will cause nearby disks to "focus" on one another and intersect. And it is clear that a thickness bound places some constraints on pairwise distances between points on the curve. However, the precise formulation of these constraints is subtle: it is clear that some pairs of points on γ must remain at distance $\geq \tau$ from each other, but other pairs (such as pairs very near to one another along the curve) will always be much closer together.

We can explain these distance and curvature constraints by giving two alternate definitions of thickness. In what follows, we will think of the parametrized curve $\gamma(s)$ as a C^2 map from $S^1 \rightarrow \mathbb{R}^3$. The self-distance function $d : S^1 \times S^1 \rightarrow \mathbb{R}$ of γ is then defined by $d(s, t) = \|\gamma(s) - \gamma(t)\|$. We let $\kappa(s)$ denote the curvature on γ .

We define the set $\text{dcsd}(\gamma)$ of *doubly-critical self-distances* to be the set of critical points of d with $s \neq t$. Taking the partial derivatives of d , we see that $(s, t) \in \text{dcsd}(\gamma)$ if and only if

$$\langle \gamma(s) - \gamma(t), \gamma'(s) \rangle = 0 \text{ and } \langle \gamma(s) - \gamma(t), \gamma'(t) \rangle = 0.$$

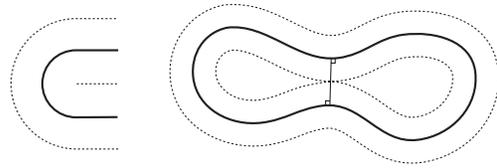


Figure 2: The thickness of a smooth curve γ is controlled by curvature (as in the left picture), and the length of chords in $\text{dcsd}(\gamma)$ (as in the right picture).

where $\langle \cdot, \cdot \rangle$ denotes the dot product.

The key idea of [20] is that for any $\varepsilon < \text{Thi}(\gamma)$, the surface of the tube of diameter ε around γ has no self-intersections and is C^2 smooth. When $\varepsilon = \text{Thi}(\gamma)$, however, the tube forms a pinch or a tangential self-intersection. This leads to our first alternate definition of thickness:

Theorem 1. [20] *The thickness of γ is the minimum of*

$$\min_s \frac{2}{\kappa(s)} \text{ and } \min_{(s,t) \in \text{dcsd}(\gamma)} d(s,t).$$

The two cases in this theorem are shown in Figure 2. A consequence of this theorem is that

$$\text{Thi}(\gamma) \geq \tau \iff \kappa(s) \leq \frac{2}{\tau} \text{ and } d(s,t) \geq \tau \text{ for } (s,t) \in \text{dcsd}(\gamma). \quad (1)$$

This curvature bound is easy to work with. But these distance bounds are not quite satisfactory, as we have only an implicit definition of the set $\text{dcsd}(\gamma)$ to work with. However, Rawdon observed that since the curvature of γ is bounded above, the arclength distance between pairs (s, t) in $\text{dcsd}(\gamma)$ can be bounded below. Doing this precisely allowed him to prove that points with arclength distance less than or equal to $\pi\tau/2$ cannot be in $\text{dcsd}(\gamma)$ [35]. This yields the following theorem.

Theorem 2. *For a C^2 curve γ , let $\text{MinRad}(\gamma) = 1/\max_s \kappa(s)$ be the minimum radius of curvature of a curve γ , and $|s - t|$ be the (shorter) arclength distance between $\gamma(s)$ and $\gamma(t)$.*

Then the thickness of γ is given by the minimum of

$$2\text{MinRad}(\gamma) \text{ and } d(s,t) \text{ for } |s - t| \geq \pi\text{MinRad}(\gamma).$$

We can slightly alter the proof of this theorem to restate (1) as

$$\text{Thi}(\gamma) \geq \tau \iff \kappa(s) \leq \frac{2}{\tau} \text{ and } d(s,t) \geq \tau \text{ for } |s - t| \geq \pi\tau/2. \quad (2)$$

This fixed collection of constraints is the inspiration for our polygonal rope animation model. Our goal now is to formulate a corresponding theorem for a polygonal model of thickness.

2.1 The definition of polygonal thickness

We now focus our attention on space polygons instead of space curves. We denote such a polygon by \mathcal{V} , and let V be the number of vertices of \mathcal{V} . Since \mathcal{V} is specified by the locations in space of these vertices, we will often think of \mathcal{V} as a point in \mathbb{R}^{3V} . To interpret (2) for such curves, we must correctly formulate a polygonal version of curvature. At any vertex v_i of the polygon, we define the polygonal radius of curvature $1/\kappa_i$ to be the radius of the (unique)

circle tangent to both adjacent edges and passing through the mid-point of the shorter one. If θ_i is the turning angle of the polygon at v_i (that is, the angle between the vectors $v_{i+1} - v_i$ and $v_i - v_{i-1}$), and $\min\{|e_{i-1}|, |e_i|\}$ is the length of the shorter edge, then

$$\frac{1}{\kappa_i} = \frac{\min\{|e_{i-1}|, |e_i|\}}{2 \tan(\theta_i/2)}. \quad (3)$$

We now define a thickness measure for polygons by a statement corresponding to that of Theorem 1.

Definition 1. Let $\text{dcsd}(\mathcal{V})$ be the set of (p, q) on \mathcal{V} with $p \neq q$ which are local minima, local maxima, or (where d is differentiable) critical points of the self-distance function on \mathcal{V} .

The thickness $\text{Thi}_p(\mathcal{V})$ of a space polygon \mathcal{V} is given by the minimum of

$$\min_i \frac{2}{\kappa_i} \quad \text{and} \quad \min_{(p,q) \in \text{dcsd}(\mathcal{V})} d(p, q).$$

When n -edge polygons \mathcal{V}_n are inscribed in a space curve γ , [33, 34, 36] have established that $\text{Thi}_p(\mathcal{V}_n) \rightarrow \text{Thi}(\gamma)$ as $n \rightarrow \infty$ under some mild geometric hypotheses.

2.2 Our formulation of the tube constraint

To express the condition $\text{Thi}_p(\mathcal{V}) \geq \tau$ as a finite system of inequality constraints, we now provide a polygonal analogue of (2) where distances between points are replaced by distances between edges and sketch its proof.

Definition 2. For two points p and q on a polygonal curve \mathcal{V} , let $\text{vb}(p, q)$ be the minimum number of vertices on the two polygonal arcs connecting p and q (including p and/or q if one or both are vertices).

Then we have the following theorem.

Theorem 3. Let \mathcal{V} be an equilateral polygonal curve with edge length ℓ . We define a set of pairs of edges (e_i, e_j) of \mathcal{V} by

$$\text{VB}_\tau(\mathcal{V}) = \{(e_i, e_j) \mid \text{vb}(p_i, p_j) \geq \frac{\pi}{2 \arctan(\frac{2}{\tau})} \text{ for } p_{i,j} \in e_{i,j}\}.$$

Then $\text{Thi}_p(\mathcal{V}) \geq \tau \iff$

$$\min_i 2/\kappa_i \geq \tau \quad \text{and} \quad d(e_i, e_j) \geq \tau \text{ for } (e_i, e_j) \in \text{VB}_\tau(\mathcal{V}). \quad (4)$$

where $d(e_i, e_j)$ is the (minimum) distance between edges e_i and e_j .

We now outline our proof of this theorem. We first claim that $\text{dcsd}(\mathcal{V}) \subset \text{VB}_\tau(\mathcal{V})$. Suppose $(p, q) \in \text{dcsd}(\mathcal{V})$. It follows from the definition of $\text{VB}_\tau(\mathcal{V})$ that if $(p, q) \notin \text{VB}_\tau(\mathcal{V})$, the shorter arc of \mathcal{V} joining p and q has total turning angle $< \pi$. Then the polygon created by closing this arc with the chord from p to q has total turning angle $< 2\pi$. But this contradicts Fenchel's theorem (a closed curve has total curvature $\geq 2\pi$). This proves the claim, and hence that

$$\min_{\text{dcsd}(\mathcal{V})} \|p - q\| \geq \min_{\text{VB}_\tau(\mathcal{V})} \|p - q\|. \quad (5)$$

If $\min_{\text{VB}_\tau(\mathcal{V})} \|p - q\|$ is realized at an interior point (p_0, q_0) of $\text{VB}_\tau(\mathcal{V})$, then (p_0, q_0) is a local minimum of the self-distance function d on \mathcal{V} . In this case, $(p_0, q_0) \in \text{dcsd}(\mathcal{V})$ so (5) is an equality and we have completed the proof. (To see this, substitute Definition 1 into the statement of the Theorem.)

Similarly, if $\min_i 2/\kappa_i < \tau$, we have also completed the proof. So we are left with the case where neither of these holds. In this case,

$\min_{\text{VB}_\tau(\mathcal{V})}$ is realized at some (p_0, q_0) on the boundary of $\text{VB}_\tau(\mathcal{V})$. We can assume that q_0 is a vertex. Schur's theorem [11] and the fact that each $\kappa_i \geq 2/\tau$ implies that the distance between p_0 and q_0 is greater than the distance between corresponding points p_0^* and q_0^* on a planar polygonal arc \mathcal{V}^* with the same edglength and each $\kappa_i = 2/\tau$. Examining the geometry of the situation, we can then prove that \mathcal{V}^* has an inscribed circle of diameter τ , and that $\|p_0^* - q_0^*\| \geq \tau$. Since (5) then implies that $\min_{\text{dcsd}(\mathcal{V})}$ is also $\geq \tau$, this completes the proof.

2.3 The gradients of $1/\kappa_i$ and $d(e_i, e_j)$

We have now given the thickness bound in the standard form of a collection of inequality constraints on functions of the vertex positions. To implement our tightening flow, we will need to know the derivatives of these functions. We begin with the derivative of $1/\kappa_i$. For any triple of adjacent vertices a, v_i, b of \mathcal{V} , $1/\kappa_i$ is a minimum of two functions $\|b - v_i\|/2 \tan(\theta/2)$ and $\|a - v_i\|/2 \tan(\theta/2)$, where θ is the turning angle of the polygon at v_i . We need the derivatives of these functions.

Let \vec{n} denote the oriented unit normal to the plane containing a, v_i , and b . If we define one scalar constant k and three vector constants \vec{u}, \vec{v} , and \vec{w} by

$$k = \frac{\|b - v_i\|}{2 \cos \theta - 2}, \quad \vec{u} = \frac{b - v_i}{2 \tan \theta \|b - v_i\|}, \\ \vec{v} = k \frac{(a - v_i) \times \vec{n}}{\|a - v_i\|^2}, \quad \vec{w} = k \frac{\vec{n} \times (b - v_i)}{\|b - v_i\|^2},$$

then a lengthy computation reveals that

$$\nabla \frac{\|b - v_i\|}{2 \tan(\theta/2)} = (\vec{u}, -\vec{u} - \vec{v} - \vec{w}, \vec{v} + \vec{w}), \quad (6)$$

where we have written the gradient as three vectors paired with a, v_i , and b respectively. If we were thinking of $1/\kappa_i$ as a function of \mathcal{V} , instead of as a function of a, v_i , and b , these would be the nine (potentially) nonzero entries in a gradient vector of length $3V$. The other gradient is similar.

Now we turn to the gradient of $d(e_i, e_j)$, which is the min-function defined by

$$d(e_i, e_j) = \min_{\alpha, \beta \in [0,1]} \|\alpha v_{i-1} + (1 - \alpha)v_i - \beta v_{j-1} - (1 - \beta)v_j\| \\ := \min_{\alpha, \beta \in [0,1]} \|\vec{s}(\alpha, \beta, v_{i-1}, v_i, v_{j-1}, v_j)\|,$$

where the vector function \vec{s} is defined as above.

If this minimum is realized by a single pair (α_0, β_0) , then by Clarke's differentiation theorem for min-functions [13], the first partials of $d(e_i, e_j)$ with respect to the positions of v_{i-1}, v_i, v_j , and v_{j-1} are the corresponding first partials of $\|\vec{s}\|$.

In this case, a straightforward computation reveals that

$$\nabla d(e_i, e_j) = \frac{1}{\|\vec{s}\|} (0, \dots, 0, \alpha_0 \vec{s}, (1 - \alpha_0) \vec{s}, 0, \dots \\ \dots, 0, -\beta_0 \vec{s}, -(1 - \beta_0) \vec{s}, 0, \dots, 0), \quad (7)$$

where we think of $d(e_i, e_j)$ as a function of $\mathcal{V} \in \mathbb{R}^{3V}$ and write the gradient as a $3V$ -vector.

Otherwise, the segments are parallel and a family of α and β yield minimum-length chords between them. In this case, an analysis

using Clarke's theorem shows that the derivative of $d(e_i, e_j)$ is controlled by the derivatives of the distances from one vertex of e_i to e_j and from one vertex of e_j to e_i . We could have handled even this degenerate case by modeling each $d(e_i, e_j)$ as a collection of inequality constraints. But this case seems very rare in practice, so we did not add the extra constraints to our implementation.

3 TIGHTENING BY CONSTRAINED-GRADIENT

We have now expressed the constraint $\text{Thi}_p(\mathcal{V}) \geq \tau$ as a collection of inequality constraints on functions of $\mathcal{V} \in \mathbb{R}^{3V}$. We can think of the knot-tightening problem as a nonlinear optimization problem on \mathbb{R}^{3V} in the standard form

$$\min_{\mathcal{V} \in \mathbb{R}^{3V}} f(\mathcal{V}) \text{ subject to } g_i(\mathcal{V}) \geq 0. \quad (8)$$

Here we will think of the *objective function* $f(\mathcal{V})$ as $\text{Length}(\mathcal{V})$ and the g_i as the various $1/\kappa_i$ and distance functions given in Theorem 3. But we will use f for $\text{Length}(\mathcal{V})$ and g_i for the collection of constraints from here on in order to simplify our notation.

Our motivating conjecture is that the tightening flow for a smooth knot γ is the closest vector field to the first variation of the length of γ which respects the uncountably infinite set of constraints in Theorem 2 to first order. We now describe the corresponding finite-dimensional construction for the tightening flow for polygons.

3.1 Infinitesimal motions

It is standard in nonlinear optimization to refer to the inequality constraints $g_i(\mathcal{V})$ as *active* when $g_i(\mathcal{V}) = 0$ and *inactive* otherwise. We will now describe the set of velocity vectors for \mathcal{V} in $T_{\mathcal{V}}(\mathbb{R}^{3V})$ which respect (8) to first order in terms of the directional derivatives of the active constraints. We let $D_{\vec{u}}f$ denote the (forward) directional derivative of f at \mathcal{V} in the direction \vec{u} .

Definition 3. The set of infinitesimal motions $I(\mathcal{V})$ is given by

$$\vec{v} \in I(\mathcal{V}) \iff D_{\vec{v}}g_i \geq 0 \text{ for each active } g_i. \quad (9)$$

A convex set C is a convex cone if $\vec{c} \in C \iff \lambda^2 \vec{c} \in C$ for all λ . One can check that the following proposition holds.

Proposition 1. The infinitesimal motions $I(\mathcal{V})$ form a convex cone.

We now define another set of motions:

Definition 4. The convex cone of resolvable motions $R(\mathcal{V})$ of \mathcal{V} is the cone generated by the negative gradient vectors $-\nabla g_i$ of all active constraints. That is, $R(\mathcal{V})$ is the set of \vec{r} which can be expressed in the form

$$\vec{r} = -\sum \lambda_i \nabla g_i, \quad \text{with } \lambda_i > 0. \quad (10)$$

The *dual cone* of a convex cone C is the set of vectors \vec{v} with $\langle \vec{v}, \vec{c} \rangle \leq 0$ for each $\vec{c} \in C$. It is easy to check the following proposition.

Proposition 2. The convex cones $R(\mathcal{V})$ and $I(\mathcal{V})$ are dual.

3.2 A decomposition of $T_{\mathcal{V}}\mathbb{R}^{3V}$

The dual cones $R(\mathcal{V})$ and $I(\mathcal{V})$ provide a kind of orthogonal decomposition of $T_{\mathcal{V}}\mathbb{R}^{3V}$ ([40], Thm. 2.8.7):

Proposition 3. Any vector $\vec{u} \in T_{\mathcal{V}}\mathbb{R}^{3V}$ may be uniquely written

$$\vec{u} = \vec{r} + \vec{i}, \quad \text{where } \langle \vec{r}, \vec{i} \rangle = 0, \quad (11)$$

$\vec{r} \in R(\mathcal{V})$ is the closest resolvable motion to \vec{u} , and $\vec{i} \in I(\mathcal{V})$ is the closest infinitesimal motion to \vec{u} .

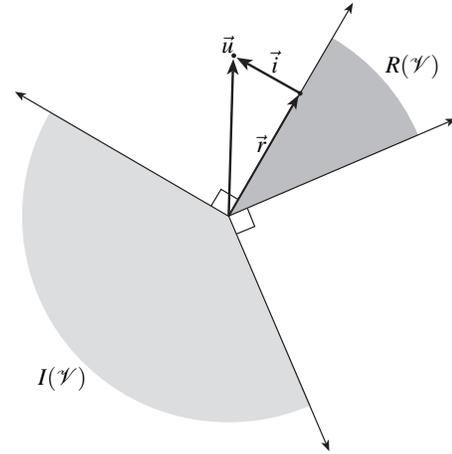


Figure 3: The geometry of Proposition 3. We see that the infinitesimal motions $I(\mathcal{V})$ and the resolvable motions $R(\mathcal{V})$ form dual convex cones. Hence, although these are not orthogonal subspaces of \mathbb{R}^{3V} , a similar decomposition property holds true.

3.3 Construction of the closest motion

We can now define our (polygonal) tightening flow:

Definition 5. The constrained gradient of a function $f: \mathbb{R}^{3V} \rightarrow \mathbb{R}$ subject to the constraints $g_i \geq 0$ at \mathcal{V} is the (unique) vector $\vec{i} \in I(\mathcal{V})$ closest to ∇f .

If $f(\mathcal{V})$ is the length of \mathcal{V} , the g_i correspond to the constraints in Theorem 3, and Proposition 3 yields $-\nabla f = \vec{r} + \vec{i}$, then the velocity of the tightening flow is \vec{i} .

The terminology "constrained gradient" is justified by proving that \vec{i} is the direction of steepest descent for f among the vectors in $I(\mathcal{V})$ [32].

We now want to compute \vec{i} and \vec{r} explicitly from a given \vec{u} . Any resolvable force is a non-negative linear combination of the negative gradients $-\nabla g_i$ of the active g_i . If we define the *rigidity matrix* A of \mathcal{V} to be the matrix whose columns are the positive gradients of the g_i , we have

$$\vec{r} \in R(\mathcal{V}) \iff A\vec{x} = -\vec{r}, \text{ where } \vec{x} \geq 0. \quad (12)$$

If we let E be the number of active constraints, then A is a $3V \times E$ matrix, and \vec{x} is a vector in \mathbb{R}^E .

It then follows from Proposition 3 that if we solve the non-negative least-squares (NNLS) problem

$$\min \|A\vec{x} + \vec{u}\|, \quad \vec{x} \geq 0 \quad (13)$$

then $\vec{r} = -A\vec{x}$. Since our constraint functions each involve only a few coordinates of $\mathcal{V} \in \mathbb{R}^{3V}$, the matrix A is extremely sparse and this is a sparse non-negative least-squares (SNNLS) problem. These problems have been well-studied in numerical linear algebra [31]. This SNNLS problem can be rewritten as a linear complementarity problem (LCP). In that form, it has been used for contact problems before, as we mentioned in the introduction.

4 SOFTWARE DESIGN

We have now defined the velocity vector of the tightening flow and so presented a mathematical formulation of our problem in a stan-

standard form—

Compute the evolution of \mathcal{V} under the (discontinuous) ODE defined by $\mathcal{V}' = \vec{i}$.

From this perspective, our simulation looks somewhat daunting to implement. For an V -vertex polygon \mathcal{V} the set VB_τ certainly contains $O(V^2)$ pairs, so we are dealing in principle with an optimization problem with $3V \simeq 3 \cdot 10^3$ variables and $O(V^2) \simeq 10^6$ constraints. In practice, the situation is much better. Geometric considerations seem to prevent too many distance constraints from becoming active at any one time, and we can avoid explicitly computing or storing inactive constraints by using a clustering algorithm (described below) to identify the active constraints.

However, using an off-the-shelf constrained optimization solver to handle this problem would require an explicit specification of all $O(V^2)$ constraints. This would likely be unworkably slow in practice. Further, other variants of gradient descent such as the reduced-gradient and projected-gradient methods have proved more popular than the constrained-gradient approach (see [14] for an evaluation of various algorithms for constrained nonlinear optimization), so we were unable to locate a modern, sparse, off-the-shelf constrained-gradient solver.

Such issues led us to design a ground-up implementation of our knot-tightening simulation using custom-built numerical methods. The result was `RidgeRunner`, a C language code for tightening knots. All of our example animations were generated by this software. We now discuss the software issues involved in generating our knot-tightening animations.

4.1 The gradient of length

To generate the negative gradient of $\text{Length}(\mathcal{V})$ as a function of the vertex positions v_i , we write $\text{Length}(\mathcal{V}) = \sum \|v_i - v_{i-1}\|$ and take

$$\begin{aligned} -\nabla \text{Length} &= -\sum \nabla \|v_i - v_{i-1}\| \\ &= \sum \left(0, \dots, 0, \frac{v_i - v_{i-1}}{\|v_i - v_{i-1}\|}, \frac{v_{i-1} - v_i}{\|v_i - v_{i-1}\|}, 0, \dots, 0 \right). \end{aligned} \quad (14)$$

Here the nonzero components of the vector inside the sum occupy positions $3i - 5, \dots, 3i$ in the $3V$ -vector $\nabla \|v_i - v_{i-1}\|$. Geometrically, $-\nabla \text{Length}$ is computed at each v_i by adding unit vectors pointing inward along each of the incident edges. At vertices with a small turning angle, these vectors sum to a small resultant since there is little opportunity to improve the length of an almost-straight polygon. At vertices with a larger turning angle, these vectors sum to a larger one, since moving the vertex in the direction of its angle bisector will produce a substantial savings in length.

4.2 Preserving equilateral edges

Our formulation of the optimization problem requires that we maintain equilateral edges during the tightening process. However, our flow will tend to shrink edges incident to vertices with large turning angles faster than edges incident to vertices of small turning angles as the knot tightens. To counteract this tendency, we add a velocity \vec{i} in the direction of the tangents to the edges proportional to $\frac{i}{n} \text{Length}(\mathcal{V}) - \text{Length}(v_i, v_0)$. This helps to move each edge length towards the desired common value. (We could have obtained an effect similar to the combination of this velocity and the length gradient by viewing each edge as a spring. But then we would have lost the freedom to tune the magnitudes of these motions separately to get the best results from our code.)

An example of the combined motion is shown in Figure 4.

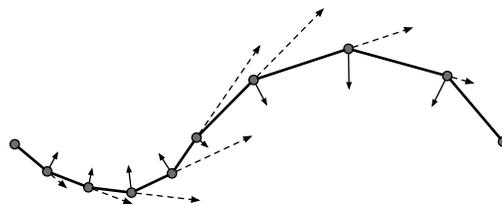


Figure 4: Our knot-tightening motion is the sum of a tangential motion \vec{i} (dotted arrows) designed to keep the polygon approximately equilateral and the negative gradient of the length of the polygon $-\nabla f$ (solid arrows).

4.3 Detecting active constraints

It is easy to check for active $1/\kappa_i$ constraints—since there are only V of them, we can check them all at each timestep. The problem of finding pairs of edges at minimum distance is harder, since there are $O(V^2)$ distances to check, but very similar to the collision detection problem in 3D. We use an octree-based method designed for the polygonal ropelength problem [1]. As with all BVH-type methods, we expect $O(V \log V)$ performance in general. (We suspect that this method is covered by the “worst-reasonable-case” $O(V^{4/3})$ time bound of [17], but have not proved it.) This method does not take advantage of the fact that the underlying polygon moves comparatively little between queries, so it would be interesting to rewrite this collision detection module using the efficiently updatable data structures of [17] or [21]. This is an area for further development of `RidgeRunner`.

4.4 Solving the NNLS problem

Having identified the active constraints, building the rigidity matrix A follows from (6) and (7). The next step is solving the NNLS problem

$$\min \|A\vec{x} + \vec{u}\|, \quad \vec{x} \geq 0. \quad (15)$$

We make use of the fast block-principal pivoting algorithm of [31], which works by solving a number of sparse unconstrained least-squares problems as intermediaries to computing the constrained solution. For these, we employ the method of normal equations and the supernodal, multifrontal sparse Cholesky code of [41].

This method for solving the NNLS problem has error proportional to the product of machine epsilon ($\sim 10^{-16}$) and the square of the condition number of the rigidity matrix, so it is only appropriate for relatively well-conditioned problems. But in our evolutions, this squared condition number¹ has an average value of $\sim 10^4$ and never exceeds $\sim 10^8$, leaving us with eight digits of accuracy even during the most delicate timesteps.

4.5 Stepping and overstepping

We have now constructed a tightening motion \vec{i} which preserves the inequality constraints of Theorem 3 to first order. We move \mathcal{V} by taking a step of some finite size h in this direction. Under ordinary circumstances, choosing the right stepsize and performing the step would be done by an ODE solver. But our situation is somewhat nonstandard. The main problem is that there is no way to predict in advance when a new constraint will become active mid-step, as occurs when two regions of a tube first make contact. So our code must be able to react to these situations as they occur and recompute stepsize accordingly. Further, even if we have computed \vec{i} accurately and no new constraints become active during a step, we

¹as estimated by LAPACK’s `rcond` function.

must still expect to violate some active constraints by an amount proportional to h^2 , since our constraints are nonlinear.

We handle both issues at once by designing our stepper to binary-search for a stepsize which introduces a fixed amount of error ϵ into the set of all constraints². Since this stepsize is typically quite small, it is acceptable to use simple Euler integration rather than a higher-order ODE method such as Runge-Kutta to make the step. We then add a procedure that reliably recovers from configurations where the constraints are violated by less than $N\epsilon$ for some large N , and switch to “error recovery mode” when this error bound is reached.

4.6 Error recovery

We use a Newton’s method procedure to recover from configurations where some constraints are violated. Recall that the columns of the rigidity matrix A are the gradients of the active constraints g_i . This means that for any given motion \vec{u} of \mathcal{V} , we have

$$A^T \vec{u} = [\langle \nabla g_i, \vec{u} \rangle] = [\nabla_{\vec{u}} g_i]. \quad (16)$$

That is, $A^T \vec{u}$ is the vector of directional derivatives of the active constraints in the direction \vec{u} . This means that we can generate a motion \vec{u} for error recovery by solving the matrix problem $A^T \vec{u} = -R$, where R is the vector of desired adjustments to the values of the g_i .

This matrix problem is always underdetermined, since (for instance) any \vec{u} generated by a rigid motion of \mathcal{V} has no effect on our constraints and so is in the kernel of A . But a fast, accurate minimum-norm solution to the problem is provided by the LSQR code of [26]. Once we have computed \vec{u} , we take the step $0.25\vec{u}$, and recompute.

The most important feature of this procedure is that the error control step adjusts all the violated constraints simultaneously using a coordinated global motion of the polygon \mathcal{V} . In this way, the tube can recover from interpenetrations even when they are trapped in the center of an almost-tight configuration of the knot.

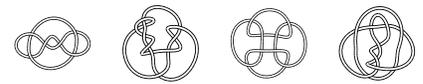
This procedure typically converges within a few steps. We do not try to eliminate all constraint violations, since we could change some active constraints to inactive ones by mistake. Instead we set an error threshold and reduce all constraint violations to within this amount. Typically, this threshold is around 10^{-5} for distance constraints, but we have made runs with a error bound several orders of magnitude lower.

5 RESULTS

We have performed `RidgeRunner` minimizations for all catalogued knots of up to ten crossings and links of up to nine crossings—384 in total. Initial configurations were obtained from the `KnotPlot` software distribution [37]. Among these, we chose three examples for our animation: the knot 9_5 and the two-component link 9_{47}^2 , which were representative of the difficulty of most knots and links in the catalogue, and the four-component link 8_3^4 , because it was among the most difficult examples in the catalogue to tighten. This last minimization is also shown in Figure 5.

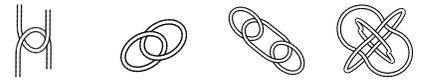
Each frame of the animation represents a constant amount of simulation time, which incorporates many steps of the underlying algorithm. The total runtime required to generate each animation is thus

²Checking all the constraints while searching (instead of only the ones active at the start of the step) is still fairly fast, since our constraint-evaluation code is carefully written, and allows us to ensure that we see any new constraints that have become active mid-step.



Vertices	95	114	136	125
Total runtime	32 m	10 m	70 m	135 m
Avg time / frame	8.25 s	1.91 s	37.46 s	22.55 s

Table 1: The runtime performance of `RidgeRunner` on a 1.8 GHz PPC G5. Total runtime is given in minutes of computation time. Average time per frame is given in seconds of computation time and represents $1/24^{\text{th}}$ of a second of simulation time.



Vertices	332	216	384	630
Polygon length	4.2636	25.1406	41.7131	29.0096
Smooth length	4.2629	8π	$12\pi + 4$	29.0030
Relative error	0.016%	0.031%	0.034%	0.022%

Table 2: The table shows preliminary results of `RidgeRunner` minimizations of some links whose minimum lengths are known theoretically or precisely conjectured. We can see that the relative error in each of these computations is quite low. In fact, it is of the same order of magnitude as the expected discretization error for these numbers of edges, so these configurations may be even closer to the minimal length polygons with the same number of edges than these numbers suggest.

controlled both by the stepsize of individual timesteps and the total amount of motion required to tighten the curve. Table 1 gives the total and per-frame average runtimes for each of the curves shown in our animation (shown in the last three columns) as well as the 6_3^2 link (shown in the first column). Although we provide only four examples here, these runtimes are representative of our experience in evolving hundreds of curves of varying complexity.

To evaluate the potential accuracy of `RidgeRunner` minimizations, we used a splining technique to generate higher-resolution versions of the final configurations of the knots and links generated in our initial animation runs, and ran them for longer periods. We now compare the final ropelength values and sets of active constraints of four `RidgeRunner`-tightened polygonal links to the corresponding smooth tight configurations. The length results are shown in Table 2 for four links: the “Gehring clasp”, created when one tube is pulled over another at right angles [7], the Hopf link of two loops of rope [8], the simple chain of three loops of rope [8], and the Borromean rings (a 3-component link) [7]. We chose these as a test set since they are among the few links whose tight ropelength has been computed analytically or precisely conjectured. As the results of the table show, the error in our computations is of the same order of magnitude as we would expect from discretizing each of these smooth links with the given number of edges. Therefore, we are unable to say anything conclusive about the differences between these polygonal ropelengths and those of the corresponding minimizing polygons in these link types other than that they might be an order of magnitude better than the relative error results given in Table 2.

We also compare the sets of active constraints for these polygonal links to the corresponding sets of active constraints for the smooth links. No $1/\kappa_i$ or curvature constraints are active for these links, so we plotted the active distance constraints as pairs of points along the curves in arclength coordinates. Figure 6 shows the result for

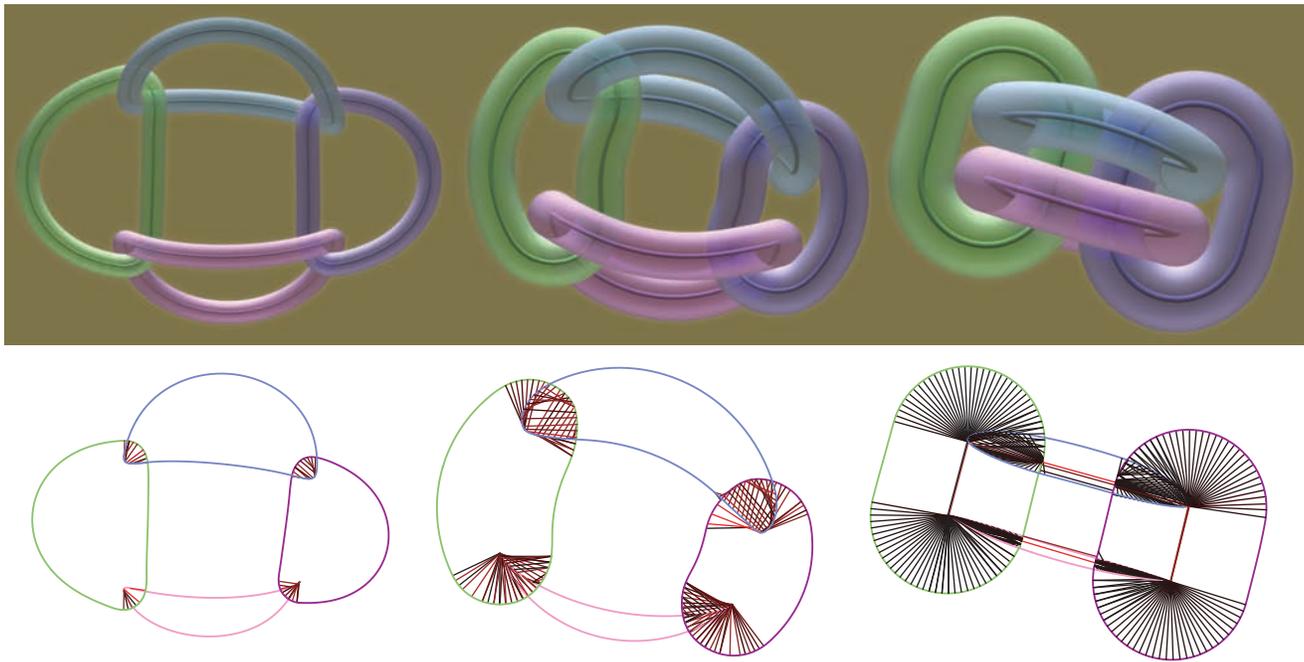


Figure 5: Three frames of the evolution of 8_3^4 and their associated sets of active struts.

the Gehring clasp, a two component link where the x and y -axes measure length along each component separately (there are no active constraints between points on the same component).

6 CONCLUSIONS AND FUTURE DIRECTIONS

This paper has described the visualization of tightening rope and the new and interesting mathematics arising from its study. The tightening animations generated by `RidgeRunner` are the first of their kind, and the final tight configurations reveal a wealth of interesting geometric detail.

Our results also suggest a number of directions for future work. For instance, we would like to explore some intriguing numerical evidence suggesting that there may be knots which are local minima for length but are very different from the true tight configurations, such as the “Gordian unknots” of Pieranski et al. [29]. A detailed examination of these cases using `RidgeRunner` might help us to prove that such (smooth) configurations really exist.

There are also some avenues for improving our simulation. While our software controls the $1/\kappa_i$ constraints at reasonable resolutions (about 10^3 vertices), these constraints are clearly more difficult to handle than the distance constraints. These discrete curvatures are numerical derivatives of the polygon data, and hence subject to well-known numerical instabilities (cf. [2]). For instance, although the condition number of the rigidity matrix was not a source of numerical instability in the evolutions discussed earlier or shown in our animation, the disparity between the magnitudes of the gradients of $1/\kappa_i$ and $d(e_i, e_j)$ can cause high condition number when curves are very finely discretized and particular types of curvature-controlled geometric structures arise. As a result, running extremely fine-grained simulations with tens or hundreds of thousands of vertices and curvature constraints will require further development of our model.

While we were originally interested in simulating a particular mathematical model, and have done so accurately, it has not escaped our notice that this is not a particularly realistic model of physical rope. It would be interesting to expand our model to include bending and

twisting stiffnesses and a frictional term. This would again require substantial development, but our existing code should be a robust foundation for handling contact in the resulting simulation.

REFERENCES

- [1] Ted Ashton and Jason Cantarella. A fast octree-based algorithm for computing ropelength. arXiv:math.DG/0409416, 2004.
- [2] Kendall E. Atkinson. *An Introduction to Numerical Analysis*. Wiley, 1989. Section 5.7 covers numerical differentiation.
- [3] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34. ACM Press, 1994.
- [4] Joel Brown, Jean-Claude Latombe, and Kevin Montgomery. Real-time knot-tying simulation. *The Visual Computer*, 20:165–179, 2004.
- [5] Gregory Buck and Jeremy Orloff. A simple energy function for knots. *Topology Appl.*, 61(3):205–214, 1995.
- [6] R. V. Buniy and T. W. Kephart. Glueballs and the universal energy spectrum of tight knots and links. arXiv:hep-ph/0408027, 2004.
- [7] Jason Cantarella, Joseph H.G. Fu, Robert B. Kusner, John M. Sullivan, and Nancy C. Wrinkle. Criticality for the Gehring link problem. arXiv:math.DG/0402212, 2004.
- [8] Jason Cantarella, Robert B. Kusner, and John M. Sullivan. On the minimum ropelength of knots and links. *Invent. Math.*, 150(2):257–286, 2002.
- [9] Jason Cantarella and Michael Piatek. TSNLS: A solver for large sparse least-squares problems with non-negative variables. arXiv:cs.MS/0408029.
- [10] M. Carlen, B. Laurie, J.H. Maddocks, and J. Smutny. *Biarcs, Global Radius of Curvature, and the Computation of Ideal Knot Shapes*. World Scientific, 2005.
- [11] S. S. Chern. Curves and surfaces in Euclidean space. In *Studies in Global Geometry and Analysis*, pages 16–56. Math. Assoc. Amer. (distributed by Prentice-Hall, Englewood Cliffs, N.J.), 1967.
- [12] Kai-Seng Chou and Xi-Ping Zhu. *The Curve Shortening Problem*. CRC Press, 2001.
- [13] Frank H. Clarke. Generalized gradients and applications. *Trans. Amer. Math. Soc.*, 205:247–262, 1975.

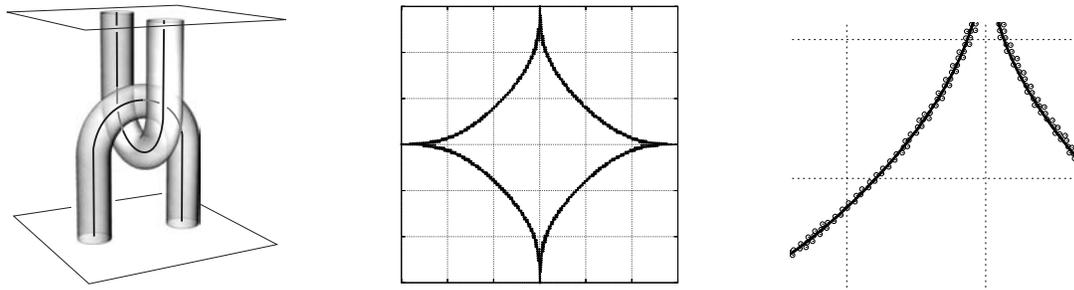


Figure 6: The left-hand picture shows a (loose) configuration of the “Gehring clasp”—a simple two-strand tangle which serves as an interesting model for the interaction between two ropes passing over each other at right angles. A ropelength-critical configuration of this tangle has been derived and studied extensively [7]. Since this derivation included a set of active constraints, it is natural to compare RidgeRunner’s results to this theoretical picture. This comparison is shown in the two plots center and right, which plot the positions of struts in arclength coordinates with the lower left corner of the plot located where each curve first begins to turn. The enlarged plot (right) shows the agreement between theoretical and computational results. To our knowledge, this is the first published comparison between theoretical and numerical sets of active constraints.

- [14] R. Fletcher. *Practical methods of optimization*. Wiley-Interscience [John Wiley & Sons], New York, second edition, 2001.
- [15] O. Gonzalez and R. de la Llave. Existence of ideal knots. *J. Knot Theory Ramifications*, 12(1):123–133, 2003.
- [16] O. Gonzalez, J. H. Maddocks, F. Schuricht, and H. von der Mosel. Global curvature and self-contact of nonlinearly elastic curves and rods. *Calc. Var. Partial Differential Equations*, 14(1):29–68, 2002.
- [17] Leonidas Guibas, An Nguyen, Daniel Russel, and Li Zhang. Collision detection for deforming necklaces. In *SCG '02: Proceedings of the eighteenth annual symposium on Computational geometry*, pages 33–42. ACM Press, 2002.
- [18] Otto Krötenheerdt and Sigrid Veit. Zur Theorie massiver Knoten. *Wiss. Beitr. Martin-Luther-Univ. Halle-Wittenberg Reihe M Math.*, 7:61–74, 1976.
- [19] Ben Laurie. Annealing ideal knots and links: methods and pitfalls. In *Ideal knots*, volume 19 of *Ser. Knots Everything*, pages 42–51. World Sci. Publishing, River Edge, NJ, 1998.
- [20] R. A. Litherland, J. Simon, O. Durumeric, and E. Rawdon. Thickness of knots. *Topology Appl.*, 91(3):233–244, 1999.
- [21] I. Lotan, F. Schwarzer, D. Halperin, and J.-C. Latombe. Efficient maintenance and self-collision testing for kinematic chains. In *Proc. 18th ACM Symposium on Computational Geometry*, pages 43–52, 2002.
- [22] P. Lötstedt. Coulomb friction in two-dimensional rigid body systems. *Zeitschrift für Angewandte Mathematik un Mechanik*, 61:605–615, 1981.
- [23] Victor J. Milenkovic and Harald Schmidl. Optimization-based animation. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 37–46, New York, NY, USA, 2001. ACM Press.
- [24] J. J. Moreau. Quadratic programming in mechanics: Dynamics of one-sided constraints. *SIAM Journal on Control*, 4(1):153–158, 1966.
- [25] Alexander Nabutovsky. Non-recursive functions, knots “with thick ropes”, and self-clenching “thick” hyperspheres. *Comm. Pure Appl. Math.*, 48(4):381–428, 1995.
- [26] Christopher C. Paige and Michael A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, 1982.
- [27] Mark Pauly, Dinesh K. Pai, and Leonidas J. Guibas. Quasi-rigid objects in contact. In *SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 109–119, New York, NY, USA, 2004. ACM Press.
- [28] J. Phillips, A. Ladd, and L. Kavradi. Simulated knot tying, 2002.
- [29] P. Pieranski, S. Przybyl, and A. Stasiak. Gordian unknots. arXiv:physics/0103080, 2004.
- [30] Piotr Pierański. In search of ideal knots. In *Ideal knots*, volume 19 of *Ser. Knots Everything*, pages 20–41. World Sci. Publishing, River Edge, NJ, 1998.
- [31] Luís F. Portugal, Joaquim J. Júdice, and Luís N. Vicente. A comparison of block pivoting and interior-point algorithms for linear least squares problems with nonnegative variables. *Math. Comp.*, 63(208):625–643, 1994.
- [32] M. J. D. Powell. Introduction to constrained optimization. In *Numerical methods for constrained optimization (Proc. Sympos., National Physical Lab., Teddington, 1974)*, pages 1–28. Academic Press, London, 1974.
- [33] Eric Rawdon. *The Thickness of Polygonal Knots*. PhD thesis, The University of Iowa, 1997.
- [34] Eric J. Rawdon. Approximating the thickness of a knot. In *Ideal knots*, volume 19 of *Ser. Knots Everything*, pages 143–150. World Sci. Publishing, River Edge, NJ, 1998.
- [35] Eric J. Rawdon. Approximating smooth thickness. *J. Knot Theory Ramifications*, 9(1):113–145, 2000.
- [36] Eric J. Rawdon. Can computers discover ideal knots? *Experiment. Math.*, 12(3):287–302, 2003.
- [37] Robert G. Scharein. KnotPlot. <http://www.knotplot.com>, 2005. Program for drawing, visualizing, manipulating, and energy minimizing knots.
- [38] Friedemann Schuricht and Heiko von der Mosel. Characterization of ideal knots. *Calc. Var. Partial Differential Equations*, 19(3):281–305, 2004.
- [39] A. Stasiak, V. Katritch, J. Bednar, D. Michoud, and J. Dubochet. Electrophoretic mobility of DNA knots. *Nature*, 384:122, November 1996.
- [40] J. Stoer and Ch. Witzgall. *Convexity and optimization in finite dimensions*, volume I. Springer-Verlag, 1970.
- [41] Sivan Toledo, Vladimir Rotkin, and Doron Chen. TAUCS: A library of sparse linear solvers, 2003. <http://www.tau.ac.il/~stoledo/taucs/>.