Numerical Knot Untangling

Max Brodeur

Department of Mathematics & Statistics, McGill University.

May 5, 2023

Abstract

The goal of this paper is to provide a detailed description of the steps required to implement a Möbius energy numerical approximation algorithm for untangling knots, along with a brief introduction to the basics of knot theory. The steps include defining a knot energy, choosing the correct independent variables for the gradient flow equation, discretizing the components of the approximation, and computing the gradient. Additionally, a dynamic time-step optimization technique is provided by bounding the magnitude of the spatial-step, resulting in a significant increase in the simulation's performance together with an improvement in the method's stability.

1 Introduction

Knot theory is the subfield of topology that focuses on the study of, you guessed it, knots. This research topic is of great interest through several areas of real-world applications ranging from the study of polymer shapes [1] in molecular biology – notably DNA knots [2] all the way to data representations such as high dimensional path planning in robotics [3].

At the end of the day, as topology often goes, it all comes down to classification. The main bulk of the research in knot theory consists of finding ways to distinguish knots, defining meaningful characterizing properties and, when given a knot, identifying its class. A wonderful book [4] by Rolfsen from 1976 makes for a great introduction to the world of knot classification and is accompanied by an extensive amount of hand drawn knots. Scharein's thesis [5] also provides a great and more recent reference on the topic, and presents his pioneering *KnotPlot* software.

Although there are many fascinating facets to topological knot theory, this paper focuses on the more computational side of things. Performant numerical approximation methods and reproducible implementations provide useful tools in the study and analysis of such abstract problems.

To achieve the task of untangling a knot, the general idea is to define a knot energy $\mathcal{E}$ that provides a metric for a curve's "knottedness". This energy must

be defined in such a way that it blows up in the case of self-intersection. The first to come up with an energy satisfying this property was O'Hara in 1988 [6] when he proposed what was later called the Möbius energy. His work was motivated by previous research based on the electrical force exerted between two particles of opposite charge [1]. O'Hara's discovery was groundbreaking in the field of knot theory, since it finally provided an "infinite barrier to self intersections" [7]. Freedman et al. [8] subsequently proved the energy's invariance to Möbius transformations, hence the given name.

To the best of my knowledge, the most cutting edge knot untangling implementation, and the main influence of this paper, was done by Yu et al. in 2021 [9]. Their method incorporates the tangent point knot energy defined by Buck and Orloff from 1995 [10]. However, for no particular reason other than curiosity, this paper concentrates on the implementation of the Möbius energy instead.

The goal of this piece of research is to provide in details the step-by-step requirements for the implementation of a knot untangling method using the Möbius energy, with a brief introduction to the basics of knot theory along the way. An important emphasis is set on the intuition behind the approximated gradient flow equation given in [9]. The entire derivation for the gradient approximation is provided in the Appendix. Moreover, a method is given for dynamically setting the approximation time step, providing an increase in performance and stability. This is achieved by setting a bound over the magnitude of the spatial-step, while ensuring safe and large time steps when appropriate.

2 Knots

In topology, knots don't have loose ends; they are closed loops. They are defined as embeddings of the circle S^1 into three-dimensional space $\mathbb{R}^3$. Two knots are considered *equivalent* if they can be smoothly deformed into one another [5]. The applied deformation is restricted in such a way that prevents the cutting, stitching or intersecting of the curve. Formally, this is called a *homeomorphism*, and two knots are said to be *homeomorphic* when such a deformation exists between them. An embedding is pronounced *unknotted* or *trivial* if it can be continuously deformed into a circle, otherwise it is *knotted*. Sometimes knots can consist of more than one embedding of the circle. These, embeddings can either be linked or disjoint, and each embedding is referred to as the one of the knot's *components*. For example the logo for the Olympics is a knot containing six linked components, although, they are often displayed with intersections which is rarely legal in knot theory.

The *unknot* decision problem consists of determining if a given knot can be *unknotted*. This problem has been proven to be in the NP complexity class [11] and motivates the study of knot untangling methods, hopefully to help gain some useful insight.

Below are descriptions of some knots that will be repeatedly studied throughout this work: the torus knot, the overhand knot, and the reef knot.

(a) Overhand knot (non-closed)

(b) Reef knot (non-closed)

Torus knot. A torus knot is generated by specifying the amount of revolutions around the origin axis and around the circle axis. The torus knot, denoted $T(p, q)$, is parametrically defined as follows

$$
\begin{aligned}
x &= [r_2 \cdot \cos(p \cdot \phi) + r_1] \cdot \cos(p \cdot \phi) \\
y &= [r_2 \cdot \cos(p \cdot \phi) + r_1] \cdot \sin(p \cdot \phi) \\
z &= -r_2 \sin(q \cdot \phi)
\end{aligned}
$$

where r_1 is the radius of the revolution around the origin, r_2 is the radius of the revolution around the circle axis, p determines the number of revolutions around the origin, q determines the number of revolutions around the circle axis, and $0 < \phi \leq 2\pi$.

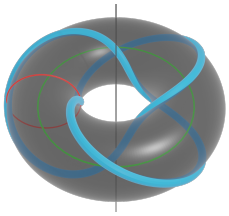
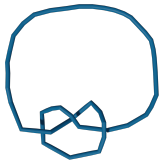


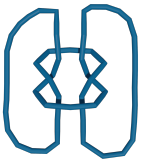
Figure 2: The *Trefoil* knot

Overhand knot. The overhand knot is a special one, since it is secretly a torus knot. To study its properties, we connect its end-points. Once closed, it suddenly becomes equivalent to the *trefoil* knot $T(2, 3)$. The trefoil is the simplest occurence of a *knotted* embedding, *i.e.* it can't be unknotted.

Reef knot. The reef knot is a simple sailing knot. Unlike the overhand knot, there are multiple ways to close it due to its four end-points.



(a) Overhand knot (closed)



(b) Reef knot (closed)

(c) Reef knot (closed)

3 Möbius energy

The goal of this paper is to provide a method for untangling knots. This method should continuously deform a knot up to its simplest form. In the trivial case the result would be a circle, and for the closed overhand knot a trefoil. In order to achieve this, the amount of *knottedness* is quantified by means of a *knot energy* $\mathcal{E}$. This energy must be steady for circles and blow up to infinity in the event of self-intersections.

Let $\gamma(s)$ be our closed curve embedding, parametrized by its arc-length s. The embedding is a function $\gamma : M \to \mathbb{R}^3$ where $M \subset \mathbb{R}^+$ is the arc-length parameter space. One of the most prominent examples of knot energies, and the one implemented in this paper, is O'Hara's Möbius energy [6]. The energy was influenced by Coulomb's inverse-square law for quantifying the electrical force between two particles [1]. As defined in [9], the non-regularized **Möbius energy** is given by

$$\mathcal{E}(\gamma) = \iint_{M \times M} \frac{1}{\|\gamma(u) - \gamma(v)\|^2} - \frac{1}{d(u,v)^2} \, dv \, du \tag{1}$$

where $d(u,v)$ is a function $d : M \times M \to \mathbb{R}^+$ that gives the shortest distance between $\gamma(u)$ and $\gamma(v)$ *along* the curve and $\| \cdot \|$ is the standard Euclidean distance in $\mathbb{R}^3$. Intuitively, the first term in the integrand echoes Coulomb's law for opposite sign particles: the force exerted between two particles scales inversely to their squared distance. As for the curve, the closer two points are, the greater the contribution from this electrical force term. Notice that it blows up to infinity as the distance approaches null. The second term in the integrand is motivated by the first. To represent knottedness, a point pair should not excert attraction when they are close along the curve itself. These two cases are well visualized on the hourglass geometry, where the points in the middle are close to each other in space but far apart along the curve.

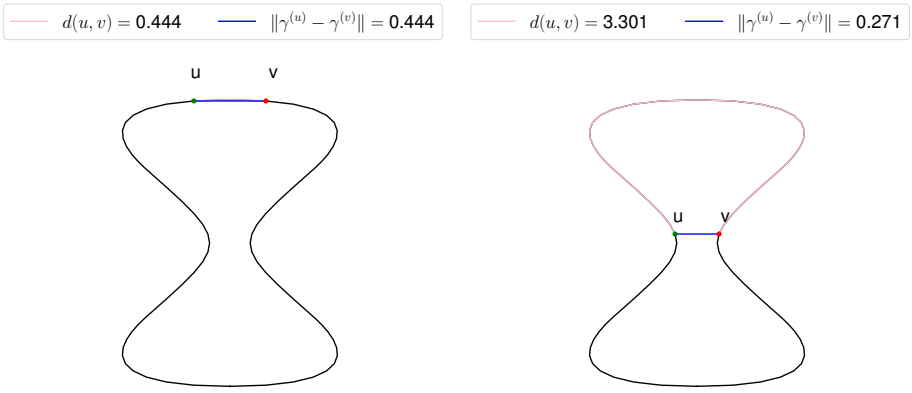


Figure 4: Curve distance (pink) vs. Euclidean distance (blue) on the hourglass geometry.

3.1 Minimizing the Möbius Energy

In order to untangle a knot, we aim to minimize equation 1. As described in [9], the behavior is modeled through a *gradient flow* equation:

$$\gamma_t = -\nabla\mathcal{E}(\gamma) \tag{2}$$

Where γ represents the three-dimensional curve as a function of time and arc-length, and $\nabla\mathcal{E}$ is the gradient of the energy function. Intuitively, as time increases, the goal is for the curve's energy to decrease. Since the gradient yields a vector in the spatial direction of the energy's steepest increase, the gradient flow equation seems to model the desired behavior.

Now, the tricky part is in finding what the gradient of our energy really means. It is unclear with respect to which independent variables the gradient in equation 2 is evaluated. Three possible candidates are considered. The simplest choice of spatial variable is the arc-length parameter s, turning equation 2 into

$$\gamma_t = -\mathcal{E}_s$$

Indeed, $\mathcal{E}$ is a function of γ which in turn is a function of s. However, the Möbius energy doesn't even depend on a choice of arc-length because it computes a double integration over all arc-length pairs. The fact that our energy is *global* implies that $\frac{\partial}{\partial s}\mathcal{E} = 0$. Geometrically, the direction of a step in arc-length is *always* along the curve and has no effect on the energy. This shows that we need to consider variations that are *not* along the curve to minimize the Möbius energy. A logical next approach could be to view the parametrized curve in the following form

$$\gamma(s) = \begin{bmatrix} x(s) \\ y(s) \\ z(s) \end{bmatrix}$$

where x, y, z are functions mapping the arc-length to their corresponding coordinates in space. It does seems reasonable to consider x, y, z as the spatial variables since a variation of either of them isn't necessarily along the curve. The gradient flow from equation 2 would then be written as

$$\begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} = - \begin{bmatrix} \mathcal{E}_x \\ \mathcal{E}_y \\ \mathcal{E}_z \end{bmatrix}$$

To see the problem with this equation, consider the translation of the curve in space. The energy of a curve centered at the origin doesn't differ from that of same curve centered somewhere else. The Möbius energy is thus invariant under variations in x, y, z. Sadly, as with the arc-length, $\mathcal{E}_x = \mathcal{E}_y = \mathcal{E}_z = 0$.

In fact, the energy only varies with respect to the curve's overall shape. This is good news, since it should not change with respect to anything else. The only variable describing the entire shape of the curve, is the curve γ itself. Therefore, the equation that correctly models our desired behavior should not be equation 2, which is confusing, but instead the following differential equation

$$\gamma_t = -\mathcal{E}_\gamma \tag{3}$$

4 Numerics

4.1 Discretization

To simulate knot untangling, equation 3 must be discretized in time and in space. The curve γ is chosen to undergo the space discretization, and is numerically represented as an array of N vertices denoted $\hat{\gamma}$. This is particularly useful for the implementation, since three-dimensional objects are usually represented the same way in `.obj` files.

The Möbius energy from equation 1, being a double integral over γ, must now iterate over a discrete set $\hat{\gamma}$ as opposed to a continuous range of values, and is now expressed as the following double summation

$$\hat{\mathcal{E}}(\hat{\gamma}) = \sum_{\mathrm{u}}^{N} \sum_{\mathrm{v}}^{N} \frac{1}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}\|^2} - \frac{1}{\hat{d}(\mathrm{u},\mathrm{v})^2} \tag{4}$$

Where $\hat{\gamma}^{(i)} \in \mathbb{R}^3$ is the i^{th} vertex on the discretized curve and $\hat{d}(\mathrm{u},\mathrm{v})$ is the discretized curve distance function. The definition for $\hat{d}$ is provided in the Appendix.

It is now officially possible to describe our equation as the gradient flow from equation 2, since the discretization of our curve $\hat{\gamma}$ creates N independent spatial variables that satisfy the behavioral requirements.

$$\hat{\gamma}_t = -\nabla \hat{\mathcal{E}}$$

The following system of differential equations is obtained

$$\begin{bmatrix} \hat{\gamma}_t^{(1)} \\ \hat{\gamma}_t^{(2)} \\ \vdots \\ \hat{\gamma}_t^{(N)} \end{bmatrix} = - \begin{bmatrix} \hat{\mathcal{E}}_{\hat{\gamma}^{(1)}} \\ \hat{\mathcal{E}}_{\hat{\gamma}^{(2)}} \\ \vdots \\ \hat{\mathcal{E}}_{\hat{\gamma}^{(N)}} \end{bmatrix}$$

Where $\hat{\mathcal{E}}_{\hat{\gamma}^{(i)}}$ is the partial derivative of the discretized Möbius energy with respect to the i^{th} vertex. The analytic solution for the gradient term $\nabla \hat{\mathcal{E}}$ is shown in detail in the Appendix.

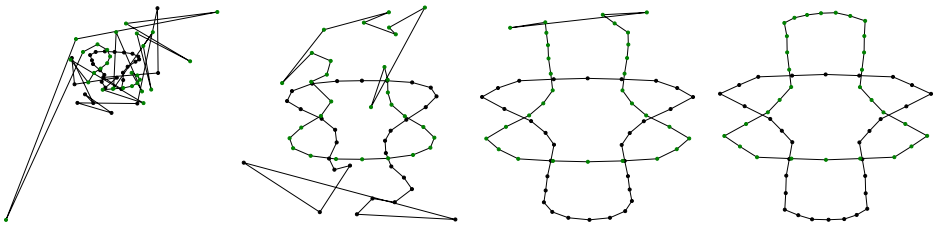
A simple forward Euler approximation is used to discretize in time

$$\hat{\gamma}_{n+1} = \hat{\gamma}_n - dt \cdot \nabla \hat{\mathcal{E}} \tag{5}$$

where dt is a small enough time step. An optimization for defining a stable time step value is proposed in the next section.

4.2 Time step optimization

The step in space taken in equation 5 is determined by the magnitude of the energy's gradient and the time step dt. For a fixed dt, when a knot is tightly tangled with high magnitude energy gradients, there is no guarantee as to where the approximation step brings the tangled segments. This is because when two curve segments get arbitrarily close, the magnitude of the Möbius energy gradient becomes arbitrarily large. Therefore the approximation's spatial step size is unbounded. Moreover, in this situation there is a possibility for self intersection with the segments opposing the tangled ones. Recall that the Möbius energy was defined to blow up at self-intersections, thus breaking the approximation. As a result, the time step required for a stable untangling is often extremely small as to avoid the possibility of self intersections. In fact, a time step of $dt = 1 \times 10^{-7}$ is required to prevent the closed reefknot from exploding when iterating $n = 10$ times.

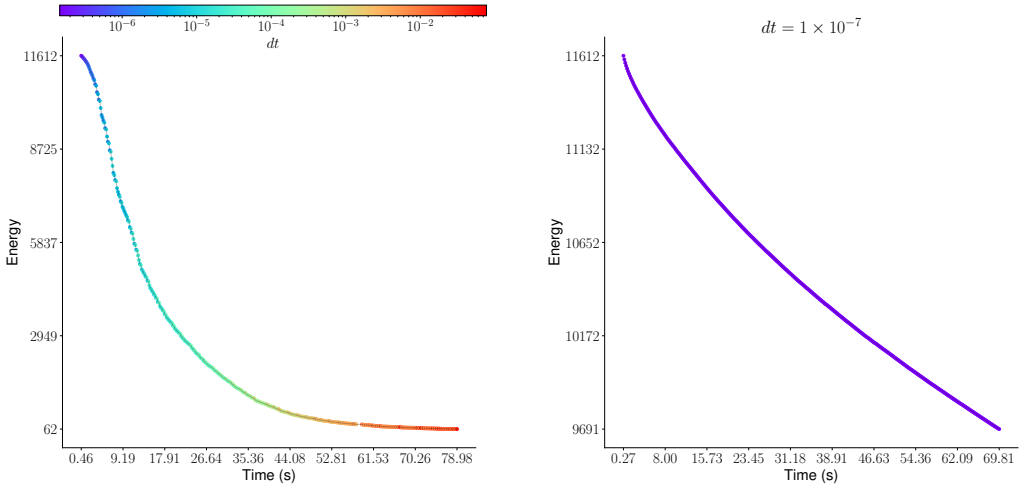


(a) $dt = 1 \times 10^{-4}$ (b) $dt = 1 \times 10^{-5}$ (c) $dt = 1 \times 10^{-6}$ (d) $dt = 1 \times 10^{-7}$

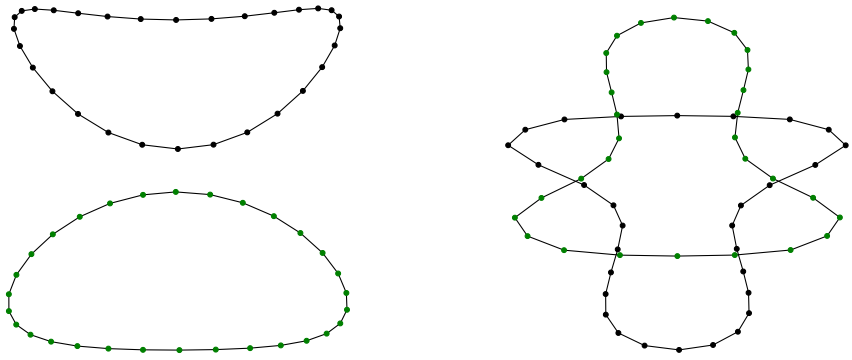
To ensure the stability of the method, the time step value dt is dynamically set to a value such that it is impossible for a spatial step in the opposing direction of the gradient to induce self intersection. The time step size is calculated like so

$$dt = \frac{\mathrm{d}_{\min} \times \text{tolerance}}{\max(\nabla \hat{\mathcal{E}})}$$

Where $\mathrm{d}_{\min}$ is the minimum distance between any two vertices on the curve. The tolerance is a user-defined parameter which indicates up to what proportion of the minimum distance the approximation step can take. A higher tolerance will lead to a quicker untangling, but cause unwanted oscillations between vertices over time. Whereas a lower tolerance yields a longer but smoother untangling experience. Figure 6 demonstrates the overwhelming performance increase this varying time step optimization provides. Indeed, in just over a minute, the varying time step method successfully untangles the closed reef knot whereas the constant time step method makes almost no progress.



(a) Energy vs. Time (in seconds).



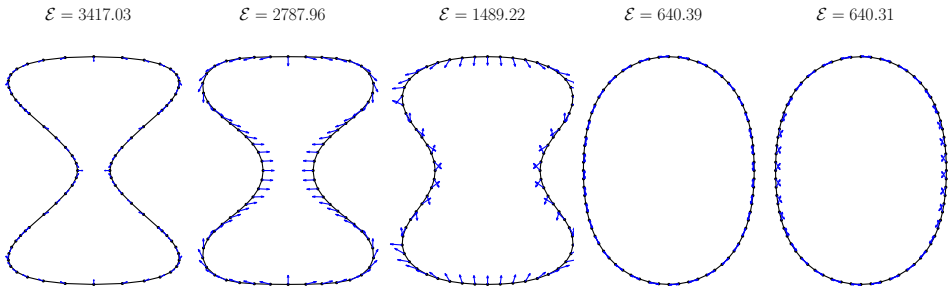
(b) Resulting shape after 400 iterations.

Figure 6: Varying time step (left) vs. constant time step (right)

5 Results

The proposed knot untangling method was implemented in python 3.10 and is available on GitHub through this link. This paper's experiments were run on a personal computer equipped with the Apple Silicon M1 Pro chip along with 32GB of memory. The energy gradient computation is parallelized over all vertex pairs and follows closely the results shown in the Appendix.

Experiments are shown with three different curves: the 2D hourglass geometry consisting of 72 vertices as a toy example to visualize the gradient directions, the closed reef knot with 60 vertices for a two component untangling demonstration, and the closed overhand knot with 69 vertices to show its conversion to the trefoil knot.

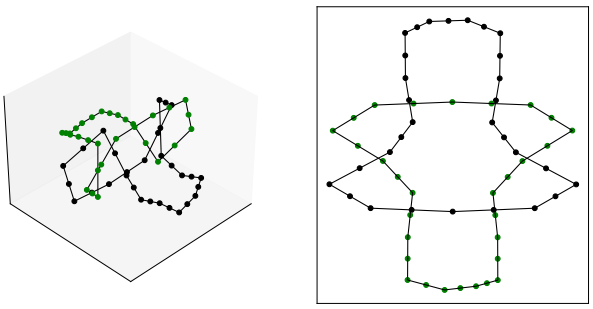


(a) $n = 0$ (b) $n = 250$ (c) $n = 500$ (d) $n = 1599$ (e) $n = 1600$

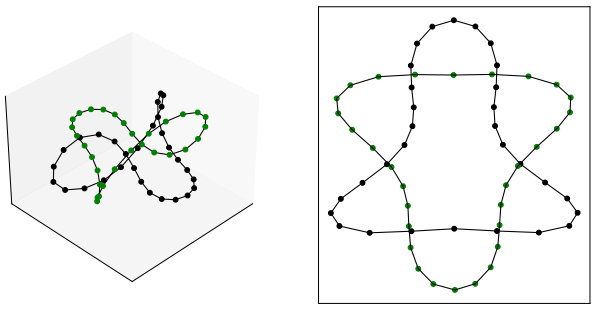
Figure 7: Möbius gradients of the hourglass geometry over 1600 iterations with tolerance=0.01

The gradient arrows of the hourglass geometry confirm the correctness the gradient computation described in the Appendix. They indicate a higher energy for closer segments, and steps in their opposite direction effectively "untangle" the curve. The reason behind the shorter arrow length at $n = 0$ stems from the geometry's vertex distribution. It is common to have higher vertex densities at higher curvature segments, causing a non-uniform distribution of the vertices along the curve. Each arrow is of length $\mathrm{d_{min}}$ – the minimum distance between any two vertex pairs along the curve computed for the time step optimization. Initially, that distance value is extremely small because of the higher density segments where neighboring vertices get extremely close. This often occurring initial state causes the method to express a *smoothing* behavior, which is explained in detail later in this section.

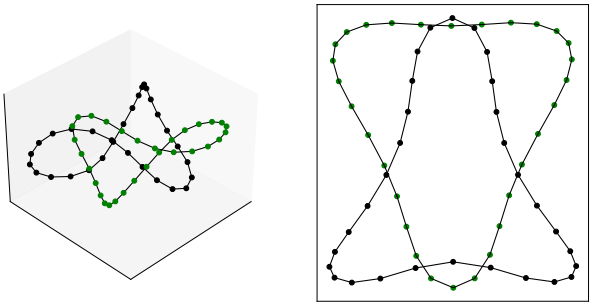
After about a thousand iterations, the hourglass shows the oscillating behavior mentioned at the end of the stability discussion. Figures 7d and 7e are at one iteration apart and the gradient arrows around the oval directly shift from pointing from one neighbor to the other. This oscillation effect can be attenuated by reducing the tolerance parameter since the iteration step would cause less disruption along the curve.



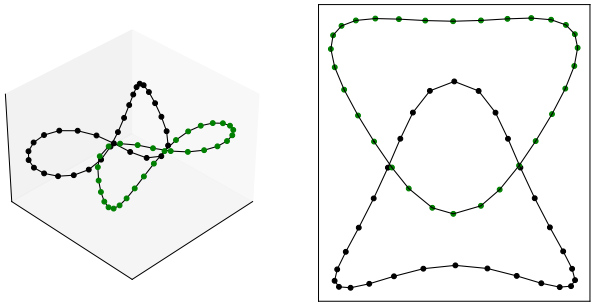
(a) $n = 0$



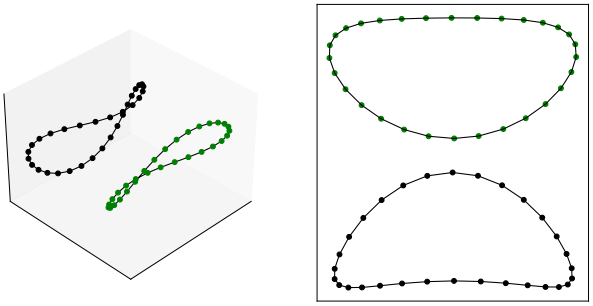
(b) $n = 100$



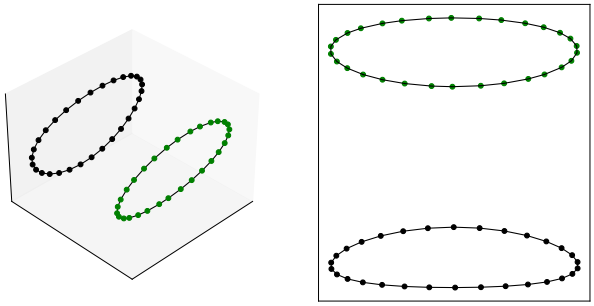
(c) $n = 200$



(d) $n = 300$

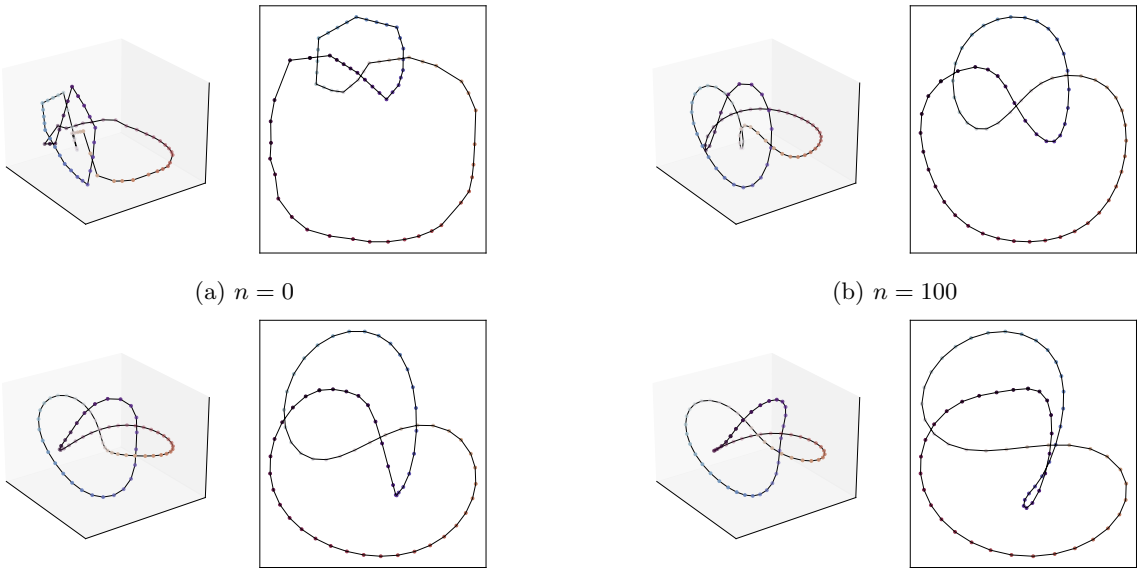


(e) $n = 400$



(f) $n = 600$

Figure 8: Closed reef knot untangling over 600 iterations with tolerance = 0.1.



(a) $n = 0$ (b) $n = 100$

(c) $n = 400$ (d) $n = 800$

Figure 9: Knot untangling from the closed overhand knot to the trefoil in 800 iterations with a tolerance of 0.1

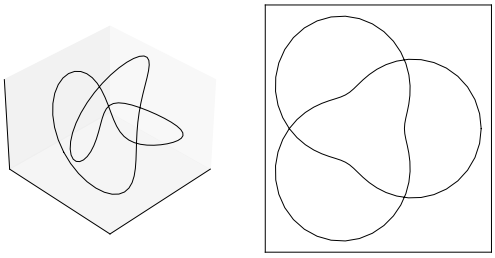


Figure 10: Generated trefoil knot $T(2, 3)$ for reference.

Aside from untangling knots, an interesting effect of the method is the initial smoothing of the curve. Throughout the experiments the first few iterations serve to *smooth out* the curve's vertices. For example, compare Figure 9a with 9b. Roughly, this is because neighbors tend to pull on each other with equal forces. Indeed, the gradient of the curve distance term (denoted B_i in the Appendix) yields an attractive force between a vertex and its neighbor in the approximation. Since the simulation is stepping in a spatial direction that decreases the energy, it seeks lower distances between vertices along the curve as to reach a higher value for $d(u, v)$. However, a shorter distance between neighboring vertices yields a higher distance with their neighbors in the opposing direction.

This is why, in the highly likely case of an initial non-uniform distribution of the vertices along the curve, the method seeks to uniformly distribute the vertices. Once the electrical force energy term overcomes the curve distance term, the untangling begins. It is difficult to demonstrate the untangling quality difference between tolerance values via still images. The oscillations created by high tolerance values propagate through out the curve, and in some cases create severe distortions. For pretty looking untangling animations, lower tolerance values such as 0.05 are often preferred, although requiring more iteration steps. Video demonstrations of experiments with other knots are included in the extra material.

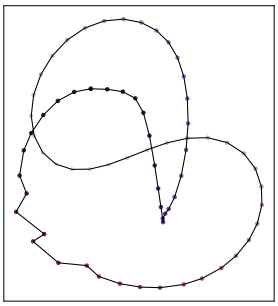


Figure 11: Distortions for tolerance = 0.99.

6 Conclusion

In the end, the Möbius energy is an intuitive function that induces the knot untangling behavior when minimized. Although it requires very small time step values to provide stability, the implementation's performance is significantly increased by applying a dynamic time step optimization scheme.

A useful application of the presented method could be to help smooth out poor designs while preventing self-intersections with other segments – it definitely fixed my poor Blender designs. Therefore, some variation of this method could definitely be used as a correction tool to help speed up the 3D modelling process.

The implementation still lacks a few important features. First of all, the method currently relies on a fixed number of iterations to terminate. It should instead rely on a metric determining an appropriate time to stop the simulation. An idea is to track the progress made in the untangling such as in [9], and stop the iterations once the progress dives under a predetermined threshold. Further, the method is short on testing, and should undergo rigorous stress tests. To date, there have been no experiments with knots containing more than 100 vertices, and the method's limits and drawbacks are unclear. Finally, to help with the oscillations at higher tolerance values, an idea could be to add a damping term to the approximated system of equations.

In the future, I believe it would be beneficial to implement the same structured implementation with several knot energies at once, such as the tangent point energy [10], with the goal of gaining further insight into different energy strengths and weaknesses.

Appendix A

A.1 Discretized curve distance function

We must adapt the curve distance function $d(u, v)$ where u, v are arc-lengths to the discretized curve $\hat{\gamma}$. The discretized curve distance function $\hat{d}(\mathrm{u}, \mathrm{v})$ computes the length of the shortest path P along the curve between vertices $\hat{\gamma}^{(\mathrm{u})}$ and $\hat{\gamma}^{(\mathrm{v})}$ where u, v are now indices. It is given by the following equation

$$\hat{d}(\mathrm{u}, \mathrm{v}) = \sum_{\mathrm{ab} \in P} \|\hat{\gamma}^{(\mathrm{a})} - \hat{\gamma}^{(\mathrm{b})}\|$$

Where $\|\cdot\|$ is the usual Euclidean distance norm. The pink path in figure 4 shows visualizations for $\hat{d}(\mathrm{u}, \mathrm{v})$.

A.2 Gradient of the discretized Möbius energy

We solve for the gradient of the discretized Möbius energy from equation 4

$$\nabla \hat{\mathcal{E}} = \begin{bmatrix} \hat{\mathcal{E}}_{\hat{\gamma}^{(1)}} \\ \hat{\mathcal{E}}_{\hat{\gamma}^{(2)}} \\ \vdots \\ \hat{\mathcal{E}}_{\hat{\gamma}^{(N)}} \end{bmatrix}$$

The partial differentiation by the i^{th} vertex $\mathcal{E}_{\hat{\gamma}^{(i)}}$ can be analytically solved by hand. Starting with

$$\begin{aligned} \hat{\mathcal{E}}_{\hat{\gamma}^{(i)}} &= \frac{\partial}{\partial \hat{\gamma}^{(i)}} \sum_{\mathrm{u}}^{N} \sum_{\mathrm{v}}^{N} \frac{1}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}\|^2} - \frac{1}{\hat{d}(\mathrm{u}, \mathrm{v})^2} \\ &= \sum_{\mathrm{u}}^{N} \sum_{\mathrm{v}}^{N} \underbrace{\frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}\|^2}}_{A_i(\mathrm{u},\mathrm{v})} - \underbrace{\frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\hat{d}(\mathrm{u}, \mathrm{v})^2}}_{B_i(\mathrm{u},\mathrm{v})} \end{aligned}$$

Where $\hat{d}(\mathrm{u}, \mathrm{v})$ is defined in Appendix A.1 as the discretized curve distance function. $A_i(\mathrm{u}, \mathrm{v})$ is the partial derivative of the electrical force term between vertices $\hat{\gamma}^{(\mathrm{u})}, \hat{\gamma}^{(\mathrm{v})}$ with respect to vertex $\hat{\gamma}^{(i)}$, and $B_i(\mathrm{u}, \mathrm{v})$ is the partial derivative of the curve distance term involving the same vertices.

Electrical force term partial derivative A_i. Evaluating the partial of the A_i term yields

$$A_i(\mathrm{u}, \mathrm{v}) = \frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}\|^2} = \begin{cases} -2 \frac{\hat{\gamma}^{(i)} - \hat{\gamma}^{(\mathrm{v})}}{\|\hat{\gamma}^{(i)} - \hat{\gamma}^{(\mathrm{v})}\|^4} & \text{if } i = u \\ 2 \frac{\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(i)}}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(i)}\|^4} & \text{if } i = v \\ 0 & \text{Otherwise.} \end{cases}$$

To show this, first consider the vector $\mathbf{v} \in \mathbb{R}^3$. The chain rule is applied to the result later to generalize for A_i. The derivative for the inverse of the squared Euclidean norm of $\mathbf{v}$ yields

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}} \frac{1}{\|\mathbf{v}\|^2} = -2\frac{\mathbf{v}}{\|\mathbf{v}\|^4}$$

Proof. Substituting the Euclidean norm definition in the equation gives

$$\begin{aligned} \frac{1}{\|\mathbf{v}\|^2} &= \frac{1}{x^2+y^2+z^2} \\ \implies \frac{\mathrm{d}}{\mathrm{d}\mathbf{v}} \frac{1}{\|\mathbf{v}\|^2} &= -\frac{1}{(x^2+y^2+z^2)^2} \frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}(x^2+y^2+z^2) \\ &= -\frac{1}{\|\mathbf{v}\|^4} \frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}(x^2+y^2+z^2) \end{aligned}$$

Now

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}(x^2+y^2+z^2) &= 2x\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}x + 2y\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}y + 2z\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}z \\ &= 2\mathbf{v} \cdot \frac{\mathrm{d}}{\mathrm{d}\mathbf{v}}\mathbf{v} && \text{(scalar product)} \\ &= 2\mathbf{v} && \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}\mathbf{v}} = \mathbb{1}_{3\times 3}\right) \end{aligned}$$

Therefore

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}} \frac{1}{\|\mathbf{v}\|^2} = -2\frac{\mathbf{v}}{\|\mathbf{v}\|^4}$$

□

Using this result, the following partial derivative is evaluated

$$A_i(\mathrm{u},\mathrm{v}) = \frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}\|^2}$$

By substituting $\mathbf{v} = \hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}$ and applying the chain rule for the cases $i \in \{u, v\}$, we get that

$$\frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(\mathrm{v})}\|^2} = \begin{cases} -2\frac{\hat{\gamma}^{(i)} - \hat{\gamma}^{(\mathrm{v})}}{\|\hat{\gamma}^{(i)} - \hat{\gamma}^{(\mathrm{v})}\|^4} & \text{if } i = u \\ 2\frac{\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(i)}}{\|\hat{\gamma}^{(\mathrm{u})} - \hat{\gamma}^{(i)}\|^4} & \text{if } i = v \\ 0 & \text{Otherwise.} \end{cases}$$

Curve distance term partial derivative B_i. The B_i partial is given by

$$B_i(\mathrm{u},\mathrm{v}) = \frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\hat{d}(\mathrm{u},\mathrm{v})^2} = -2\frac{1}{\hat{d}(\mathrm{u},\mathrm{v})^3} \left[\frac{\partial}{\partial \hat{\gamma}^{(i)}} \hat{d}(\mathrm{u},\mathrm{v})\right]$$

where

$$\frac{\partial}{\partial \hat{\gamma}^{(i)}} \hat{d}(\mathrm{u}, \mathrm{v}) = \begin{cases} \frac{\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}}{\|\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}\|} - \frac{\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}}{\|\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}\|} & i \in P \text{ and } i \neq \mathrm{u}, i \neq \mathrm{v} \\ \frac{\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}}{\|\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}\|} & i - 1 \in P \text{ and } i \in \{\mathrm{u}, \mathrm{v}\} \\ -\frac{\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}}{\|\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}\|} & i + 1 \in P \text{ and } i \in \{\mathrm{u}, \mathrm{v}\} \\ 0 & \text{Otherwise.} \end{cases}$$

Where P is the path of shortest length from vertex $\hat{\gamma}^{(\mathrm{u})}$ to $\hat{\gamma}^{(\mathrm{v})}$. This result is shown by analyzing the different shortest path P cases obtained when computing the discretized curve distance function. The goal is to find

$$\begin{aligned} B_i(\mathrm{u}, \mathrm{v}) &= \frac{\partial}{\partial \hat{\gamma}^{(i)}} \frac{1}{\hat{d}(\mathrm{u}, \mathrm{v})^2} \\ &= -2 \frac{1}{\hat{d}(\mathrm{u}, \mathrm{v})^3} \left[\frac{\partial}{\partial \hat{\gamma}^{(i)}} \hat{d}(\mathrm{u}, \mathrm{v}) \right] \end{aligned}$$

The difficulty is in evaluating the following term

$$\frac{\partial}{\partial \hat{\gamma}^{(i)}} \hat{d}(\mathrm{u}, \mathrm{v}) = \frac{\partial}{\partial \hat{\gamma}^{(i)}} \sum_{\mathrm{ab} \in P} \|\hat{\gamma}^{(\mathrm{a})} - \hat{\gamma}^{(\mathrm{b})}\|$$

First of all, if the i^{th} vertex is not in the shortest path from vertex u to v then $B_i(\mathrm{u}, \mathrm{v}) = 0$. When it is in P, we have three different cases. We first note that any term involving $\hat{\gamma}^{(i)}$ in the summation will either be

$$\|\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}\| \qquad \text{or} \qquad \|\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}\|$$

Assuming the neighbors are indexed sequentially and the i^{th} vertex's neighbors are $\gamma^{(i+1)}$ and $\gamma^{(i-1)}$. Now if both of its neighbors are in the path, both terms are in the distance computation. However, if $i \in \{\mathrm{u}, \mathrm{v}\}$, then the derivative depends on whether i marks the beginning or the end of the path, and P accordingly contains only one of its neighbors. Putting this all together yields

$$\frac{\partial}{\partial \hat{\gamma}^{(i)}} \hat{d}(\mathrm{u}, \mathrm{v}) = \begin{cases} \frac{\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}}{\|\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}\|} - \frac{\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}}{\|\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}\|} & i \in P \text{ and } i \neq \mathrm{u}, i \neq \mathrm{v} \\ \frac{\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}}{\|\hat{\gamma}^{(i)} - \hat{\gamma}^{(i-1)}\|} & i - 1 \in P \text{ and } i \in \{\mathrm{u}, \mathrm{v}\} \\ -\frac{\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}}{\|\hat{\gamma}^{(i+1)} - \hat{\gamma}^{(i)}\|} & i + 1 \in P \text{ and } i \in \{\mathrm{u}, \mathrm{v}\} \\ 0 & \text{Otherwise.} \end{cases}$$

These derivatives come from the chain rule applied to the fact that for a vector $\mathbf{v} \in \mathbb{R}^3$, its Euclidean norm derivative yields

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{v}} \|\mathbf{v}\| = \frac{\mathbf{v}}{\|\mathbf{v}\|}$$

which has a similar proof to that of inverse norm squared derivatives.

References

[1] Louis H. Kauffman, Milana Huang, and Robert P. Greszczuk. *Self-Repelling Knots and Local Energy Minima*, pages 29–36. Springer New York, New York, NY, 1998.

[2] K. Murasugi. *Knot Theory and Its Applications*. Birkhäuser, 1996.

[3] Andrew M. Ladd and Lydia E. Kavraki. *Motion Planning for Knot Untangling*, pages 7–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[4] Dale Rolfsen. *Knots and links*. American Mathematical Society, 2004.

[5] Robert Glenn Scharein. *Interactive topological drawing*. PhD thesis, University of British Columbia, 1998.

[6] Jun O'Hara. Energy of a knot. *Topology*, 30(2):241–247, 1991.

[7] R. B. KUSNER and J. M. SULLIVAN. Möbius-invariant knot energies. *Ideal Knots*, page 315–352, 1998.

[8] Michael H. Freedman, Zheng-Xu He, and Zhenghan Wang. Mobius energy of knots and unknots. *Annals of Mathematics*, 139(1):1–50, 1994.

[9] Chris Yu, Henrik Schumacher, and Keenan Crane. Repulsive curves. *ACM Trans. Graph.*, 40(2), may 2021.

[10] Gregory Buck and Jeremey Orloff. A simple energy function for knots. *Topology and its Applications*, 61(3):205–214, 1995.

[11] Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, mar 1999.