# Optimization of B-Spline–based Genus-1 Objects

Avik Das, Victor Huang
Adviser: Carlo Séquin

*CS Division, University of California, Berkeley*

May 16, 2011

**Abstract**

We adapt the problem of minimizing the surface bending energy of a three-dimensional shape to genus-1 objects represented as B-Splines in order to incorporate the concept of twisting. Using a standard gradient descent algorithm, we investigate smooth deformations of these objects, with the goal of a minimizing a weighted linear combination of the bending energy and the twisting energy.

*key words*: torus, bending, twisting energy, gradient descent

## 1 Background

Topologically, an object may undergo a smooth deformation, or more generally, a homeomorphism, in which the object may warp and stretch without creating any creases or sharp bends. In doing so, the object may pass through itself. The objects, before and after the transformation, are considered topologically equivalent to one another.

We consider the class of objects that can be modelled using a closed space curve, such as some variation on a torus. In rendering such a shape, the cross-section is be oriented along an orthonormal frame attached to the curve, the orientation of which is computed as a rotation minimizing frame (RMF) as described in Wang et al.[2] However, it is not guaranteed that the orientation of the cross-section at the start of the curve will match the orientation of the cross-section at the end of the curve. In the case of a planar curve, there will be no mismatch, as the RMF will not change orientation at any point. Curves exhibiting symmetry may also have no mismatch, as the accumulated rotation of the RMF will cancel out to zero. However, other curves will demonstrate a mismatch.

This cross-sectional mismatch is modelled as a twist, in which the difference in orientation between the two cross-sections is compensated for by applying

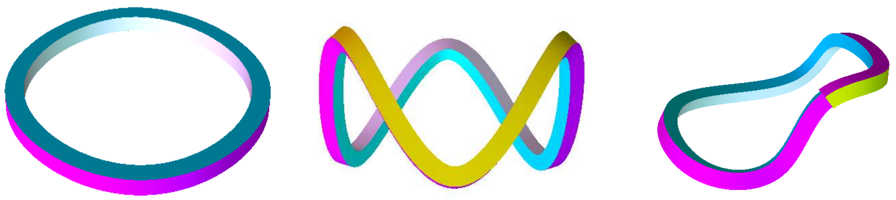


Figure 1: Various space curves with and without cross-sectional mismatch

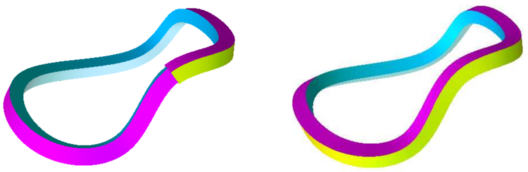


Figure 2: Cross-sectional mismatch compensated by a twist

an additional rotation to the RMF along the curve gradually. In the case of a smooth deformation, the cross-sectional mismatch may change as the shape modelled by the curve warps through space, such as when a planar torus, modelled by a circle, smoothly transitions in to one of the other shapes in Figure 1. However, a special case arises when compensating rotation, or the twist, nears 360°. To maintain the continuity of the transformation, it must be ensured that the twist not transition from just under 360° to just over 0°, even if a small twist is sufficent to align the cross-sections. This applies to twists surrounding all multiples of a full rotation.

In this way, we say that during certain continuous deformations, the twist of an object accumulates. As seen in Figure 3, a near-planar figure eight with minimal twist can be continuously deformed into a planar circle with a twist of 360°, even though all planar curves can be rendered with matching starting and ending RMF's, that is with no twist. For this reason, an untwisted planar figure eight is not topologically equivalent to an untwisted planar circle, and indeed, no homeomorphism exists between the two, despite the fact that both are genus-1 objects.

## 2 Energy Functionals

The energy functional developed for this project is composed of three parts, all dictated by the chosen representation. This energy functional is then minimized for a given input shape while ensuring all transformations are homeomorphic, that is they are continuous and preserve topological equivalence.

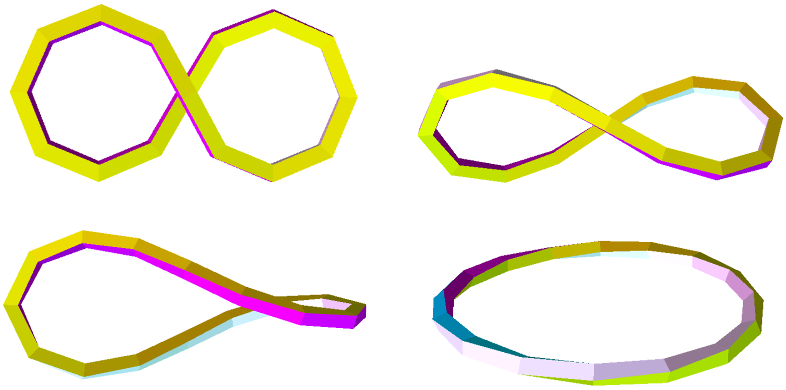


Figure 3: Continuous deformation of an untwisted near-planar figure eight into a twisted planar circle. The shape is rendered so that the individual control points are easily visible, highlighting the movement of these points during the transformation.

## 2.1 Representation

A B-Spline was used to represent all objects being optimized. This simplifies calculation of twist when considering topological equivalence classes, at the cost of restricting the scope of the representation to genus-1 objects, assuming all curves are closed. In this representation, a curve consists of a number of control points, which through sampling and interpolation can be rendered as a smooth curve. Calculation of the energy functional is performed solely on the control polygon, with the control points as the vertices. The edges between two neighboring vertices is a strut, with the $i$th strut denoted as the vector $S_i$ in the following calculations.

## 2.2 Spring Energy

For aesthetic and practical purposes, the size of the curve was restricted by imposing a penalty on all deviation of strut lengths from a predefined rest length. The penalty is calculated by modelling the struts as springs, then using Hooke's Law, summed over all the struts:

$$\sum_i \frac{1}{2} \cdot k \cdot (\|S_i\| - R)$$

where $k$ is a constant denoting the rigidity of the springs and $R$ is the predefined rest length.

## 2.3 Bending Energy

Common approaches to measuring bending energy typically consider the curvature of the surface at each point integrated over the entire surface, including those evaluated and compared by Joshi and Séquin[1]. Due to our B-Spline–based representation, we need only consider bending along one direction. Secondly, the integration of curvature across a surface is commonly used in order to prevent unbounded growth despite the decrease in curvature as a shape grows, while the spring model constituting one part of our functional already prevents such growth. Thus, the bending energy calculation in our functional considers only the deviation from colinearity between three neighboring vertices:

$$\sum_i \left( \pi - \arccos\left( \frac{S_{i-1} \cdot S_i}{\|S_{i-1}\| \cdot \|S_i\|} \right) \right)^2$$

The calculation of the inverse cosine of the normalized dot product is the standard calculation of the angle between two vectors, with the assumption that both vectors have their tail at a common point. In our case, this means that both struts have their tail in the middle of three neighboring vertices and point toward the other two. This angle is then substracted from the optimal 180°, which would be angle between the two struts if all three vertices were colinear. Finally, the square of this deviation is summed across the curve.

This formulation of the bending energy suffices because the shape of the cross-section of the object in question is constant, and thus only the sweep of the curve itself is used to compare two objects with respect to their energies. Note that this implies a Clifford torus need not be achieved if a topologically equivalent shape is optimized, and in fact the optimal in that case is the unit circle, regardless of the cross-section used to render the shape.

## 2.4 Twist Energy

Our energy functional penalizes heavily twisted objects by adding a third component to the energy functional based on the magnitude of the twist present in the shape. This requires calculating the additional rotation needed to compensate for the cross-sectional mismatch due to the sweep of the curve.

The framework used to render the B-Spline using rotation minimizing frames is able to apply an additional global twist to any curve. This twist is distributed among all sampled cross-sections, so with $n$ sampled cross-sections and a global twist of $t$, the local twist of the $i$th sampled cross-section is $\frac{i}{n} \cdot t$. Note that the cross section at $i = 0$ is sampled again at the very end, with $i = n$. Then, when the $i$th RMF is calculated as per Wang et al.[2], the RMF is further rotated around the tangent to the curve by the local twist.

Despite the fact that both the starting and ending cross-sections are sampled at the same point on the curve, the combination of the rotation minimizing sweep and the global twist may result in a mismatch between the two cross-sections. The normal vector to the curve within each cross-section is then compared to each other, and the angle between these two vectors is the mismatch between the two cross-sections. Adding this amount to the global twist will then cause the two cross-sections to align by distributing the corrective factor along the entire curve.

If the global twist is kept updated after each change to the shape, and each change is incremental, the twist will also change gradually, satisfying the accumulation of twist described above. This is because the cross-sectional mismatch is calculated with the previous global twist applied, and the mismatch is added to the previous global twist. Even if the cross-sectional mismatch without any global twist applied were to transition from just under 360° to just over 0°, the mismatch with the previous global twist applied will only be a few degrees, and adding this to the previous twist will ensure the new global twist is over 360°.

The twist energy is simply the square of the current global twist, in radians, needed to compensate for the cross-sectional mismatch.

### 2.5 The Combined Functional

For the purposes of investigating the effect of penalizing twist, the amount of penalty incurred by the twist is controllable by the user. Then, given a spring energy $S$, a bending energy $B$ and a twist energy $T$, the final energy of the shape is

$$E = S + (1 - \alpha) \cdot B + \alpha \cdot T$$

where $\alpha$ is the user-controlled parameter, known as the twist penalty weight.

## 3 Optimization

### 3.1 Gradient Descent

A standard gradient descent algorithm is used to optimize an input shape by minimizing the energy of the shape. While an analytic approach to solving a set of linear equations using gradient descent is possible, this type of analytic approach does not apply to minimizing our energy functional. Firstly, the spring and bending energy components are difficult to analyze in closed form. Secondly, the twist energy is determined by the numerical method of approximating the RMF using a sequence of frames sampled along the curve, and the accumulation of twist is calculating by updating a parameter that is used in the calculation of the change in twist in the subsequent iteration. Thus, we take the approach of varying each parameter of the optimization independently by a small amount and recalculating the energy after each change. The gradient is then the vector that points in the direction of the greatest decrease in energy with respect to each of these parameters.

Formally, we consider an independent parameter for each of the three components of the position in space of each control point, for a total of $3n$ parameters given $n$ control points. Each parameter $p_i$ is separately varied by an amount

$\Delta p$, the step size, then by $-\Delta p$. After each change, the energy of the resulting shape is calculated, denoting the first energy as $E^+$ and the second as $E^-$. Then, the $i$th component of the gradient is

$$\Delta E_i = \begin{cases} E^- - E^+ & E^- < E^0 \text{ or } E^+ < E^0 \\ 0 & \text{otherwise} \end{cases}$$

where $E^0$ is the energy of the shape before the parameter is varied. In this way, the gradient, $\Delta E =< \Delta E_0, \ldots, \Delta E_{3n} >$, will point in the direction of the largest decrease in energy, with the components weighted by the magnitude of the decrease due to each parameter. The gradient is then normalized so its magnitude is the step size, $\Delta p$, and this change is applied to the parameters to produce the result of the current iteration.

## 3.2 Step Size

The amount to vary each parameter, as well as the distance to travel in each iteration, is crucial to accurate optimization. Additionally, aesthetic considerations must be taken into account, as the optimization should proceed at a pace conducive to understanding the way in which a shape may be continuously deformed to produce an optimized variation.

For the purposes of this project, an initial and ending step size were chosen empirically. At each iteration, if no improvement could be achieved using that step size, that is the gradient using that step size was calculated to be the zero vector, the step size was halved, and the next iteration attempted to decrease the energy using the new step size. When the step size dropped below the predefined ending step size, the optimization was halted, and the shape was said to be in the optimal configuration.

The step size affects the result of the optimization, both based on the energy functional chosen, and the input shape itself. Too large a step size may result in travelling past one local minimum into a valley containing a potentially higher local minimum. Too small a step size may result in converging to an undesirably high local minimum, or an unacceptably large convergence time.

One approach is to use a line search algorithm, in which the direction of the gradient is searched to determine which distance results in the largest decrease in energy along that direction, but for simplicity, this algorithm was not implemented. Additionally, such an implementation would cause large discontinuities in the progress of the optimization, a trait undesirable when attempting to view the optimization process in order to understand the resulting transformation.

## 3.3 Limitations

Undesirable behavior when optimizing shapes is the result of two sources of error.

First, the analytical properties of the energy functional determines how the optimization convergences, and to which point it converges. In particular, the

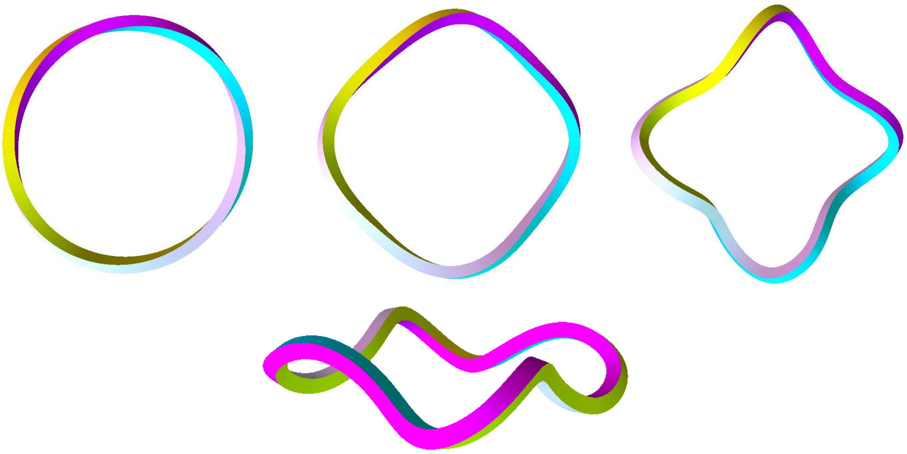


Figure 4: A planar circle with an intial twist of 360° bends inwards to minimize its twist. Optimization performed using $\alpha = 0.5$.

non-linearity, and more specifically the higher-order nature, of the functional is a major issue, as the presence of local minima may cause the optimization to progress toward a configuration that decreases the energy of the shape without resulting in the desired final configuration.

The second source of error is in how well the energy functional approximates the intentions of the authors. For example, it was discovered that the use of the global twist to calculate the twist energy, while accurately modelling the undesirability of the twist in many common scenarios, did not account for the case in which a large amount of twisting cancelled itself out due to the symmetry of a particular curve.

These limitations are apparent when a planar circular torus with an initial twist of 360° is optimized, as seen in Figure 4. Instead of folding itself into a figure eight, the planar twisted circle bends inwards. As expected, this does increase the bending energy significantly, in this particular case by more than five-fold. However, the twisting energy decreases by thirty-six times! This is related to the symmetry of the final shape, combined with the non-planar configuration, in which a great deal of the twist cancels out from one end of the curve to the other. At this point, the optimization cannot proceed further, due to the presence of a local minimum. A plot of each component of the energy as the optimizing progresses is shown in Figure 5.

# 4 Results

The most fundamental benchmark for this project is the unfolding of a planar, untwisted figure eight into a planar circle with a twist of 360°. This is shown in Figure 3, with the control points clearly visible. It must be noted that the

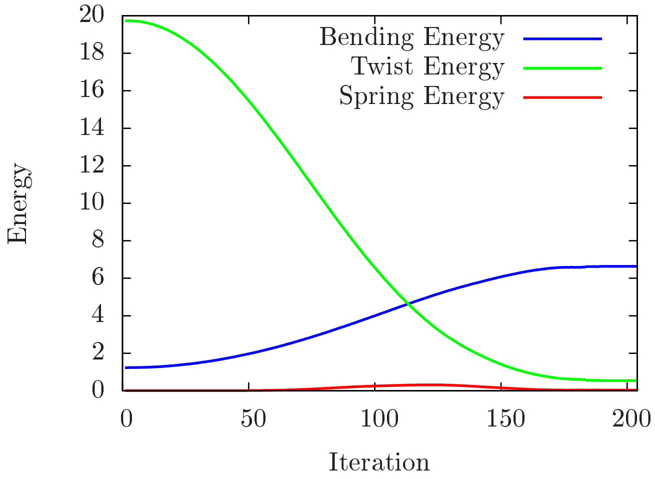


Figure 5: Energy Components for Figure 4 by Iteration

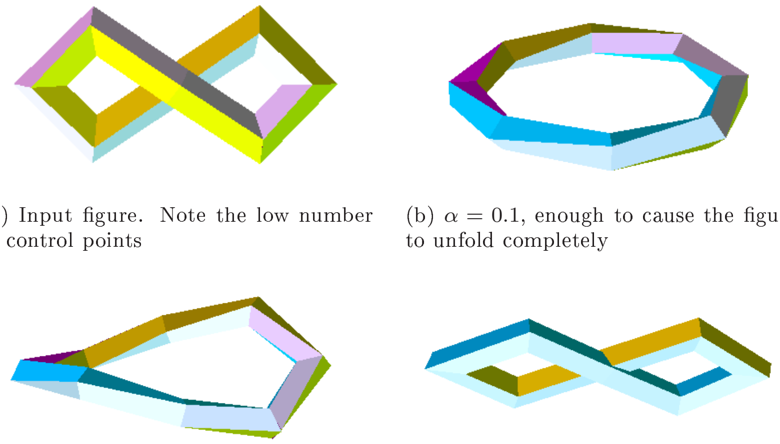
input shape has one of the center points perturbed off the plane in which the rest of the points lie; this is to hint to the optimization which direction to start proceeding. Without perturbing one point, the energy will not decrease when any individual parameter is varied independently of the other parameters. The optimization is performed using $\alpha = 0$.

The result of the optimization of this input is highly sensitive to the value of $\alpha$. Empirically, it was determined that values of $\alpha$ below 0.048324 resulted in the shape unfolding into a twisted circle, whereas any value of $\alpha$ larger than this particular number resulted in very little change from the initial configuration.

Intermediate configurations between a planar, untwisted figure eight and a planar, twisted circle were achievable when the input shape contained fewer control points. With the correct values for this optimization parameter, the shape attempted to balance both the bending energy, which is minimized by a circular configuration, and the twisting energy, which is minimized by the figure eight. The result was an intermediate state of unfolding. These results are summarized in Figure 6.

While a planar circle with a twist of 360° demonstrated the limitations of our system, the same shape with a slight perturbation of one of the control points off the plane in which the other points lie is indeed correctly optimized, albeit in an undesirable manner. The perturbation effectively signals to the optimization that some form of folding out of the plane should occur, but the optimization proceeds locally before producing the desired planar, untwisted figure eight. This result is summarized in Figure 7.

Finally, testing with a doubly-looped circle, both with and without an initial twist, demonstrate successful optimization. A doubly-looped circle with an initial twist of 360° successfully unfolds itself into a planar, untwisted, single-



(a) Input figure. Note the low number of control points

(b) $\alpha = 0.1$, enough to cause the figure to unfold completely

(c) $\alpha = 0.15$, resulting in an intermediate state

(d) $\alpha = 0.2$, enough to prevent any unfolding

Figure 6: Intermediate states of unfolding in order to balance bending and twisting energy for different vaues of $\alpha$.

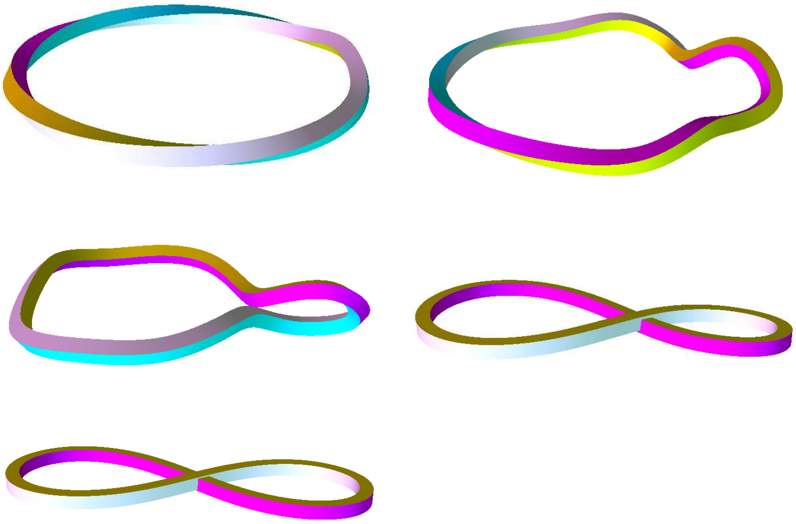


Figure 7: A nearly-planar circle with a twist of 360° folds itself into a figure eight. The optimization proceeds locally before balancing all the components of the shape. This optimization was performed with $\alpha = 0.5$.

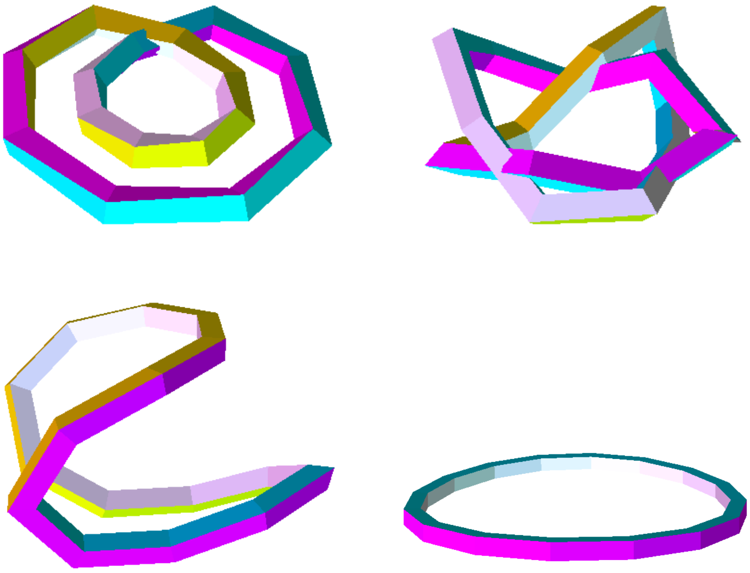


Figure 8: A double-looped circle with an initial twist of 360° unfolds into a planar, untwisted circle, with $\alpha = 0.5$. The camera orientation differs in each capture to highlight the prominent features in the intermediate configurations.

looped circle when optimized with $\alpha = 0.5$, while an untwisted doubly-looped circle successfully unfolds into a planar single-looped circle with a twist of 360° when optimized with $\alpha = 0$. These results are summarized in Figures 8 and 9 respectively.

## 5 Acknowledgements

We would like to thank Satish Rao and Umesh Vazirani for providing valuable insight regarding gradient descent, and Carlo Séquin, without whose advice and guidance, this project would not have been possible.

## References

[1] Pushkar Joshi and Carlo Séquin. Energy Minimizers for Curvature-Based Surface Functionals. *Computer-Aided Design & Applications*, 2007.

[2] Wenping Wang, Bert Jüttler, Dayue Zheng, and Yang Liu. Computation of Rotation Minimizing Frame in Computer Graphics. Technical report, University of Hong Kong, March 2008.

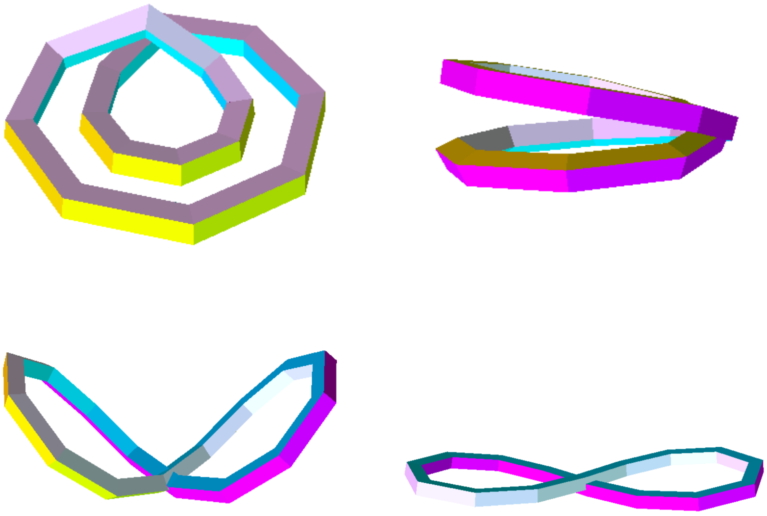


Figure 9: A double-looped circle with no initial twist unfolds into a planar, twisted circle, with $\alpha = 0$. The camera orientation differs in each capture to highlight the prominent features in the intermediate configurations. The second half of the optimization is omitted, as it does not differ from Figure 3.