Bending and Torsion Minimization of Toroidal Loops

Avik Das

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-165
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-165.html

June 8, 2012

Acknowledgement

The author would like to thank Victor Huang for participating in the initial stages of the project, 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.

Bending and Torsion Minimization of Toroidal Loops

Avik Das
Adviser: Carlo H. Séquin

CS Division, University of California, Berkeley

June 7, 2012

Abstract

We focus on an optimization problem on parameterized surfaces of genus one. In particular we trade off the penalty functions for bending a toroidal path and for applying a twist to it and aim to find local minima of this cost function. This analysis forms a key element in demonstrating the different regular homotopy classes of tori. A generalization of this surface optimization, which considers curvature as well as any shearing of its parameter grid, may be used to find the most optimal direct path from an arbitrary closed manifold of genus one into one of the four basic representatives of the four regular homotopy classes of tori.

key words: torus, bending energy, twisting energy, gradient descent, regular homotopy.

1 Problem Statement and Background

Hass and Hughes show that an orientable 2-manifold of genus one immersed in $\mathbb{R}^3$ has four regular homotopy classes[1] and Séquin constructs representatives for each class[2]. Any torus immersed in $\mathbb{R}^3$ can be smoothly deformed into another torus in the same regular homotopy class without introducing any tears, creases or pinch-off points with infinite curvature. A surface may, however, pass through itself, as long as it does not introduce these problems. The converse holds: no regular homotopy transformation exists between objects in different regular homotopy classes.

In particular, to show that a particular object belongs to a certain regular homotopy class, it suffices to demonstrate a regular homotopy from that object into some object known to belong to the same class. However, finding such a transformation may be difficult, and ideally, some automated system would be able to construct the regular homotopy given only the start and end configurations.

Séquin demonstrates a number of regular homotopy moves, particularly the figure-8 cross-over move[2]. In this move, a figure-8 (Figure 1a) can be unfolded into a circle – from a torus of type O8 to one of type OO in the paper's terminology – but this introduces a twist of $\pm 360°$ (Figure 1c). Another application of the move can deform the basic representative of the OO class, essentially circle with zero twist (Figure 2a), into a figure-8 with $\pm 360°$ of twist (Figure 2b), and beyond into another circle (Figure 2c). In the ending configuration, we find the overall twist has changed by exactly $\pm 720°$. All these objects are in the same regular homotopy class, so twist can be considered modulo $720°$.

To visualize regular homotopies of tori without specifying any intermediate configurations, we aim to construct penalty functions, i.e. energy functionals, defined on a torus and a system that smoothly deforms the torus so as to minimize that energy. We choose a structured representation of the tori consisting of relatively few independent parameters, allowing the optimization to progress at interactive speeds. The energy function then considers both the integrated bending and the overall twisting. By adjusting the relative weights for these two terms, we can choose to obtain either circular loops with some possible non-zero twist or some true 3D space curve with minimized twist.

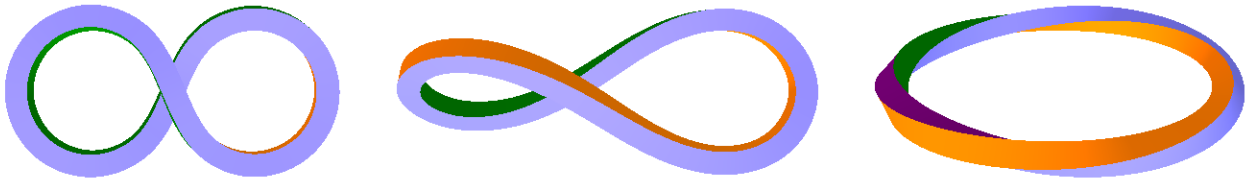


Figure 1: A figure-8-shaped minimum-rotation sweep being unfolded into a circle using the cross-over move: (a) Figure-8 starting configuration with zero twist; (b) Figure-8 unfolding; (c) Circular ending configartion with 360° of twist.

Figure 2: The full figure-8 cross-over move adds ±720° of torsion: (a) Circular start configuration with zero torsion; (b) Figure-8 with 360° of twist; (c) Circular end configuration with 720° of twist.

2 Torus Representation and Energy Functionals

The sweep path of the toroid is rendered as a closed B-spline, regularly sampled in parameter space with a user-defined number of segments. The few control points of the B-spline allow the user to define the geometry of the starting configuration. The end-to-end torsion is then calculated using a rotation-minimizng frame[3][4], and some compensating twist is introduced to generate properly aligned end-frames. This artificially introduced twist is uniformly spread out over the whole B-spline loop.

The user can optionally introduce additional twist. In most cases, this twist will be a multiple of 360° in order to ensure the surface corresponds to a torus with proper parameterization, in which all parameter lines close onto themselves if one goes once around the loop. As the sweep curve is deformed gradually, the overall compensating twist is adjusted incrementally from one iteration to the next, to guarantee that the end-to-end connectivity of the torus does not suddenly change by ±360°.

2.1 Representation

Even though the sweep curve is rendered as a "smooth" B-spline, all calculations of energy and all optimization moves are carried out on the control polygon. This dramatically reduces the degrees of freedom in the system and the amount of derivative computations in the optimization process. An edge between two subsequent control vertices is called a strut, with the ith strut denoted as the vector S_i in the following calculations.

2.2 Spring Energy

To keep the curve length roughly the same as in the original input, and to avoid tedious recscaling steps, the average length of all the initial struts $\|S_i\|$ is defined to be their desired rest length R during the optimization process. The struts are then modelled as ideal springs, and Hooke's law is applied to each strut and the

result summed over all the struts:

$$E_S = \sum_i \frac{1}{2} k \left(\|S_i\| - R \right)^2$$

2.3 Bending Energy

Common approaches to calculating the bending energy of a 2-manifold typically consider the mean curvature of the surface at each point integrated over the entire surface. Since our study concentrates on thin, tube-like structures with a constant cross section, we restrict ourselves to considering only the amount of bending found in the sweep path itself. The bending energy of a smooth curve, calculated as an arc-length integration of curvature squared, is not scale-invariant. But due to the strut length restriction imposed by the spring energy, only the angles between consecutive struts need to be considered. Each control point on the curve contributes the square of its deviation from collinearity:

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

2.4 Twist Energy

Since we are interested not just in the pure geometrical shape of a toroid under investigation, but also in the deformation of a parameter grid embedded in its surface, we have to add a third term to our cost function corresponding to the twist energy. This is a measure of the deviation of a given state of the toroid from the geometry produced by a pure rotation minimizing frame (RMF)[3][4].

Despite the fact that both the starting and ending cross-sections are sampled at the same point on the curve, the incremental forward projection of the RMF, calculated as per Wang et al. [5] [6], may result in a mismatch between the two cross-sections, as seen in Figure 3a. The azimuthal rotation angles between the two cross-sections are compared to each other, and the difference in orientation is compensated by applying an additional global twist to the whole curve, so that the beginning and ending coordinate frames line up and the longitudinal parameter lines in the toroid surface properly join with themselves after one pass around the loop, as seen in Figure 3b.

This additional twist is distributed evenly among all line segments of the sweep path; thus with a global twist of t and with n sampled cross-sections, the local twist along each of the n line segments is $\frac{i}{n} \cdot t$. After a match of the two end-frames has been established, the user can add additional global twist in increments of $\pm 360^\circ$, which keeps all longitudinal parameter lines properly connected to themselves.

The simplest way to define a twist penalty is to use the square of this total twist, comprising the component needed to establish the matching coordinate frames as well as any additional twist added by the user.

$$E_T = (\text{Built-in end-to-end azimuthal mismatch} + \text{user-added twist})^2$$

Once this toroidal end-around connectivity has been initialized, it is important that it be maintained throughout the whole optimization process. Thus the global twist is updated incrementally after each change to the curve shape, so that the two end frames keep their alignment. This prevents the toroid from suddenly experiencing $\pm 360^\circ$ jumps in the twist, as the update looks for a minimal change in the global twist to keep the two coordinate frames aligned.

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

$$E = E_S + (1 - \alpha) \cdot E_B + \alpha \cdot E_T$$

First the spring constant k in E_S is adjusted so that obtaining struts that are all close to the rest length has priority over any other adjustments to the control points, without making the optimization problem too stiff. A value of $k = 10$ worked well.

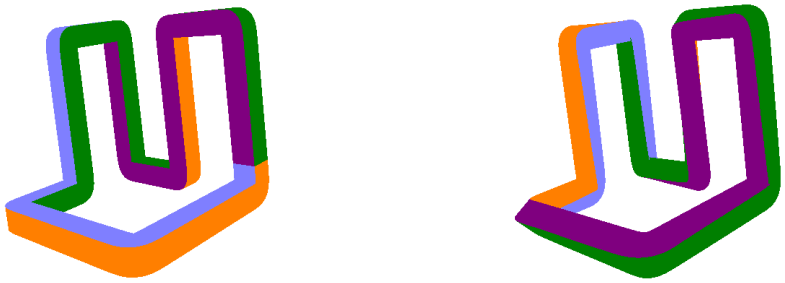


Figure 3: (a) The forward projection used for the rotation minimizing frame causes the starting and ending frames to misalign. (b) Gradually adding in 180° of twist across the sweep causes the frames to align.

Then the user can play with the parameter α, which trades off the twist penalty weight against bending costs. Adjusting this ratio is the primary way to affect the outcome of the optimization.

3 Optimization

3.1 Gradient Descent

A standard gradient descent algorithm is used to optimize the given input toroid by minimizing the energy of the final configuration. While an analytic approach to solving a set of linear equations using gradient descent is possible, this type of analytic approach is not well suited to minimizing our energy functional. Firstly, the bending energy components by themselves are already 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 in subsequent iterations, a process that is difficult to capture in closed form.

We employ finite differencing to determine the influence of each primary parameter on the overall cost of the resulting configuration. Each parameter – in our case, each coordinate component of each control point – is varied independently by a small amount, and the resulting change in the energy is calculated. Given a sweep curve defined by n control points, there are a total of $3n$ parameters. Each of these parameters p_i is first changed by an amount $+\Delta p$, resulting in a new toroid energy E_i^+, and subsequently by $-\Delta p$, resulting in a toroid energy E_i^-. The gradient, that is the direction of greatest decrease in energy, is the vector with components $-\frac{\Delta E_i}{\Delta p_i} = \frac{E_i^- - E_i^+}{2\Delta p_i}$. A small step is then taken along this combined vector.

3.2 Step Size

The amount by which parameters should be varied to reliably calculate the energy gradient vector, as well as the distance along that vector that should be traveled in one iteration, are crucial to an efficient and stable optimization process. In addition, if we are interested in creating a real-time interactive visualization or a smoothly evolving demonstration movie, additional constraints on the step size and thus on the optimization speed, may come into play.

For the purposes of this project, the initial finite-differencing step is set to 0.04. This implies that each control point is moved by 4% of the desired rest length of a strut to determine the influence on the total energy.

The initial optimization step size is set to 0.04. This means that the gradient vector is normalized to the rest length of a strut, and each control point is moved by 4% of the length dictated by the normalized gradient vector.

Before each iteration step is carried out, it is checked that the intended step would indeed result in a reduction in the overall energy function. If this is not the case, the iteration step size is cut in half, and subsequent iterations attempt to decrease the energy by using this reduced step size. When the step size drops below a predefined ending step size, empirically chosen as 4×10^{-6}, the optimization halts, and it is inferred that the toroid has reached a local optimum.

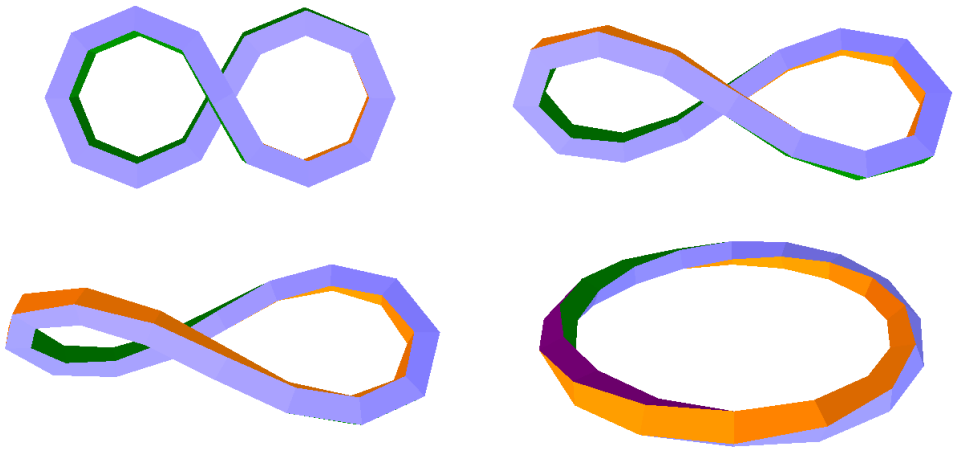


Figure 4: Continuous deformation of an untwisted near-planar figure-8 loop into a twisted planar circle. The shape is rendered so that the individual control points are easily visible, highlighting the movement of these points during the transformation.

The iteration step size has an important influence on the progress of the optimization. Too large a step size may result in flying past one local minimum into a valley containing another local minimum of possibly higher energy, or it may cause tunneling through a barrier into an inappropriate region of the energy landscape. Too small a step size will result in unacceptably long convergence times.

A more sophisticated optimization approach may employ a line search algorithm, in which the direction of the gradient is searched in a more refined manner to find an appropriate energy minimum on that line. This algorithm was not used for this project. A first reason was simplicity and that fact that satisfactory results were obtained with a straight-forward gradient-descent algorithm. Another concern was also that such a line-search algorithm may result in a rather uneven rate of progress. This would be rather undesirable when trying to create real-time interactive visualizations or smoothly progressing movies of the evolution of a toroidal shape.

3.3 Elementary Test Cases

The most fundamental benchmark for this project is the unfolding of a planar, untwisted figure-8 loop into a planar circle with a twist of 360°. This evolution is shown in Figure 4, with the control points implicitly visible at the joints between subsequent straight line segments. This optimization has been performed using $\alpha = 0$, i.e., no twist penalty at all.

It must be noted that the input shape has one of its two center points slightly raised above the plane in which the rest of the points lie; this is to break the symmetrical "log-jam" and give the optimization a clear direction in which to start its evolution. Without breaking the symmetry of this configuration in some way, the optimization will not get under way.

In Figure 5, a planar circle with a twist of 360° with a slight perturbation in one control point is correctly optimized, albeit in an unexpected, not very aesthetically pleasing 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-8.

3.4 Effect of α Values on Optimal State

Because the value of α determines the relative importance of bending and twisting on the energy functional, it is logical to hypothesize that certain values of α will result in an intermediate state that balances both factors. To test this hypothesis, we start with a partially unfolded figure-8, half-way between an untwisted figure-8 and a twisted planar circle. As seen in Figure 6, sufficiently low values of α cause the unfolding

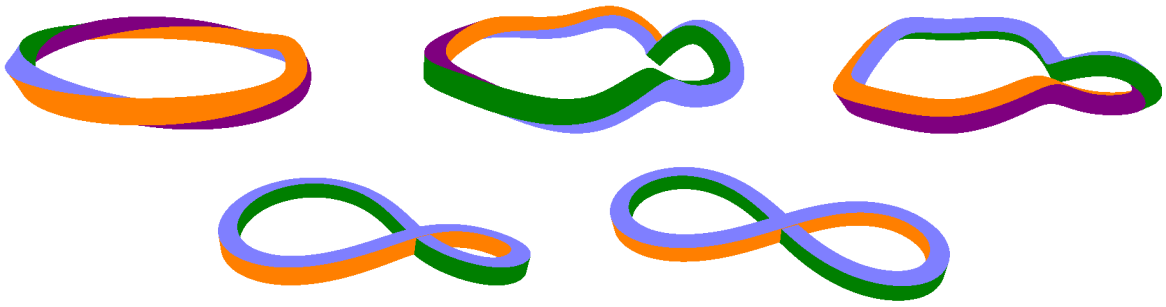


Figure 5: A nearly-planar circle with a twist of 360° folds itself into a figure eight through a sequence of somewhat counter-intuitive moves. This optimization was performed with $\alpha = 0.5$.

process to proceed further toward a planar circle, while sufficiently high values of α cause the shape to fold back toward a figure-8. As expected, intermediate values of α, between 0.09 and 0.12, cause the shape to stabilize near the half-way point, balancing the bending and the twisting.

Having verified the existence of such an intermediate state, we can attempt to achieve the intermediate state starting at either a twisted planar circle or an untwisted figure-8. For the 16-segment figure-8 in Figure 7a, however, the result of the optimization is highly sensitive to the value of α. Empirically, it was determined that α values below 0.048324 resulted in the shape unfolding into a twisted circle, whereas any value of α larger than this particular number resulted in the shape settling back into a planar figure-8.

It appears that given this starting point and α values above the threshold, changes to a single parameter do not decrease the bending enough to allow the gradient descent algorithm to proceed with the unfolding process. Below the threshold, the unfolding can occur, and because the value of α is well below the range required to achieve the intermediate state, the optimization is driven all the way to a twisted circle. To overcome this, a significantly larger initial perturbation can be introduced, as shown in Figure 7b. From this starting point, any value of α greater than 0.0825 causes the shape to settle back into a figure-8. Slightly lower values, such as $\alpha = 0.8$, allow the unfolding to proceed, and the value of α is still sufficiently large enough to achieve a state similar to the half-way point, just like the result in Figure 6d.

Finally, the sensitivity to α is present when starting at a perturbed, twisted planar circle, as in Figure 7c. Values of α below 0.1593 result in the shape settling into a planar circle. This is because changes to a single parameter do not reduce the twist enough to compensate for the increase in bending when near the starting configuration. Values of α above this value succeed in starting the folding process, and because the α value is above the range required for an intermediate state, the folding concludes in a planar figure-8.

Intermediate configurations are achievable when the input shape contains fewer control points. With only 8 control points in the loop and suitable parameter values for α, the toroid shape manages to achieve an intermediate state, as shown in Figure 8. The same values of α result in the same ending configurations when starting with a twisted planar circle. In both cases, the low number of control points allow the gradient descent algorithm to discover optimal moves away from the starting state, even when α is within the range required to achieve an intermediate state.

3.5 Unexpected Behavior

There were some optimization sequences that surprised us, where we first thought that there was something wrong with our torus representation or with our gradient-descent process. One such problem sequence is shown in 9. We start with a perfectly symmetrical circular loop with 16 control points and a twist of 360°. But rather than unfolding into a planar figure-8 loop as in Figure 5, this shape ends up in a 4-fold symmetrical hoop with four wavy "sides". When we monitor the behavior of the various energy components (Figure 10), we see that the twist energy is reduced dramatically, by a factor of 36, while there is only a modest increase in bending energy of about a factor of 5.

This solution still leaves us with approximately 60° of twist, and the bending energy is almost twice the bending energy present in an untwisted figure-8, which is the solution we would expect given the starting point and $\alpha = 0.5$. So what is going on here?

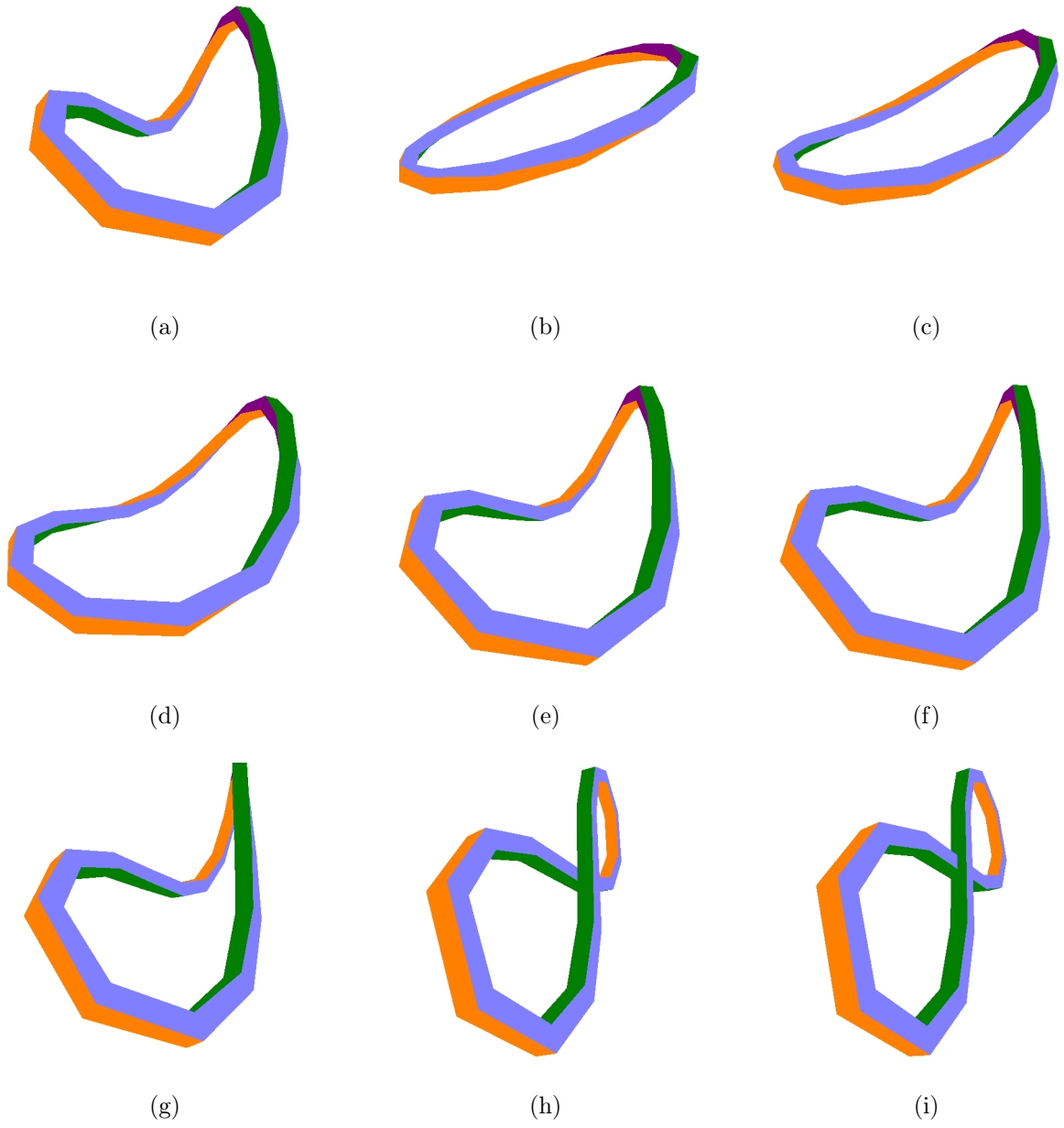
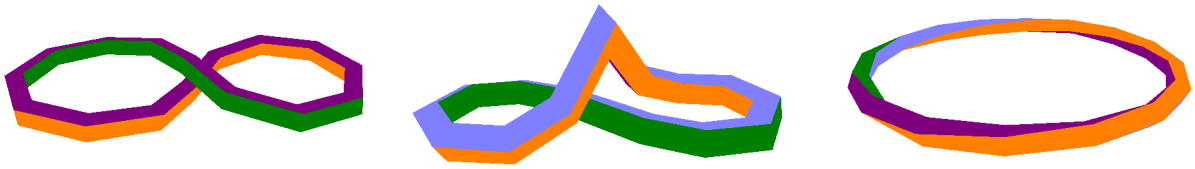
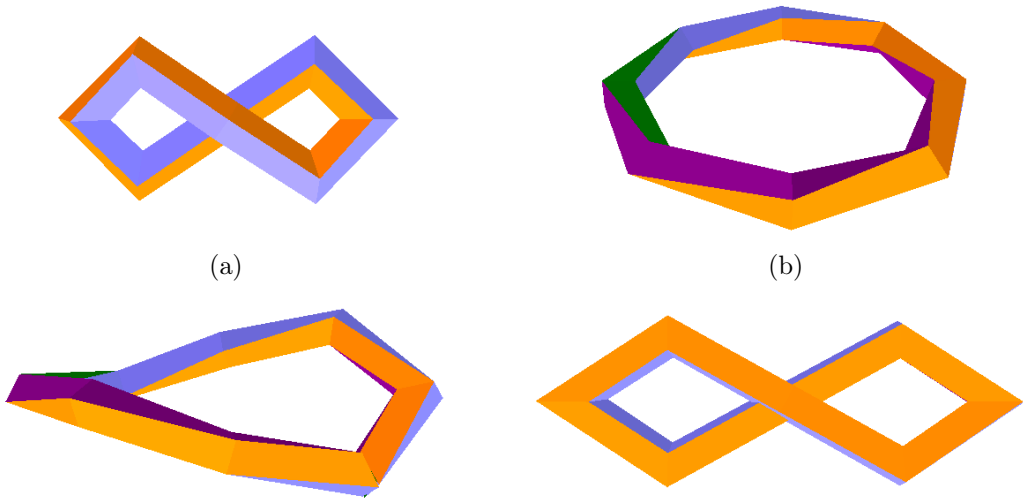


Figure 6: Intermediate states of unfolding in order to balance bending and twisting energy for different vaues of α. (a) Initial shape: a half-unfolded figure-8, with two osculating planes opposite to each other at a 90° offset. The figure has 16 control points; (b) $\alpha = 0$, low enough to cause the figure to unfold completely into a circle; (c-d) $\alpha = 0.07$ and $\alpha = 0.08$ respectively, low enough to cause the figure to unfold most of the way to a circle; (e-g) $\alpha = 0.09$, $\alpha = 0.11$ and $\alpha = 0.12$ respectively cause the figure to remain in the intermediate state; (h-i) $\alpha = 0.15$ and $\alpha = 0.2$ respectively, high enough to cause the figure to fold back nearly to a figure-8.



(a) (b) (c)

Figure 7: Starting points for attempts to achieve an intermediate state. (a) No intermediate state achievable; (b) a larger initial perturbation allows an intermediate to be achieved; (c) no intermediate state is achievable.



(a) (b) (c) (d)

Figure 8: Intermediate states of unfolding in order to balance bending and twisting energy for different vaues of α. (a) initial shape: a slightly perturbed figure-8 loop. Note the low number of control points; (b) $\alpha = 0.1$, low enough to cause the figure to unfold completely into a circle; (c) $\alpha = 0.15$, resulting in an intermediate state; (d) $\alpha = 0.2$, resulting in an optimal state closer to the figure-8 shape. The results hold even when starting with a twisted planar circle.

Our solution Possible solution

Figure 9: A planar circle with an initial twist of 360° bends inwards to minimize its twist. Modeled with 16 segments, and optimization performed using $\alpha = 0.5$. The last image shows a four S-shaped curves joining together in such a way as to absorb the 360° of twist. This resembles our the solution found by our system.

Actually, we could synthesize a perfect, twist-free solution with a rectilinear cubist control polygon that produces four planar, S-shaped curve segments which join together with four vertical junctions where the Frenet frame makes an instant 90° turn. This accommodates an overall twist of 360° with no twist penalty, since all four junctions allow an RMF frame to pass through completely undisturbed.

This example demonstrates that there is a local minimum in which this optimization run is getting trapped. The perturbation in the initial state depicted in Figure 9a is slightly different from that used in Figure 7c, and this can lead to a different outcome altogether! First, we thought that this is a manifestation of "local twistedness," and we started to look for ways to overcome this problem. On second thought, we then realized that "local twistedness" is not a good concept, at least not until we introduce separate variables for the azimuthal rotation at each control vertex.

In our model, the twist in the azimuth parameter of the toroidal tube is not explicitly considered until an RMF has been run over the entire curve. Whatever end-to-end azimuthal mismatch is detected by our system is assumed to be optimally distributed along the curve, that is the needed azimuth increments are uniformly spread out along the curve. Under this assumption, there can never be any local concentration or non-uniformity of the twistedness.

Consider the specific case of a nearly straight curve, where the RMF maintains a mostly fixed orientation in space. Suppose the control points are disturbed minimally to introduce a slight helical deformation into the curve. This forces the Frenet frame of the curve to rotate strongly, despite the minimal displacements of control points. However, the rotation minimizing frames along the toroidal tube remain mostly unaffected. Thus, there is some twist due to the delical deformation but no reasonable way to derive a twist penalty from the RMF. The curve must remove the deformation using only the bending penalties, which may be a very slow process.

4 Application to Torus Eversion

Chéritat demonstrates the process of turning a torus inside out via a regular homotopy[7]. However, we wish to find a transformation that is energetically optimal. Séquin has found a symmetrical half-way point for a possible eversion process[2], shown in Figure 11a, with the goal that it be perturbed slightly and it would

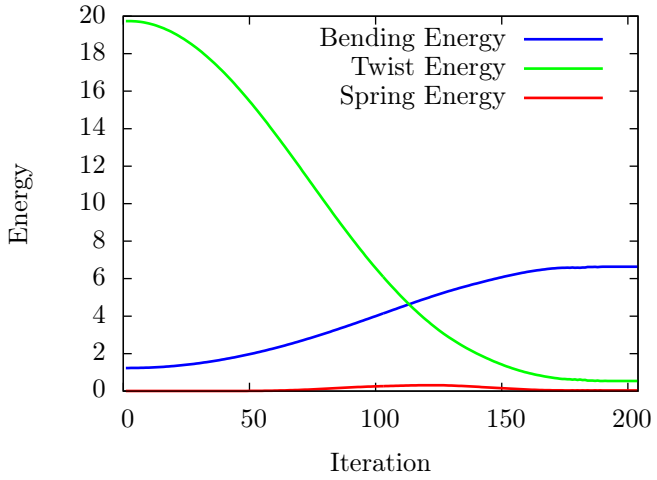


Figure 10: Energy Components for Figure 9 by Iteration

then evolve under an energy minimizer such as our system. To achieve the latter, two changes are required: a suitable representation for a shape that contains a Klein bottle mouth and a modified energy functional that drives the energy minimization toward completing the eversion.

4.1 Representation

The sweep path used previously cannot represent the torus eversion process because throughout the transformation, the geometry contains two Klein bottle mouths that cannot be captured using a simple sweep. A suitable representation must allow us to visualize the eversion process starting at a halfway point up until the point right before one of two the segments between the mouths is ready to be collapsed as in Chéritat's visualization. Because this geometry contains two arms joining two Klein bottle mouths, the arms are modelled as separate, non-closed sweeps with their endpoints constrained so as to always lie along the same line segment, but with differing radii. The mouths can then be visualized as half of a torus joining one endpoint of the first curve with the corresponding endpoint of the second. The mouths are present as fixed shapes througout the entire eversion process, so they do not affect the energy.

4.2 Energy Functional

The first change requires the calculation of the twist across two separate curves, neither of which is closed. We assume no twist is present in the Klein bottle mouths, so two corresponding endpoints are always aligned. At each endpoint of one curve, a coordinate frame is defined such that one frame would superimpose on top of the other if one were projected forward straight onto the other. The forward projection used to calculate the RMF is carried out along the sweep of the curve, starting at one end, and the amount by which the projected frame misaligns with the ending frame is the twist along that arm. The same starting and ending coordinate frames are used for the second arm due to the alignment condition at the mouths.

Two additional terms are added to the functional. First, each control point is allowed to have a variable radius associated with its cross section, and the radii at the endpoints are held constant. One term in the energy function penalizes large changes in this radius between consecutive control points. If additional penalties were associated with two cross sections intersecting, it would force regions with high curvature to be thinner to prevent such intersections. A second term in the energy functional penalizes large changes in the relative lengths of consecutive struts. This replaces the spring energy in the previous functional, since the struts near the endpoints are held constant in length.

The bending energy in the previous functional is retained. These components are combined in various ways to form different energy functionals that aid in the eversion process.

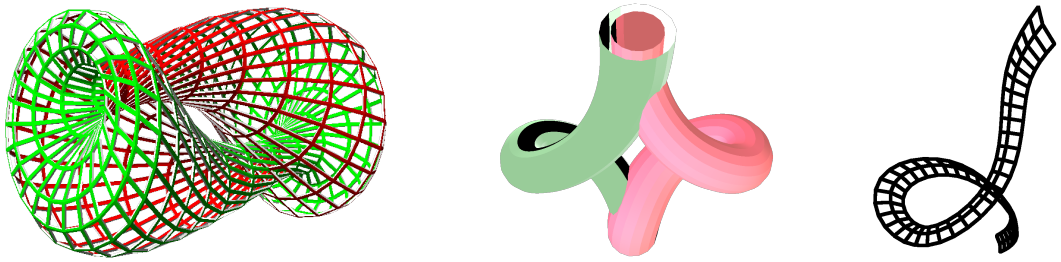


Figure 11: The symmetrical half-way point for a possible eversion process. (a) Séquin's rendering[2]. Note the twist in each arm; (b) the same configuration represented and optimized for minimal twisting using our system; (c) the stripe on the green arm in isolation as it twists around 180° along the sweep.

4.3 Optimization

From Séquin's analysis[2], we know that in the configuration shown in Figure 11a, the amount of twist in each arm is ±180°, and these twist specifications were used as the overall constraints in our simulation. This forces the arms to take on a configuration in space so that the RMF provides this 180° of twist, yielding the configuration in Figure 11b. The stripe along the green arm is shown in isolation in Figure 11c, and it can be seen that this RMF stripe twists nearly 180° as it winds around the helical sweep. This allows the arm to accomodate the necessary twist without requiring additional twist to be added on top of the RMF.

Starting from this configuration, the only way to complete the actual eversion process is to deliberately break the symmetry. We force all the twist in the green arm to be transferred to the red arm by penalizing any twist in the green arm, and giving the red arm a twist of 360°, ensuring the total twist across both arms will not change.

The energy landscape under this functional contains a strong local minimum in the neighborhood of half-way point, making it difficult for an optimization process to escape this valley and move to a lower energy region. To create a time-dependent visualization of our desired result, we separate the eversion into a number of discrete steps, each time imposing a different contrainst to promote evolution toward a particular intermediate configuration.

We start by removing any control points in the green arm that do not lie near the endpoints and adding a few more to the red arm. This allows the red arm to be flexible enough to perform the desired out-of-plane rotation. Any twist in the green arm is penalized.

First, the red arm is given a twist of 360° and the optimization is run, giving Figure 12a. This configuration is then given a twist of only 270°, meaning the end caps will line up when red arm takes on a configuration in space so that the RMF contributes 90° of end-to-end azimuthal mismatch, yielding Figure 12b. In this fashion, starting with the configuration obtained by the previous optimization and decreasing the initial twist to 180°, 90° and no twist, the optimization yields Figures 12c-e respectively. In each case, the green arm is held constant, and in Figure 12f, the green arm is shortened to better visualize the final stages of the eversion.

The process of starting at an intermediate configuration and altering the twist constraint imposed on the system essentially tilts the energy landscape, allowing the optimization to proceed in a direction that would have been considered as a direction of increasing energy in the untilted landscape. The fact that we need to perform the optimization in multiple steps shows that a single optimization starting from the symmetric half-way point and ending at a nearly circular configuration is not achievable using the current energy functional.

5 Conclusions

With these experiments we have demonstrated that simple CAD models can be constructed with suitable cost functions that will evolve toroidal structures to their simplest representative within the appropriate regular homotopy class. When applying this process to a partially everted torus represented within our

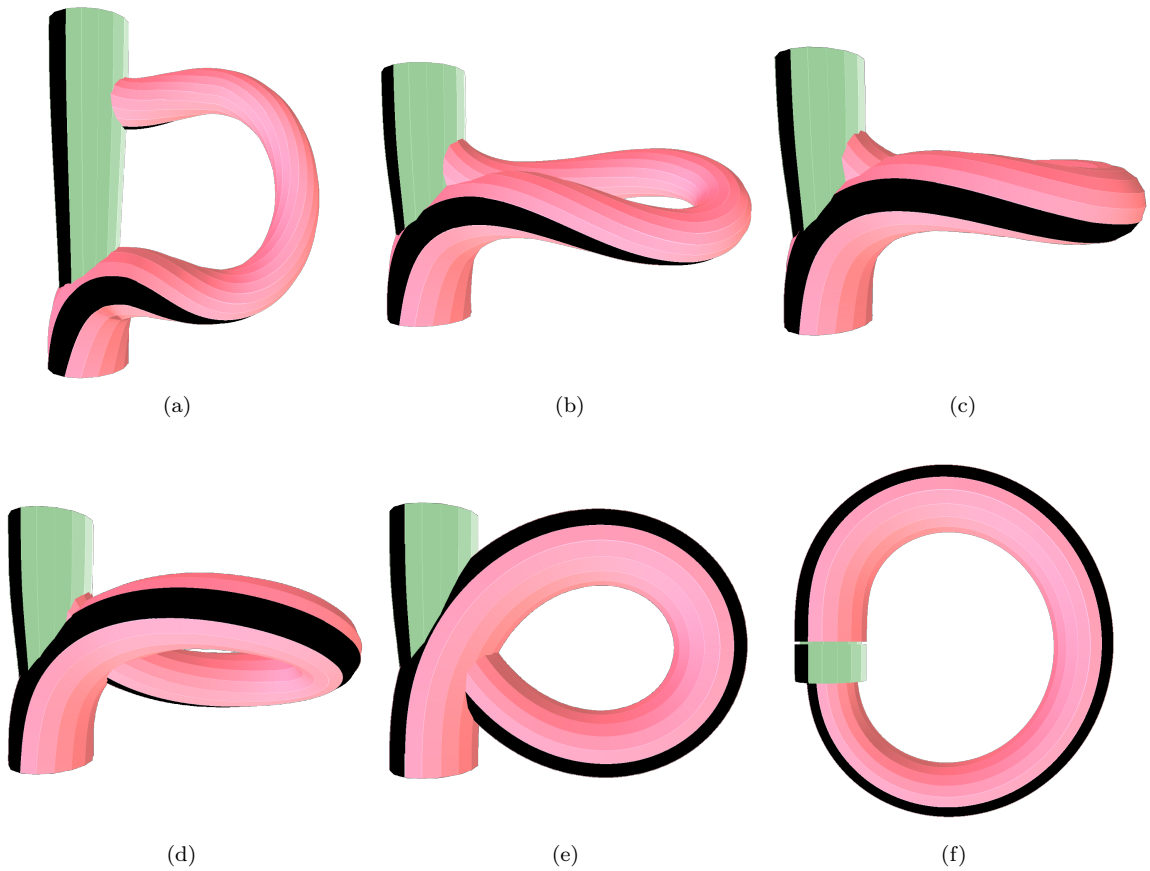


Figure 12: The eversion process. (a) The red arm absorbs all 360° of twist while the green arm straightens; (b)-(e) the red arm undergoes an out-of-plane rotation in order to remove the 360° of twist. (f) the torus right before the green arm is ready to be collapsed.

system, the energy landscape is not conducive to an automatic eversion via an optimization process. This suggests the scope of further research to determine if a suitable energy functional can be obtained, as in the case of the sphere eversion.

6 Acknowledgements

The author would like to thank Victor Huang for participating in the initial stages of the project, 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] J. Hass and J. Hughes. Immersions of Surfaces in 3-Manifolds. *Topology*, 24(1):97–112, 1985.

[2] Carlo H. Séquin. Tori Story. In *Bridges Conference Proceedings*, pages 121–130, Coimbra, Portugal, July 27-31, 2011.

[3] Richard L. Bishop. There is More than One Way to Frame a Curve. *The American Mathematical Monthly*, 82(3):246–251, March 1975.

[4] F. Klok. Two moving coordinate frames for sweeping along a 3D trajectory. *Computer Aided Geometric Design*, 3:217–229, November 1986.

[5] W. Wang and B. Joe. Robust computation of the rotation minimizing frame for sweep surface modeling. *Computer-Aided Design*, 23(5):379–391, 1997.

[6] Wenping Wang, Bert Jüttler, Dayue Zheng, and Yang Liu. Computation of Rotation Minimizing Frame in Computer Graphics. *ACM Transactions on Graphics*, 27(1), March 2008.

[7] A. Chéritat. Torus eversion: turing a torus inside out. Video (49sec) – http://www.youtube.com/watch?v=kQcy5DvpvlM, 2008.