Knot-Spanning Surface Generator

Rahul Khardekar Xiaorui Chen

May 15, 2006

Abstract

A knot spanning surface is an orientable or non-orientable surface that is bounded by a knot. We present a tool that can construct and visualize knot-spanning surfaces for user-defined knots. Our tool can either display a non-orientable surface or an orientable surface by inflating the non-orientable surface.

We first allow the user to draw a knot in two dimensions using a sketching tool. The sketching tool then outputs the geometry into a SLF file that can be read into SLIDE for visualization. We use sharp Catmull-Clark subdivision scheme to obtain smooth surfaces for visualization.

1 Introduction

Knot theory provides the mathematical foundation for the study of knots. A knot is defined as a closed non-self-intersecting curve that is embedded in three dimensions and cannot be untangled to produce a simple loop [3]. Fig. 1(a) shows a simple trefoil knot. A knot can also be represented as a braid. Knot theory is used in the study of string theory, DNA replication, and statistical mechanics. A knot can be represented in 2D by a line diagram with the crossings as shown in Fig. 2(a). Another kind of representation is the braid representation as shown in Fig. 2(b).

Knot invariants like the minimum number of crossings in a diagram of a knot and genus are defined for comparing two knots to determine if they are topologically distinct knot. A seifert surface is a connected, oriented compact surface that has a knot as its boundary The genus of the minimal genus seifert surface bounded by a knot is the genus of the knot [2]. Instead of forming an oriented surface bounded by a knot, a possibly non-orientable surface can also be constructed as shown in Figure 1(b). In this project, we developed a tool to visualize knot spanning surfaces as well as a seifert surface for a user-defined knot. The two different surfaces for mode 1 and mode 2 for a simple trefoil knot are shown in Figure 1 (b) and (c). We use SLIDE for visualization.

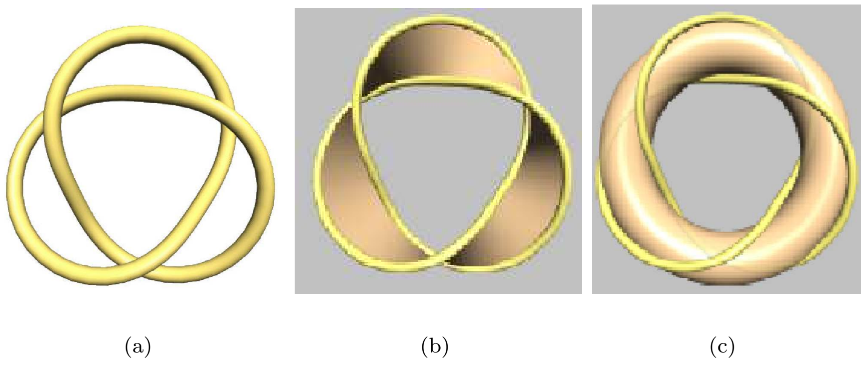


Figure 1: (a) Simple trefoil knot; (b) Non-orientable knot-spanning surface; (b) Orientable knot-spanning surface.

2 Related Work

There are only a few tools available for drawing and visualizing knots. A two dimensional sketching tool called Knot-Sketcher by Livinston et al. lets the user draw two dimensional diagrams representing knots [1]. Knot-Skethcer is easy to use but it does not visualize the knot in three dimensions, Wijk and Cohen developed a three dimensional seifert view visualizer [2]. This tool uses the braid representation that is not easy to come up with for complicated knots. Also, this tool does not visuzlize non-orientable knot-spanning surfaces.

3 Knot-Spanning Surface

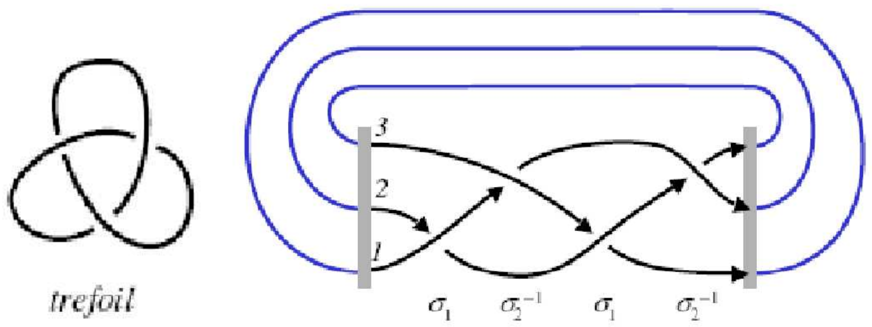
Our tool follows the following steps to construct and visualize knot-spanning surfaces:

1. Let the user draw the knot in two-dimensions.
2. Identify regions covered by knot-spanning surfaces, having odd winding numbers.
3. Construct mesh for the knot-spanning surface.
4. Use a subdivision scheme to smoothen the surface.

The following sections describe the above steps in detail.

3.1 Knot Sketcher

We developed a simple two dimensional sketching tool to draw knots interactively. The tool screen-shot is shown in Figure 3. The user can add points



(a) (b)

Figure 2: Two types of representation for a knot. (a) Sketch; (b) Braid.

interactively by clicking on the scene. The sketcher computes the crossings on the fly and builds a data structure that is useful later to write SLF files for visualization. The under and over crossings are shown in Figure 3. Currently the user is restricted to put the last point in such that the segment closing the contour does not create any new crossings. Also, it is recommended that every new line segment that user adds should only cross one previous segment due to a known bug that may miss some crossings if the new segment crosses more than one existing segments.

3.2 Calculating Winding Numbers

We identify regions with odd winding numbers before mesh construction. While winding number defines the number of revolutions around a point when marching along the closed contour(s), winding rules categorize which regions are inside or outside. The GLU library provides odd, nonzero, positive, negative, and absolute-value-greater-or-equal-to-two winding rules. In this project, we use odd winding rule. Winding numbers are same for all points in a connected region, which does not cross any boundary of the contour(s). Given one or more closed contours, winding numbers for each connected region can be calculated by the GLU library. Regions having winding numbers specified by the pre-defined winding rule are output as a set of closed counterclockwise oriented contours. For the simplicity of implementation, in the project, we only consider one contour for the knot.

3.3 Mesh Construction

A single knot encloses regions with odd winding numbers and crosses itself at the intersection of these regions (see Fig. 4(a)). For each such region and each

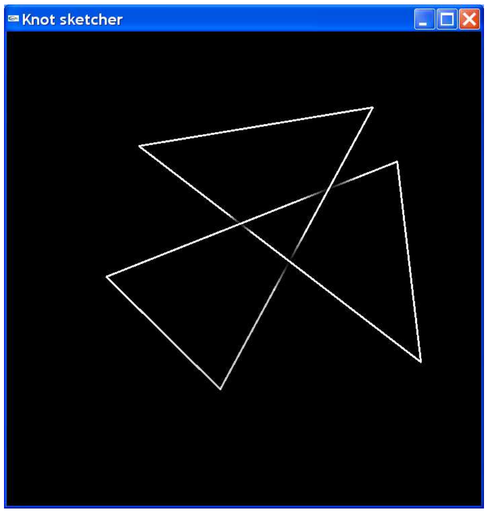


Figure 3: Screenshot of our knot-sketcher

crossing, we construct a set of mesh facets. Since each crossing is an intersection point, it is not suitable for mesh construction and rendering. Therefore, we insert four points on the knot near each crossing (see Fig. 4(b)). The mesh facets are connected to form the final mesh. How to construct mesh facets for mode 1 and mode 2 knot-spanning surfaces is shown in Table 1 and Fig. 5.

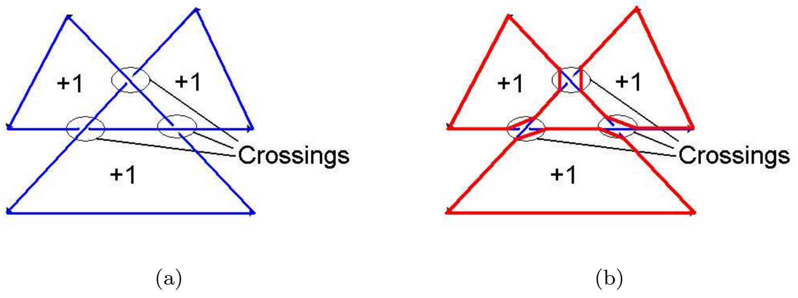
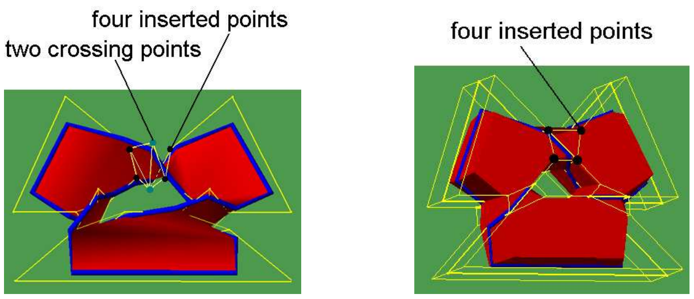


Figure 4: (a) A knot with regions having odd winding numbers and its crossings (all in blue); (b) Regions with odd winding numbers are now bounded by red lines after four points are inserted on the knot for each crossing; blue lines are for crossings.

Mode 1 surface is either single-sided or double-sided with an open boundary. The knot path coincides with the open boundary. When the surface is single-sided and the mesh facets go around one cycle from one side, they will meet the other side of the same facets, which makes at least one mesh edge having two

Table 1: Mesh construction for mode 1 and mode 2 knot-spanning surfaces

	mode 1	mode 2
Each region with odd winding number	One mesh facet on x-y plane	One mesh facet with positive z coordinates; one mesh facet with negative z coordinates; two vertical rectangular mesh facets for each edge except at intersections with crossings
Each crossing	Four triangular mesh facets with normal changing 180° gradually	Two triangular facets with positive z coordinates, separated by the knot; two triangular facets with negative z coordinates, separated by the knot; two vertical rectangular facets for each of the two edges having normal pointing outside the solid



(a) (b)

Figure 5: Mesh construction. (a) mode 1; (b) mode 2 knot-spanning surface.

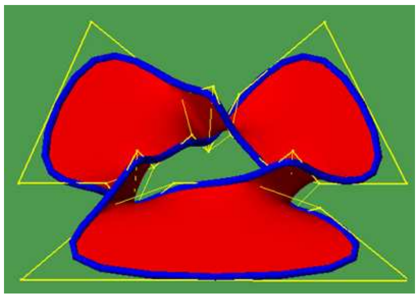
edge uses in the same direction. Since this is not allowed in SLIDE, we replicate all the vertices and construct two sets of mesh facets. These two sets of mesh facets are connected with each other at the edges having two edge uses in the same direction. That is, we construct two mesh facets with the same geometry but opposite normal directions. Table 1 only shows one set of mesh facets.

Mode 2 surface bounds a solid. Therefore, the surface is always double-sided. The knot path goes around the surface tightly. When coinciding with the knot path, the mesh edge is tagged with a sharpness of infinity, for the reason that will explained in the next section.

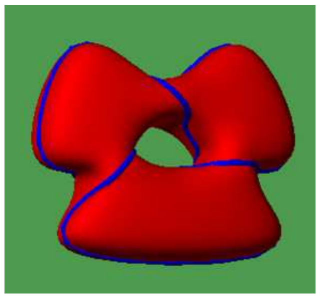
In SLIDE, subdivision accepts concave mesh facets but will give wrong results when a new vertex is inserted at the centroid of a mesh facet and the tessellation goes beyond the facet. We could have tessellated each region with odd winding numbers into convex polygons before the mesh is constructed. But due to time limitation, we assume that the tessellation using the centroid are always correct.

3.4 Surface Subdivision

After the mesh is constructed, we run a subdivision scheme on both the mesh and the knot path so as to achieve smoothness. The knot path is subdivided into a cubic b-spline. To make the b-spline bound tightly the subdivided mesh surface, we choose sharp Catmull-Clark subdivision scheme since it is based on bi-cubic tensor product b-spline. Besides, sharp Catmull-Clark takes any type of convex mesh facets, including non-triangular facets and non-rectangular facets, which makes the mesh construction easier without having to tessellate every polygon into triangular or rectangular mesh facets. Furthermore, we can tag some of the mesh edges with certain sharpness. With a sharpness n, sharp subdivision rules are used for n subdivision steps to calculate new vertices near the tagged edge; afterwards, the general subdivision rules, i.e., rules used in Catmull-Clark subdivision scheme, are used. Edges that are tagged with a sharpness 0 or that are not tagged with any sharpness are subdivided in the same way as for Catmull-Clark subdivision scheme. By tagging the mesh edges that lie on the knot path with a sharpness of infinity, these mesh edges when connected together can be approximated by a cubic b-spline, which coincides with the approximated knot and makes the knot bound the surface tightly. The subdivided surfaces for meshes shown in Fig. 5 are shown in Fig. 6. An alternative way instead of tagging the mesh edges with sharpness is to duplicate these mesh edges and make an open boundary along them so that the solid surface can be split apart along these mesh edges without tearing the surface except at the open boundary. But the mesh construction is more complex than simply tagging them with a sharpness of infinity.



(a)



(b)

Figure 6: Subdivided surface at subdivision level 3. (a) mode 1; (b) mode 2 knot-spanning surface.

4 Results and Discussions

In this project, we implement a knot sketcher with knots projected onto the x-y plane, and a knot-spanning surface generator for visualizing knots in 3D. The output of the knot sketcher is imported into SLIDE for knot-spanning surface visualization. We implemented two styles of surfaces (mode 1 and mode 2), which enables us to visualize both single-sided and double-sided surfaces. More examples of the knot-spanning surface generator are shown in Fig. 7.

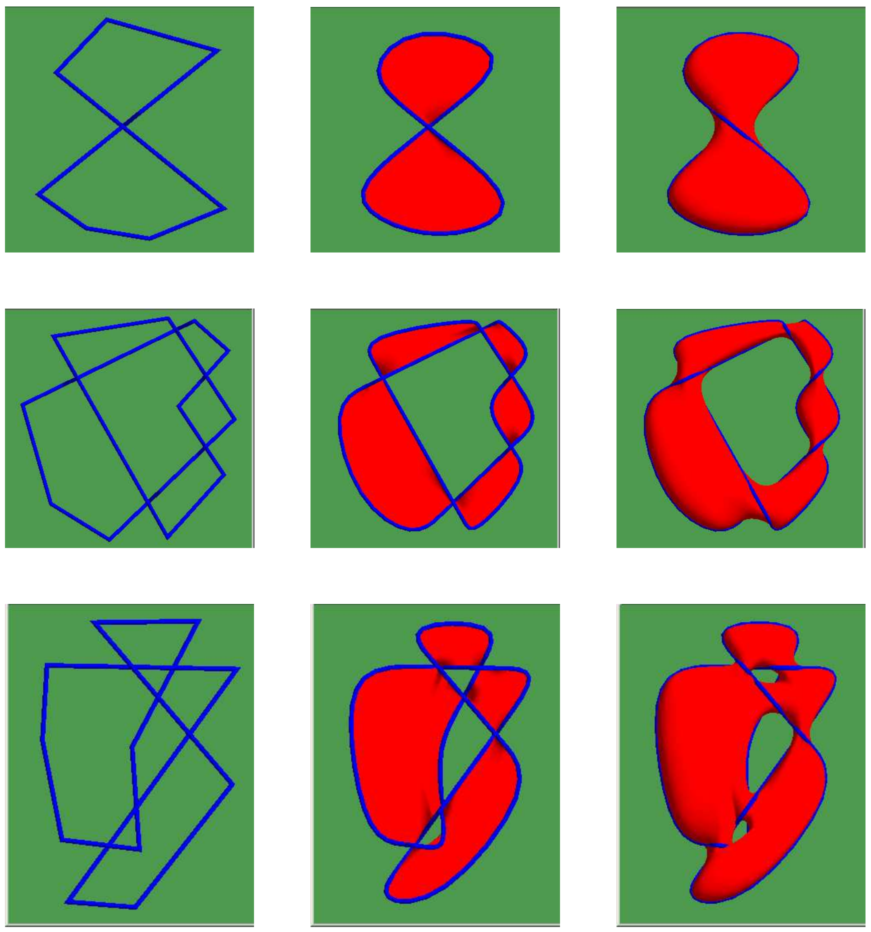


Figure 7: Examples, subdivided surfaces having subdivision level 3; column 1: knot; column 2: mode 1; column 3: mode 2 knot-spanning surfaces.

Currently, we use simple mesh to construct the knot-spanning surfaces. For mode 1 surfaces, only four triangular mesh facets are created for each crossing. The shape of the twisted surfaces connecting regions with odd winding numbers may be smoother if we add more mesh facets. For mode 2 surfaces, we can also increase the number of mesh facets to achieve greater smoothness. Another possible optimization is to offset those mesh edges lying on the x-y plane outward and insert a farther (relative to the x-y plane) point above or below the center of each region having odd winding numbers to make the solid shape rounder.

Where to insert the four points on the knot near each crossing may become a variable. Different locations can be tried to choose an optimal one for the best smoothness of the subdivided surface. Surface evolver may be another direction to go instead of subdivision.

Some other limitations of the current implementations are:

- Only one contour is considered for the knot.
- The centroid of each region having an odd winding number is assumed to be inside the region and the tessellation using this centroid do not go beyond the boundary of the original region.

These limitations can be easily overcome however, if time permits.

5 Conclusions

The goal of our project is to visualize all kinds of knots on their spanning surfaces. The shapes of the subdivided knots are close to the original user input, which makes the knot-spanning surfaces easier to understand even for non-experts. The visualization software, SLIDE, provides easy control on variables. Users can choose the optimal shape of the knot-spanning surface by simply changing the values for each variable.

References

[1] http://www.indiana.edu/%7Eknotinfo/homelinks/knotsketcher.html.

[2] Jarke J. van Wijk and Arjeh M. Cohen. Visualization of the Genus of Knots. In C. Silva, E. Gröller, and H. Rushmeier, editors, *Proceedings IEEE Visualization 2005*. IEEE CS Press.

[3] Mathworld website. Knot thoery article. *mathworld.wolfram.com*.