



Visualization of Seifert surfaces

Jarke J. van Wijk

Arjeh M. Cohen

Originally published in IEEE Transactions on Visualization and Computer Graphics, Vol. 12, No. 4, 485-496, July-Aug. 2006. Republication with permission from the IEEE.

Abstract. The genus of a knot or link can be defined via Seifert surfaces. A Seifert surface of a knot or link is an oriented surface whose boundary coincides with that knot or link. Schematic images of these surfaces are shown in every text book on knot theory, but from these it is hard to understand their shape and structure. In this article the visualization of such surfaces is discussed. A method is presented to produce different styles of surface for knots and links, starting from the so-called braid representation. Application of Seifert's algorithm leads to depictions that show the structure of the knot and the surface, while successive relaxation via a physically based model gives shapes that are natural and resemble the familiar representations of knots. Also, we present how to generate closed oriented surfaces in which the knot is embedded, such that the knot subdivides the surface into two parts. These closed surfaces provide a direct visualization of the genus of a knot. All methods have been integrated in a freely available tool, called SeifertView, which can be used for educational and presentation purposes.

Contents

I	Introduction	219
II	Background	220
III	Closed surfaces	225
IV	Structure	226
V	Geometry	228
VI	Results	239
VII	Discussion	244
	Bibliography	244

I Introduction

To introduce the topic discussed in this article, we start with a puzzle. Consider a trefoil, the simplest knot (Fig. 1). It is easy to define a surface that has this knot as its boundary: Take a strip, twist it three times, and glue the ends together. If we try to color the sides of the surface differently, we see that there is something strange. The strip is a kind of Möbius strip, and cannot be oriented, because there is only one side. The puzzle now is to define an orientable surface that has the trefoil as its boundary.

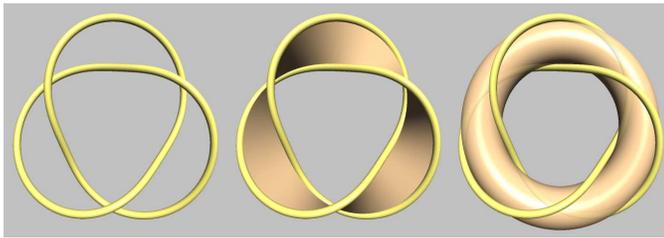


Figure 1: Trefoil

A second puzzle. It is easy to embed a trefoil in a closed surface: A trefoil is a so-called torus knot. However, this knot does not divide the torus into two parts, contrary to what one might expect from local inspection. Can we embed the knot on a closed surface, in such a way that it divides this surface into two parts?

The first puzzle has been solved in 1930 by Frankl and Pontrjagin [7], who showed that such a surface can be found for any knot.

Oriented surfaces whose boundaries are a knot K are called Seifert surfaces of K , after Herbert Seifert, who gave an algorithm to construct such a surface from a diagram describing the knot in 1934 [13]. His algorithm is easy to understand, but this does not hold for the geometric shape of the resulting surfaces. Texts on knot theory only contain schematic drawings, from which it is hard to capture what is going on. In the cited paper, Seifert also introduced the notion of the genus of a knot as the minimal genus of a Seifert surface. The present article is dedicated to the visualization of Seifert surfaces, as well as the direct visualization of the genus of knots.

This article is an extended version of a paper presented at IEEE Visualization 2005 [15]. The most important extensions are the inclusion of Scharein's method to produce smooth and natural knots and links, and the work we have done on dissemination of the results described here.

In section II we give a short overview of concepts from topology and knot theory. In section III we give a solution for the second puzzle: We

show how a closed surface can be constructed in which a knot is embedded, in such a way that it divides the surface in two parts. Whereas a Seifert surface consists of disks and bands, such a closed surface consists of spheres and tubes. In section IV we discuss how these elements can be derived and positioned from an abstract notation of a knot. In section V we show how surfaces can be generated. Results are shown in section VI, in the form of images of well-known knots and links. Also, SeifertView, our freely available tool which can be used to generate and view knots and Seifert surfaces, is described. Finally, in section VII the results are discussed and suggestions for future work are made.

II Background

In this section we informally introduce a number of definitions and concepts from topology in general and knot theory in particular. We limit ourselves to those results that are directly relevant for the work presented here. More information can be found in several books, and also on the Web many resources are available. *The Knot Book* [1] of Colin Adams gives a very readable and accessible introduction for non-experts; more depth can be found in [11, 8, 9].

Knot theorists have enumerated knots by means of diagrams or braid words, with invariants like the genus for distinguishing them. Results can be found in the literature and on the Web. The Knot Atlas of Bar-Natan provides many tables of knots and invariants [2]; the KnotInfo table [10] of Livingston and Cha was a very valuable resource for us.

A. Topology

Knot theory is a subfield of topology. Topology is the mathematical study of the properties of objects that are preserved through deformations of objects. Two surfaces are *homotopic* if each of them can be continuously deformed into the other. If this can be done without passing the object through itself, they are not only homotopic but also *isotopic*. For instance, a torus is isotopic (and hence also homotopic) to a cup with one handle, and homotopic (but not isotopic) to a tube in the shape of a trefoil.

Two surfaces are homotopic when three conditions are satisfied. First of all, either both should be orientable or neither; secondly, the number of boundary components must be the same; and finally, the Euler characteristic χ must be the same. The *Euler characteristic* χ is an invariant for surfaces. Given an arbitrary (but regular) polygonalization of a surface, $\chi = V - E + F$, with V the number of vertices, E the number of edges, and F the number of faces. Closed oriented surfaces are homotopic to a sphere with g handles (or, equivalently, a donut with g

holes). The number g is called the genus of the surface. For surfaces with m boundaries $\chi = 2 - 2g - m$. The genus of a surface with boundaries is defined to be equal to that of the surface that results when all components of the boundaries are capped off with (topological) disks.

B. Knot theory

Knot theory studies the properties of mathematical knots and links. A mathematical knot is a tamely embedded closed curve embedded in \mathbb{R}^3 . Here an embedding of a closed curve is called tame if it can be extended to an embedding of a tube (of fixed diameter) around the curve. A link consists of multiple components, each of which is a knot. A knot or link can be continuously deformed as long as it does not intersect itself. The result of such a deformation is a knot isotopic to the original one. Up to isotopy, a knot can be represented by a non-intersecting closed polyline (finite sequence of line segments in three-space).

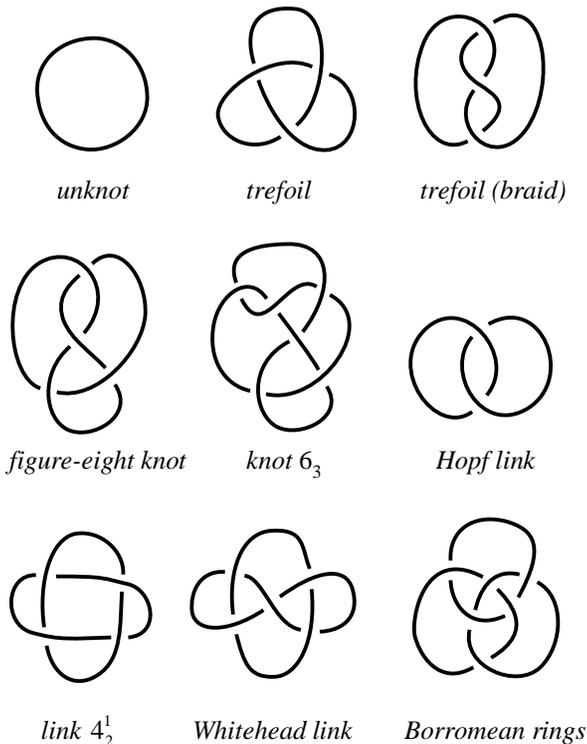


Figure 2: Knot and link diagrams

Knots and links are usually studied using *projections* or *diagrams*, such as shown in Fig. 2. One knot can be projected in many different ways; as an example two different projections of the trefoil are shown. A diagram consists of edges and crossings. If an orientation is assigned to the knot, we see that two different types of crossings exist: right-hand crossings and left-hand crossings (Fig. 3).

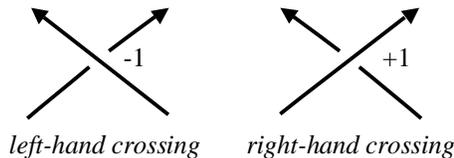


Figure 3: Two different types of crossings

Some important questions in knot theory are whether two knots are the same or not, and especially if a knot is equal to the unknot; how many different knots do exist (given constraints), and how to classify knots. One approach to this is to define invariants of knots. A classic one is the minimum number of crossings in a diagram of a knot; more powerful and distinctive are so-called knot polynomials, such as the Jones polynomial [1].

C. Seifert surfaces

The genus of a knot, introduced by Seifert [13], is another classic invariant in knot theory. The Euler characteristic for a 1-dimensional object is 0 when applied to a knot, hence that does not lead to a distinction. Seifert therefore used a connected, oriented, compact surface that has the knot as its boundary to define the genus of a knot. At first sight, it is surprising that such a surface exists for any knot or link. Seifert showed that such a surface can be derived from a knot diagram using a simple algorithm. It consists of four steps (Fig. 4). First of all, assign an orientation to the components of the knot or link. Secondly, eliminate all crossings. At each crossing two strands (say, A and B) meet. A crossing is eliminated by cutting the strands, and connecting the incoming strand of A with the outgoing strand of B , and vice versa. This gives a set of non-intersecting (topological) circles, called Seifert circles. Thirdly, if circles are nested in each other, offset them in a direction perpendicular to the diagram. Fill in the circles, giving disks. Finally, connect the disks using twisted bands. Each band corresponds to a crossing, and has one twist, with orientation derived from the crossing type. A twist is a rotation over plus (right-hand) or minus (left-hand) 180 degrees. Note that the crossing type does not influence the circles that are generated. The resulting surface satisfies the

requirements. Different projections of the knot lead to different surfaces, possibly also with a different genus. The genus of a knot is defined as the minimal genus of all oriented surfaces bounded by the knot. Note that not all surfaces bounded by a knot arise from Seifert’s algorithm, and there are examples with genus lower than that computed from the algorithm.

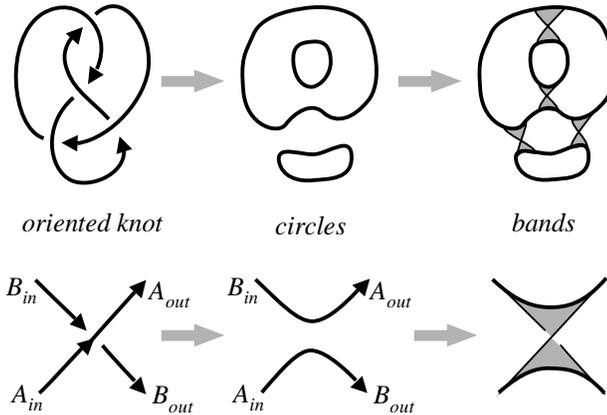


Figure 4: Seifert’s algorithm: Assign orientation, eliminate crossings, and add bands; shown for a knot and a crossing

D. Challenge

Texts on knot theory show figures similar to Fig. 4. From these it is hard to understand the shape of the surface. One reason is that such surfaces are not familiar and are rarely encountered in the real world. We have searched the literature and the Web, but could not find satisfying visualizations of Seifert surfaces. The KnotPlot package of Robert Scharein [12] has a very rich set of features and is a delight to work (and play) with, but even this has no option to show Seifert surfaces. We therefore found it a challenge to develop a method to visualize Seifert surfaces. Specifically, our aim was to enable the viewer to generate and view Seifert surfaces interactively in 3D for arbitrary knots and links in different styles.

One possible route is to consider a Seifert surface as a minimal surface (i.e., the surface with zero mean curvature, also known as the soap bubble surface) using the knot as its fixed boundary. However, this requires that a three-dimensional knot is available. Also, the definition of a suitable initial surface mesh and the iterative calculation of the minimal surface are not easy to implement and are compute intensive. We therefore opted for a different approach. Given an abstract notation of a knot, derive the structure of the Seifert surface and find a smooth geometry in a quick and deterministic way.

E. Braid representation

To generate Seifert surfaces for arbitrary knots and links, we need an encoding for these knots and links. Many different encodings have been developed, such as the Conway notation and the Dowker-Thistlethwaite notation. For our purposes we found the braid representation to be very useful. By means of braids, several different styles of surfaces can be generated easily; and also, the braid representation lends itself well to experimentation. It does have its limitations though, as we discuss in section V.

A braid consists of a set of n strings, running (here) from a left bar to a right bar (Fig. 5). Strings are allowed to cross, and the pattern can be encoded by enumerating the crossings from left to right. A crossing is denoted by σ_k^j , which means that strings at the k 'th and $k + 1$ 'th row are twisted j times, where $j = 1$ denotes a right-hand crossing and $j = -1$ a left-hand crossing cf. Fig. 3. The closure of the braid is defined by attaching the left bar to the right bar, such that no further crossings are introduced. In other words, we add n extra strings that connect the beginnings and ends of strings at the same row, without further crossings. Every knot and link can be defined as a braid. A trefoil has the *braid word* $\sigma_1\sigma_1\sigma_1 = \sigma_1^3$, a figure eight knot can be represented as $\sigma_1\sigma_2^{-1}\sigma_1\sigma_2^{-1}$. An alternative notation for braids is to use uppercase letters for right crossings and lowercase letters for left crossings, and where the character denotes the strings effected, according to alphabetic order. Hence, a trefoil is encoded by AAA , and a figure eight knot by $AbAb$. Furthermore, every possible braid word defines a knot or a link, which makes this representation well suited for experimentation.

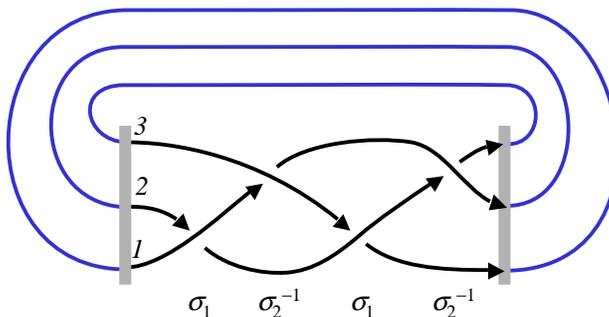


Figure 5: Braid representation of figure-eight knot

III Closed surfaces

Besides visualization of Seifert surfaces, another aim was to make the genus of a knot 'more visible'. A trefoil or a figure eight knot has genus 1, hence the corresponding Seifert surfaces are homotopic to a torus with a hole in the surface. Via a number of steps in which the Seifert surface is deformed, cut, and glued, this equivalence can be shown, but it is not really intuitive. Closed surfaces are easier to understand, hence we studied how a closed surface can be generated that contains the Seifert surface as an embedded subsurface. We call such a surface a closed Seifert surface. The following reasoning is straightforward, but we could not find it in the literature.

The standard approach of topologists is to cap off boundaries (here the m boundaries of the Seifert surface) with (topological) disks. This leads to a surface that is homotopic to a closed surface, but not isotopic. What we need here to close the surface in a more decent way, is an oriented surface that has the m components of the link as boundary. But this is exactly the definition of a Seifert surface itself, which leads us immediately to a solution. Using a physical analogy, the solution is to take two identical Seifert surfaces, glue them together at the boundaries, and inflate the closed object. This is shown in Fig. 6 for a trefoil (which also shows a possible solution to the puzzles posed in the introduction). The Seifert surface consists here of two disks, connected by three bands; the closed Seifert surface consists of two spheres, connected by three tubes. The knot splits the closed surface into two parts.

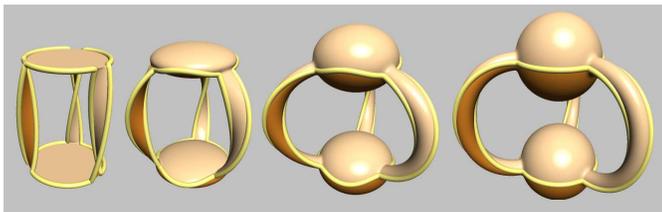


Figure 6: Inflating two Seifert surfaces, glued together at their boundaries

The genus of a closed Seifert surface can be determined as follows. The Euler characteristic of a Seifert surface is $\chi_s = 2 - 2g_s - m$, with g_s the genus and m the number of components of the knot. For the Euler characteristic χ_c of the closed surface we find $\chi_c = 2\chi_s$: The number of vertices, faces and edges doubles, but at the boundaries a certain number of edges and the same number of vertices disappear. However, as V and E have opposite signs in the definition of χ , this does not influence the resulting value. For a closed surface $\chi_c = 2 - 2g_c$, with g_c the genus of the closed surface. This leads to

$$g_c = 2g_s + m - 1.$$

This gives us a direct way of finding oriented closed surfaces in which to embed a knot or link of genus g_s such that the knot divides it into two parts. For instance, for a trefoil or figure eight knot a genus 2 surface can be used (such as a donut with two holes, or two spheres connected by three tubes), and in greater generality, for a knot of one component a donut with $2g_s$ holes can be used.

IV Structure

In this section we derive the structure of the Seifert surfaces, starting from the braid word. The aim here is to determine the number of disks (or spheres) and their position in space, and the bands (or tubes), with the number of twists and attachment positions to the disks as attributes. The disks are positioned in 3D (x, y, z) space. We take x and y in the plane of the diagram, and z perpendicular to the plane. Disks are parallel to the x, y plane. Each disk has two sides, denoted A and B . For each disk a decision must be made if the A or B side is positioned upwards.

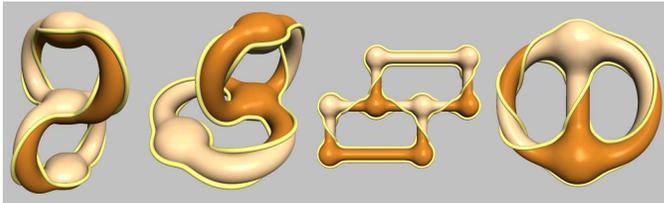


Figure 7: Figure eight knot in stacked, split, flat, and reduced style

Because of the regular structure of braids, various styles of Seifert surfaces can easily be derived from these. Fig. 7 shows four styles for a figure eight knot, using ellipsoids and tubes. First, the *stacked* style. If all closing strings are positioned in the default way, it is easy to see that the Seifert circles are all nested. Hence, the corresponding Seifert surface consists of a stack of disks, where each disk is connected with bands to its neighbors (Fig. 8). All disks have the A side facing upwards, their position is $(0, 0, (i - 1)D)$, where i is the index of the row to which the disk corresponds, and D a distance between the disks. A nice geometric representation is obtained by subdividing each disk into k sectors, where k is the total number of crossings. Sectors of neighboring disks are connected with bands when appropriate. Using a suitable setting for the geometry, we generate an object similar to a wedding-cake.

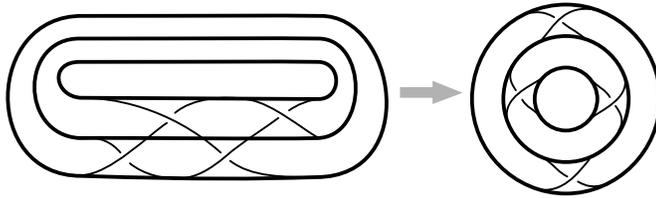


Figure 8: Standard braid representation gives stacked disks

As a variation, one set of closing strings can be positioned above, and the remaining set can be positioned below the braid. This gives the *split* style: two sets of stacked disks in wedding-cake style, where the lower disks of each set are connected by bands in the plane. One set has the *A* side facing up, the other set has the *B* side upwards. As an example, in Fig. 9 two strings are positioned above and one is positioned below the braid. We introduced this style in order to produce for instance the Seifert surface that results from the standard projection of the figure eight knot.

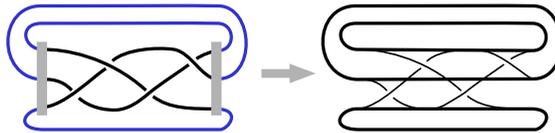


Figure 9: Split style

An alternative style, the *flat* style, is obtained as follows. The upper closing string is positioned above of the braid, the lowest closing string below the braid, and the closing strings in between are put downward, pushed perpendicular to the plane of the braid. Strings of the last kind introduce extra crossings. Their number can be minimized by carefully choosing the path of the string (Fig. 10). From this lay-out of the strings, disks and bands can be derived using Seifert’s algorithm. Thus a set of non-nested, disjoint Seifert circles will be obtained, so they can be positioned in a plane. The structure can be constructed as follows. Suppose that σ_k^j is the i -th crossing. We add two disks, one with *A* up (brown) at position $(iD, kD, 0)$ and one with *B* up (yellow) at $(iD, (k + 1)D, 0)$. In other words, at each upper and lower triangle of an original crossing disks are positioned. Next, vertical bands are added that represent the original crossings, with a twist according to the crossing. Finally, horizontal bands are added between disks on the same row. If both disks have the same side up, no twists are added. An *A* up disk on the left and a *B* up disk on the right are connected by a band with a single negative twist, and a single positive twist is used for the reverse order.

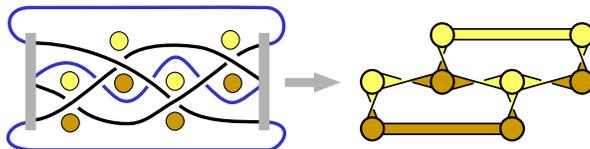


Figure 10: Flat style

The flat style is not particularly interesting, but this planar lay-out can be simplified further, giving the more attractive *reduced* style. Several disks have only two bands attached to them. Such a disk can be removed, and the original two bands can be replaced by a single band, with the number of twists equal to the sum of the number of twists of the original bands. Application of this rule to the figure eight knot leads to a simple structure of two disks, connected by three bands with 1, 1 and -3 twists respectively (Fig. 11). Such a knot, with a Seifert surface that consists of two disks, connected by parallel twisted bands, is known as a pretzel knot. The trefoil is a $(1, 1, 1)$ pretzel knot.

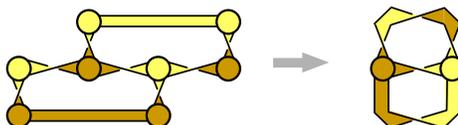


Figure 11: Reduced style

The structure of the pattern of disks and bands can be described as a planar graph, with each disk mapped to a vertex, each band to an edge, and each hole to a face. For the optimal lay-out of such graphs a number of algorithms exist [5]. We implemented a simplistic one (using a trial-and-error approach), which gave satisfactory results for the graphs produced here.

V Geometry

In the previous section we have discussed how to generate disks and bands from a braid word, and how to position and orient the disks. The next step is to produce a surface to visualize the Seifert surface or the corresponding closed surface. We use ellipsoids as the basic shape for disks and spheres, and curved cylinders with an elliptical cross section for the bands and tubes. These are approximated with polygons. Smoothing can be applied to obtain smoother knots and surfaces. Furthermore, we describe two extensions of the basic method: definition of multiple vertical twists and of double knots.

A. Ellipsoids

In the standard position, an ellipsoid with two axes of equal length (representing a squeezed sphere) can be described by

$$\mathbf{p}(\mathbf{u}) = (d \cos u \cos v, d \sin u \cos v, h \sin v)/2,$$

with spherical coordinates $\mathbf{u} = (u, v) \in [-\pi, \pi) \times [-\pi/2, \pi/2]$, and with the diameter d and height h as parameters. Obviously, setting h close to zero gives a disk, setting $d = h$ gives a sphere. The ellipsoid is subdivided into n_s sectors, where each sector has at most one tube attached. Consider one such a sector $(u, v) \in [-U, U] \times [-V, V]$, where $U = \pi/n_s$ and $V = \pi/2$. The top half ($v \in (0, V]$) belongs to either A or B , the bottom part belongs to the other part of the surface. If no band is attached, then this sector can be straightforwardly polygonized with a rectangular mesh with size parameters I and J . The vertices are $\mathbf{p}_{ij} = \mathbf{p}(\mathbf{u}_R(i, j))$, with

$$\mathbf{u}_R(i, j) = (Ui/I, Vj/J)$$

and $(i, j) \in [-I, I] \times [-J, J]$. Obviously, the vertices at the poles coincide.

If a band or tube is attached, a hole must be made in this mesh, and some care is required to make sure that this hole conforms with the end of the tube. The cross-section of bands and tubes is described as an ellipse, with width w and height d . Obviously, setting w close to 0 gives a band, setting $w = d$ gives a tube. Suppose that the attachment point of the centerline of the tube is $\mathbf{p}_A = \mathbf{p}(\mathbf{u}_A)$. Typically, $u_A = 0$, and v_A is an optional offset in the direction of the poles to move the attachment point closer to the disk to which the other side of the tube points. This was used for instance in Fig. 6. We model the boundary of the hole in the ellipsoid in spherical coordinates as

$$\mathbf{u}_B(s) = \mathbf{u}_A + (a \cos s\pi/2, b \sin s\pi/2),$$

with $s \in [0, 4)$ (Fig. A.). The lengths of the semi-axes a and b are chosen so as to match the distances of $\mathbf{p}(\mathbf{u}_B(0))$ and $\mathbf{p}(\mathbf{u}_B(1))$ to $\mathbf{p}(\mathbf{u}_A)$, measured along the surface of the ellipsoid, with $w/2$ and $d/2$, respectively. This hole is a perfect ellipse in (u, v) space, and, for our purposes, a good enough approximation of an ellipse in 3D space.

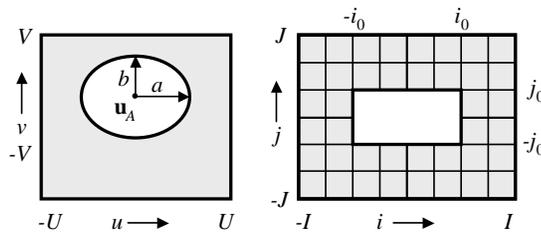
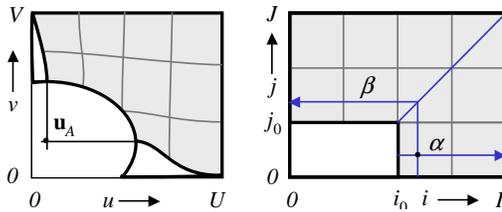


Figure 12: Sector of ellipsoid in (u, v) and (i, j) coordinates

Figure 13: Upper right quadrant in (u, v) and (i, j) coordinates

Also, we define a rectangular hole in the mesh space: $(-i_0, i_0) \times (-j_0, j_0)$. The mesh has to be warped such that the inner boundary conforms with the hole in the ellipsoid, while the outer boundary still conforms with the standard boundary of the sector. We have modeled this as follows. Consider the upper-right quadrant of the sector (Fig. 13). We measure the position of a mesh-point (i, j) in a kind of polar coordinates (α, β) , where $\beta \in [0, 1]$ denotes the angle, and $\alpha \in [0, 1]$ how close we are to the inner boundary ($\alpha = 0$) or the outer boundary ($\alpha = 1$). Specifically, we use

$$\alpha_{ij} = \max(\alpha_i, \alpha_j) \quad \text{with}$$

$$\alpha_i = \frac{i - i_0}{I - i_0} \quad \text{and} \quad \alpha_j = \frac{j - j_0}{J - j_0};$$

and

$$\beta_{i,j} = \begin{cases} j/L(i, j) & \text{if } \alpha_i > \alpha_j \\ 1 - i/L(i, j) & \text{otherwise} \end{cases}$$

with

$$L(i, j) = (1 - \alpha_{ij})(i_0 + j_0) + \alpha_{ij}(I + J).$$

If only the hole has to be taken care of, mesh points can be found using

$$\mathbf{u}_C(i, j) = \mathbf{u}_A + \left((\alpha_{ij}a + (1 - \alpha_{ij})(U - u_a)) \cos \beta_{ij}\pi/2, \right. \\ \left. (\alpha_{ij}b + (1 - \alpha_{ij})(V - v_a)) \sin \beta_{ij}\pi/2 \right).$$

To obtain a smooth transition from the inner to the outer boundary, we determine the vertices $\mathbf{p}_{ij} = \mathbf{p}(\mathbf{u}_H)$ by blending circular and rectangular coordinates via

$$\mathbf{u}_H(i, j) = (1 - h(\alpha_{ij}))\mathbf{u}_C(i, j) + h(\alpha_{ij})\mathbf{u}_R(i, j) \quad \text{with}$$

$$h(t) = -2t^3 + 3t^2.$$

The blending function $h(t)$ gives a smooth transition at the boundaries because $h'(0) = 0$ and $h'(1) = 0$. The other quadrants are dealt with similarly. A result is shown in Fig. 14.

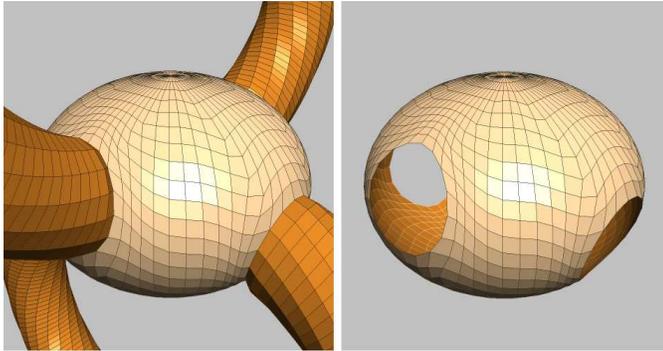


Figure 14: Mesh of ellipsoid

B. Tubes

The tubes are also modeled via a rectangular mesh of polygons. We use a mesh $\mathbf{c}_{ij}, i \in [0..P - 1], j \in [0..Q]$, where i runs around the cross section of the tube, and j along the centerline. The centerline of a tube is modeled by use of a cubic Bézier curve [6]. Such a curve is given by

$$\mathbf{b}(t) = (1 - t)^3\mathbf{b}_0 + 3(1 - t)^2t\mathbf{b}_1 + 3(1 - t)t^2\mathbf{b}_2 + t^3\mathbf{b}_3$$

with $t \in [0, 1]$. For \mathbf{b}_0 and \mathbf{b}_3 we use the end points of the tube, i.e., the attachment points \mathbf{p}_A . The control point \mathbf{b}_1 is derived from the normal \mathbf{n}_0 on the surface of the ellipsoid

$$\mathbf{b}_1 = \mathbf{b}_0 + \mu\mathbf{n}_0/3|\mathbf{b}_3 - \mathbf{b}_0|$$

where μ (typically 1) can be tuned to vary the offset of the tubes. The other control point \mathbf{b}_2 is defined similarly.

To generate the surface of the tube, contours must be rotated and interpolated. We use a Frenet frame as a natural reference frame along the centerline, given by

$$\mathbf{f}_3(t) = \mathbf{b}'/|\mathbf{b}'|, \mathbf{f}_2(t) = \mathbf{f}'_3/|\mathbf{f}'_3|, \mathbf{f}_1(t) = \mathbf{f}_3 \times \mathbf{f}_2,$$

where $\mathbf{b}' = d\mathbf{b}/dt$. A Frenet frame is undefined when the curvature is zero. When the control points are colinear, an arbitrary frame can be chosen instead. When locally the curvature is zero, the frame can rotate over 180 degrees, which has to be checked and corrected for.

Suppose that the start contour consists of a sequence of points \mathbf{p}_i , with $i = 0, \dots, P - 1$, such that \mathbf{p}_0 is located at the boundary between the A and B part of the surface, and with a counterclockwise orientation when viewed from outside the ellipsoid. Here $P = 4i_0 + 4j_0$. The end contour

with points \mathbf{q}_i is defined similarly, also with P points, except that we assume here a clockwise orientation. We use a rotating frame for the rotation of the contour, given by

$$\begin{aligned}\mathbf{g}_1(t) &= \cos \phi \mathbf{f}_1 - \sin \phi \mathbf{f}_2 \\ \mathbf{g}_2(t) &= \sin \phi \mathbf{f}_1 + \cos \phi \mathbf{f}_2 \\ \mathbf{g}_3(t) &= \mathbf{f}_3\end{aligned}$$

with

$$\phi = \phi(t) = (\phi_1 - \phi_0 + T2\pi)t + \phi_0.$$

The offset ϕ_0 is set in such a way that initially \mathbf{g}_1 is aligned with $\mathbf{p}_0 - \mathbf{b}_0$. We measure this initial offset relative to the Frenet frame with

$$\phi_0 = \arctan \frac{\mathbf{p}_0^* \cdot \mathbf{f}_2(0)}{\mathbf{p}_0^* \cdot \mathbf{f}_1(0)}$$

where

$$\mathbf{p}_0^* = \mathbf{p}_0 - \mathbf{b}_0 - ((\mathbf{p}_0 - \mathbf{b}_0) \cdot \mathbf{f}_3(0))\mathbf{f}_3.$$

The final offset ϕ_1 is defined similarly. The value of T is chosen such that the total rotation $\phi(1) - \phi(0)$ matches with the desired number of twists R of the tube, e.g.,

$$T = \text{round} \frac{\phi_0 - \phi_1 + R\pi}{2\pi}.$$

Contours are interpolated in a local frame, using a cubic Bézier spline again, i.e.,

$$\mathbf{c}_i^*(t) = (1-t)^3 \mathbf{c}_{i0}^* + 3(1-t)^2 t \mathbf{c}_{i1}^* + 3(1-t)t^2 \mathbf{c}_{i2}^* + t^3 \mathbf{c}_{i3}^*.$$

For \mathbf{c}_{i0}^* we use start contour points, transformed by use of the $\mathbf{g}(0)$ frame:

$$\mathbf{c}_{i0}^* = (\mathbf{g}_1(0) \cdot (\mathbf{p}_i - \mathbf{b}_0), \mathbf{g}_2(0) \cdot (\mathbf{p}_i - \mathbf{b}_0), \mathbf{g}_3(0) \cdot (\mathbf{p}_i - \mathbf{b}_0)).$$

For \mathbf{c}_{i3}^* the end contour points are used:

$$\mathbf{c}_{i3}^* = (\mathbf{g}_1(1) \cdot (\mathbf{q}_i - \mathbf{b}_3), \mathbf{g}_2(1) \cdot (\mathbf{q}_i - \mathbf{b}_3), \mathbf{g}_3(1) \cdot (\mathbf{q}_i - \mathbf{b}_3)).$$

For the contours in between we use ellipses:

$$\mathbf{c}_{i1}^* = \mathbf{c}_{i2}^* = (w \cos 2\pi i/P, d \sin 2\pi i/P, 0).$$

The points of the mesh of the tube are now finally given by

$$\mathbf{c}_{ij} = \mathbf{b}(j/Q) + (\mathbf{g}_1(j/Q), \mathbf{g}_2(j/Q), \mathbf{g}_3(j/Q)) \cdot \mathbf{c}_i^*(j/Q).$$

C. Smoothing

The preceding approach gives ellipsoids and tubes. To obtain smoother surfaces, especially to render less abrupt transitions between tubes and ellipsoids, smoothing can be applied. In [15] we proposed to use geometric smoothing, based on Catmull-Clark subdivision [3]. This does indeed give more attractive shapes, but the resulting knots and links often still did not resemble their natural counterparts, shown in textbooks, or produced by Scharein's KnotPlot [12]. The latter immediately suggests a solution: Apply Scharein's method for smoothing the links here also, and let the surface follow. In the following we describe this procedure in more detail.

Scharein's approach is to use a relatively simple physics-based iterative procedure. Each vertex of a link is considered as a point mass, and is attracted by its neighbors and repelled by all other vertices of all links. The positions of the vertices are incrementally updated taking the forces into account, until a stable or attractive configuration results. In more detail, in his model for the magnitude of the attracting force F_a between two neighboring vertices

$$F_a(r) = Hr^{1+\beta}$$

is used, modeling a generalization of Hooke's law. The use of $\beta = 0$ gives the standard linear version. For the repelling force F_r between vertices a generalized electrostatic model is used, i.e.,

$$F_r(r) = Kr^{-(2+\alpha)},$$

where the use of $\alpha = 0$ gives the standard inverse quadratic version. We used a slight adaptation. Instead of $F_a(r)$ and $F_r(r)$ we use $F_a(r/r_a)$ and $F_r(r/r_a)$, where r_a is the initial average distance between neighboring vertices. This reduces the effect of the initial scale of the model on the final result.

For the calculation of the motion of the vertices Newton's laws and a simple Euler scheme are used. Each vertex has an associated velocity \mathbf{v} , which is updated for time step i according to

$$\mathbf{v}_{i+1} = (1 - \gamma)\mathbf{v}_i + \mathbf{F}\Delta t_i.$$

The amount of damping (and hence dissipation of energy) can be controlled via γ , and \mathbf{F} is the sum of all forces acting on a vertex of a link. The new position \mathbf{p}_{i+1} follows from

$$\mathbf{p}_{i+1} = \mathbf{p}_i + \min(d_{\max}, |\mathbf{v}_{i+1}\Delta t_i|) \frac{\mathbf{v}_{i+1}\Delta t_i}{|\mathbf{v}_{i+1}\Delta t_i|}.$$

For the vertices of the surfaces we used almost the same force model (including normalization by r_a), except that only attracting forces and no repelling forces are used. As a result, the surface follows the link, but

does not influence it. Using a physical analogy, the knot is modeled as a steel rod, and the surface as a thin flexible rubber sheet. This simple model for the surfaces does not lead to a minimal surface, but it does lead to smooth surfaces with faces of similar size and shape.

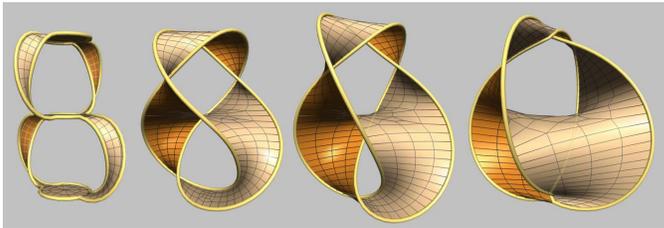


Figure 15: Smoothing using relaxation

The amount of displacement is clamped to a value d_{\max} . Furthermore, if the new position of the vertex is closer than $d_{\text{close}} (> d_{\max})$ from non-neighbouring edges, the update is ignored. Scharein has proved [12] that this combination of measures prevents self-intersection of the knot. Surfaces are not checked for self-intersection. Self-intersection can occur when the simulation is continued in search of a minimal energy, but often such a configuration is visually not attractive. Also, a check for self-intersecting surfaces would give a high performance penalty.

The time taken per time step is quadratic in the number of vertices of the link, and linear in the number of vertices of the surfaces. For a smooth interactive performance, a low number of vertices has to be used. Hence, we use by default low resolution settings for the meshes. For the mesh of the disks we use a scheme in which the number of meridians is constant between two tubes, and independent of the angle between the tubes.

Furthermore, selection of a proper time step Δt_i is important. Too high a value gives an unstable result, too low a value does not give enough progress. To prevent both extremes, we use an exponentially decreasing time step

$$\Delta t_{i+1} = (1 - \mu)\Delta t_i,$$

where μ denotes the strength of the decrease. As a result, initially large steps are made, whereas later on the shape stabilizes to a smooth shape. This is not necessarily the minimal energy configuration, but that one did not always seem to be the most attractive anyway. In our implementation, each time the user presses a *smooth* button, a new cycle of iterations is started: the time step is reset to an initial large value, and the model is smoothed further, which gives an easy control over the amount of relaxation desired. Each cycle takes typically 5-10 seconds, shown as a smooth animation on the screen. Fig. 15 shows the effect of this smoothing

procedure. On the left the original mesh is shown, followed by application of one, three, and a large number of cycles of iterations. Already after one cycle an attractive result is obtained. The last version is the minimal energy configuration for this parameter setting, where the collision check prevents further smoothing. The resulting shape is geometrically simple, but less attractive than its predecessors.

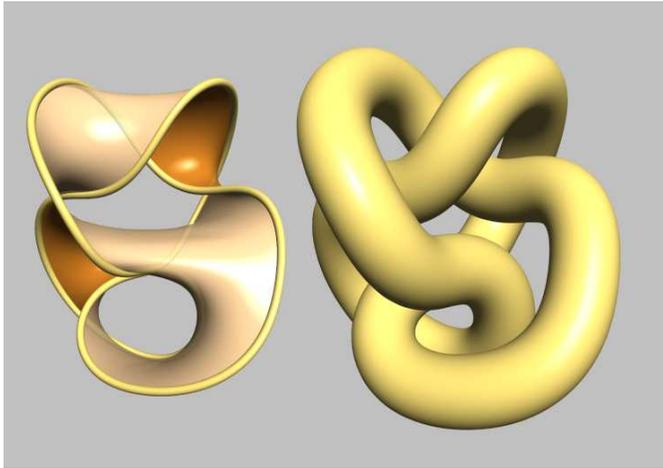


Figure 16: Use of a high value for α

We used here $\alpha = 0$ and $\beta = 1$, which we found to give nice results for stacked disks configurations. Scharein recommends to use a higher value for the repulsion coefficient α , such as $\alpha = 4$. This gives a result as shown in Fig. 16. A high α has the effect that the knot is surrounded by a hard tube, the force quickly increases when the knot is approached. This gives a more irregular knot and surface. However, when the aim is to show a knot represented by a thick tube in a small space, which is typical for KnotPlot, a high value of α is required.

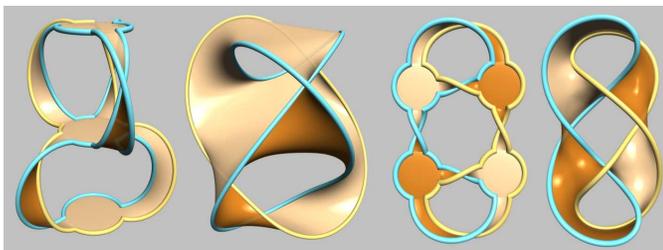


Figure 17: Different initial configurations for Whitehead link.

Different initial configurations lead to different results. An example using the Whitehead link is shown in Fig. 17. On the left, a stacked disks style is used, which gives a ring, around which another link is twisted in a figure eight way. On the right, a reduced style is used, which leads to two symmetric links. In this case, the simple relaxation scheme used will never lead to the same result, because the lengths of the links are different in the first and the same in the second case. Also, it seems as if there are several local minima. As the aim here is mainly to obtain a smooth and understandable result, rather than a global optimum, this is not a problem. Usually, the most pleasant results were obtained with the simple stacked disks style. This model is regular (all bands are similar) and leads to three-dimensional shapes, in contrast to the other styles. Smoothing therefore leads fluently to spatial surfaces.

If one would aim at a quick and useful implementation of visualization of Seifert surfaces, our recommendation is to start with the stacked disks style in combination with relaxation as described before. This combination is relatively easy to implement, leads to results that show the structure of the surface, and yields smooth surfaces bounded by natural representations of the knot.

For presentation purposes, higher resolution meshes are convenient, and we therefore kept an option for geometric refinement. Upon user request, the links are refined by means of an interpolation scheme following the Catmull-Rom spline [4]; for the surfaces Catmull-Clark subdivision [3] is used.

D. Knot representation

It is convenient to have an explicit representation of the knot or the components of the link that correspond to the surfaces. For this purpose, the geometry of the knot is derived from the surfaces. Each polygon is assigned to part A or B of the surface, components are found by tracing edges that bound polygons that belong to different parts. The knot is shown as a tube. Optionally, an offset can be specified, such that the knot is shifted perpendicular to the surface in an outward normal direction. In Fig. 7 we used an offset of the radius of the tube, such that the knot touches the surface. Also, this is useful for visualizing the linking number of the offset with the original knot, a quantity that plays a role in knot invariants like the Alexander polynomial.

We have added an option to use transparency for more insight in the resulting shape. Transparency itself is not without problems using the Z-buffer algorithm employed in graphics cards. For an optimal result with transparent surfaces, all polygons should be sorted and rendered in back to front order, which is a time consuming operation. We use a shorter route. For insight into the structure, understanding the shape of the knot is vital,

hence it is advantageous to see the knot through surfaces. We implemented this idea by rendering first all surfaces, followed by rendering the knot transparently but only when behind the surfaces, and finally rendering the knot again opaquely when the knot is in front of the surfaces.

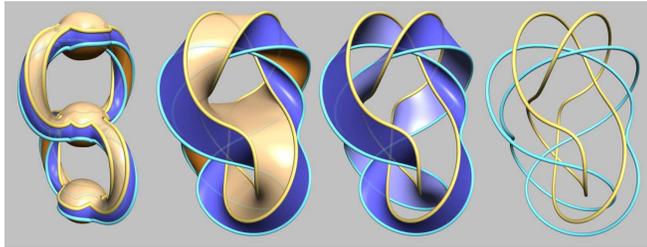


Figure 18: Double figure eight knot

E. Extensions

In the approach so far, a knot or link separates two surfaces (say A and B). We can split the knot into two parallel knots and introduce a new surface C in between them. We implemented this as follows. The algorithms produce a mesh where each face is labelled A or B . If we now change these labels to C for all faces that meet a face with a different label, we obtain a strip of two faces wide that is labeled C , assuming that the knot is bounded by at least two faces on each side with the same label. Repeated application of this step gives a wider strip labeled C . Next, if the standard tracing method for finding links in space is used, a parallel knot emerges. This extension was easy to implement, but the results are complex, as shown in Fig. 18, where for a figure eight knot a stacked balls version, and various views of a smoothed version are shown. The blue surface C is a ribbon in the shape of a figure eight knot.

Another extension is related to a limitation of the braid representation: It does not always yield a minimal genus surface. Consider Fig. 19, where knot 5_1 , also known as the cinquefoil knot, and the almost similar 6_1 knot are compared. The knot 5_1 has the braid word $AAAAA$, the knot 6_1 has the braid word $AABacBc$. If we use these braid words to generate Seifert surfaces, we find a good result for the cinquefoil knot. The closed Seifert surface has four holes, which matches with its genus 2. However, the surface for the 6_1 knot also has four holes. The 6_1 has genus 1, and to visualize this, the shape should have two holes, which can be achieved by visualizing the 6_1 knot as a $(5,-1,-1)$ pretzel knot. The flat style, closest to the original braid representation, is messy. Merging bands and eliminating disks gives the more compact reduced representation, but these steps cannot reduce the genus.

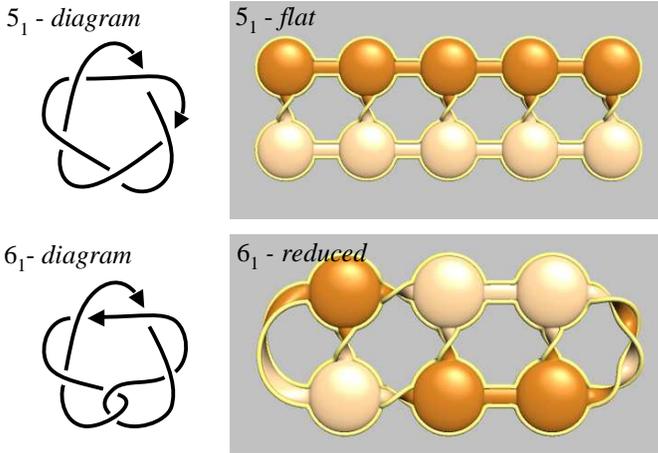


Figure 19: 5_1 cinquefoil knot ($AAAAA$) and 6_1 knot ($AABacBc$)

This limitation can be explained in various ways. The main difference between the upper parts of 5_1 and 6_1 is that in the former the strands run parallel, while in the latter their directions are opposite. The braid notation excels in representing parallel twisted strands, but cannot compactly represent twisted strands with opposite directions. Knots with many crossings and a low genus typically have twisted strands with opposite directions, pretzel knots are a good example of these.

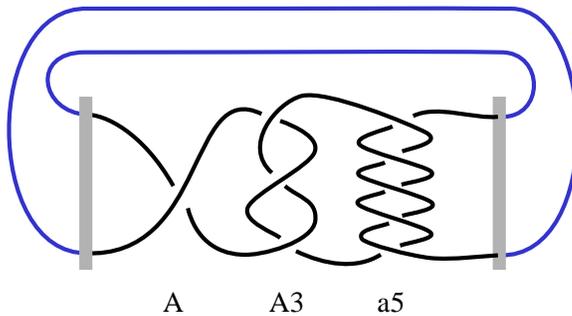


Figure 20: Extended braids: multiple vertical twists

We implemented a simple extension to handle a large number of such knots as well. In the letter based braid notation, each symbol represents a single twist of two parallel strands. We extended this by allowing also the definition of vertical twists (Fig. 20). Each letter can be followed by a number that gives the number of vertical twists, such that for instance

a $(1, 3, -5)$ pretzel knot is defined as $AA3a5$. One limitation we impose is that the number of vertical twists should be either odd or even for all bands connecting the same disks, i.e., the same for all A 's and a 's, all B 's and b 's, etc. If this condition is met, then processing these extra twists is straightforward. One change is that when even twists are used, the orientation of disks changes. With this extension shapes such as chain rings can be defined easily via a sequence $A2A2\dots$ (Fig. 21).

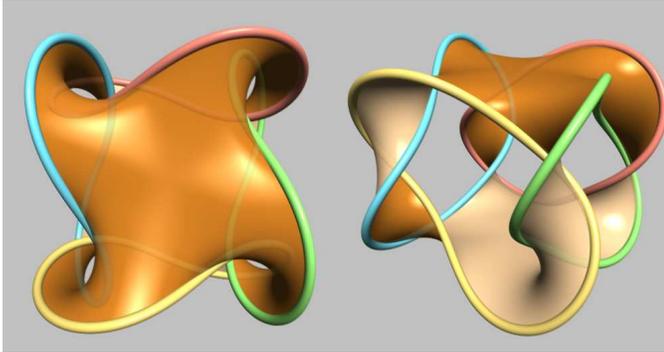


Figure 21: Chain ring (A2A2A2A2)

VI Results

A. Examples

Interactive viewing provides much better insight in the 3D shape than watching static images. Nevertheless, we show some more examples of results. As mentioned in the previous section, the braid representation does not always yield a surface with minimal genus. This property can also be used as a feature, i.e., to produce surfaces with a high genus that are bounded by simple knots and links. Consider the knots and links produced by a sequence $AaAaAa\dots$. One strand is always on top of the other here (Fig. 22), hence this produces either two loose rings or one unknot, for an even or odd number L of letters, respectively. The Seifert surface is more complex, and contains $L - 1$ holes (Fig. 23). The result of $AaAa$ is intriguing. Locally, the shape is simple to understand, but it is hard to form a mental image of the complete shape, like one can imagine a sphere or a torus.

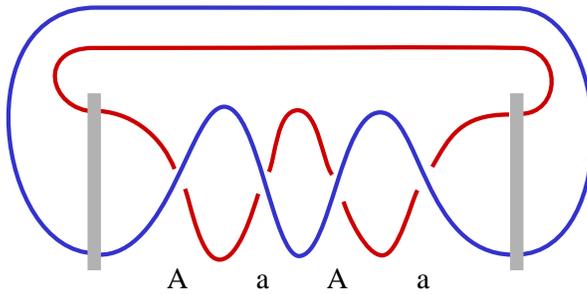


Figure 22: $AaAa$ gives simple boundaries, but a complex topology of the surface

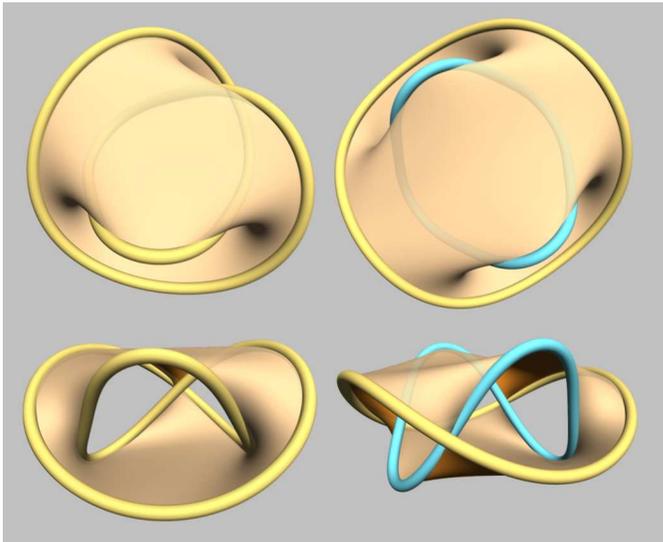


Figure 23: AaA (left) and $AaAa$ (right)

Fig. 25 shows a number of standard knots, Fig. 26 shows a number of standard links. For each knot or link two views are given: one with a minimal number of crossings and one that shows the spatial structure of the surface. In [15] we have given examples of the same set, using stacked and reduced styles, in combination with geometric smoothing. Whereas these images showed the structure clearly, the use of physically based smoothing leads to results that resemble the natural shapes of the knots much better.

B. Dissemination

The visualization of Seifert surfaces is useful for knot theorists to illustrate and explain their work. Our first experience in a course on knot theory was

very positive in this respect. Also, we think that the concepts presented and methods used here are interesting for a wider audience. Knot theory is pure mathematics, but can be presented at a basic level without any formula. In this spirit, our work could be used for tutorial and educational purposes, such as for instance special projects on higher mathematics at high schools. We already spent some effort in bridging the gap between our research results and application on a wider scale.

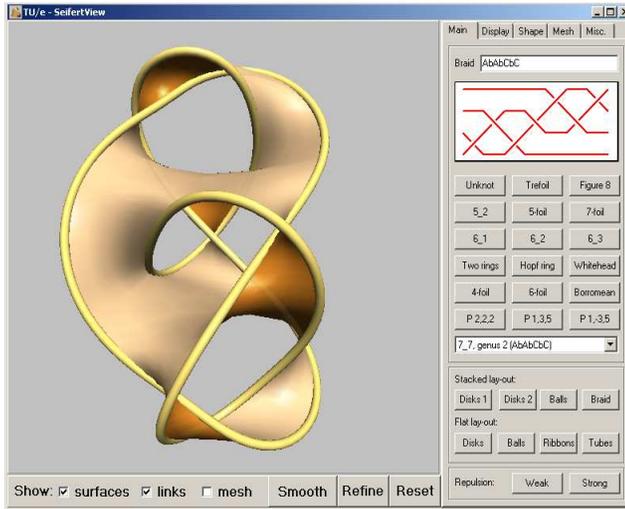


Figure 24: User interface SeifertView

First of all, we have tried to turn our research prototype into a useful and interesting tool for an extended audience. The result is a Microsoft Windows application, which we have called *SeifertView*. A snapshot of the user interface is shown in Fig. 24. The user can view and rotate the knot (here knot 7_7) in the main area. With the controls below the main view area, the user can select which parts have to be shown, trigger a smoothing cycle, refine the mesh, or reset to the original shape. The first tab sheet, shown on the right, provides basic functionality which enables an occasional user to have a quick result. The user can define knots and links by pressing a button, via specification of a braid word, or by selection from a table with all knots having up to ten crossings¹ (obtained from [10]). A schematic representation of the corresponding braid is shown. Eight presets are offered to select a presentation style. Pressing such a button not only selects a different algorithm, but also dimensions are

¹Added in this republication: This table was derived by Gittings, and was obtained from [10]. See: Thomas A. Gittings. Minimum Braids: A Complete Invariant of Knots And Links. arXiv:math/0401051, 2004.

tuned to obtain a satisfying result. Furthermore, a selection for weak or strong repulsion during smoothing is offered. The other tab sheets contain a large number of options for tuning various aspects, such as the shading, the geometry, and the mesh generation and relaxation.

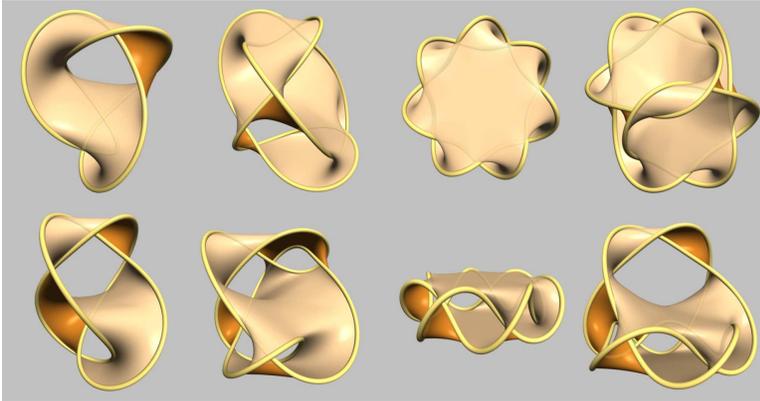


Figure 25: From left to right: Figure eight knot, knot 6_3 ($AAbAbb$), knot 7_1 ($AAAAAAA$), and knot 8_5 ($AAAbAAAAb$)

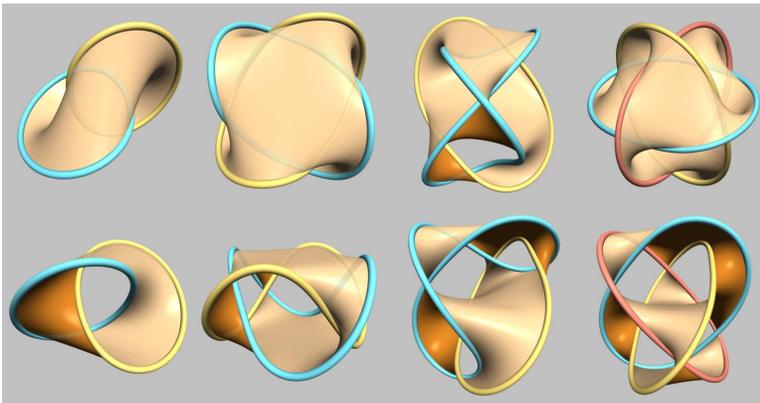


Figure 26: From left to right: Hopf link (AA), link 4_2^1 ($AAAA$), Whitehead link ($AbAbb$), and Borromean rings ($AbAbAb$)

We have added various features based on discussions with prospective users. For instance, an option is provided to hide all controls for classroom presentation purposes. Also, an option is offered to produce anti-aliased high resolution images for printing purposes directly from the application. As an illustration, in Fig. 27 the effect of oversampling each pixel 25 times,

using a jittered grid, and averaging with a Mitchell filter is shown for a small (200×100) image. By means of tiling, images with a resolution of 3000×3000 can be produced. Finally, we offer a special feature for a younger public: users can study a knot in detail with a thrill ride in a roller coaster (Fig. 28).

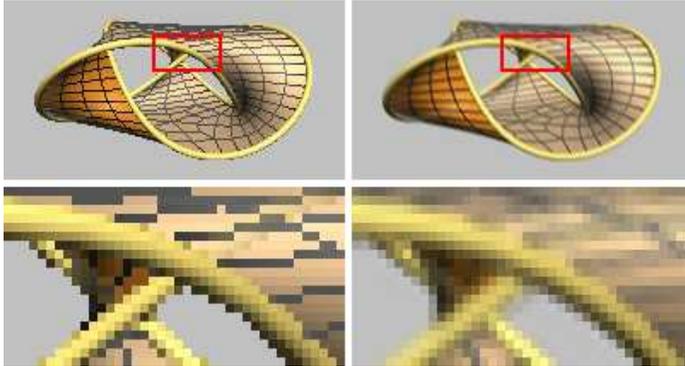


Figure 27: Anti-aliasing: left a screenshot, right an anti-aliased version

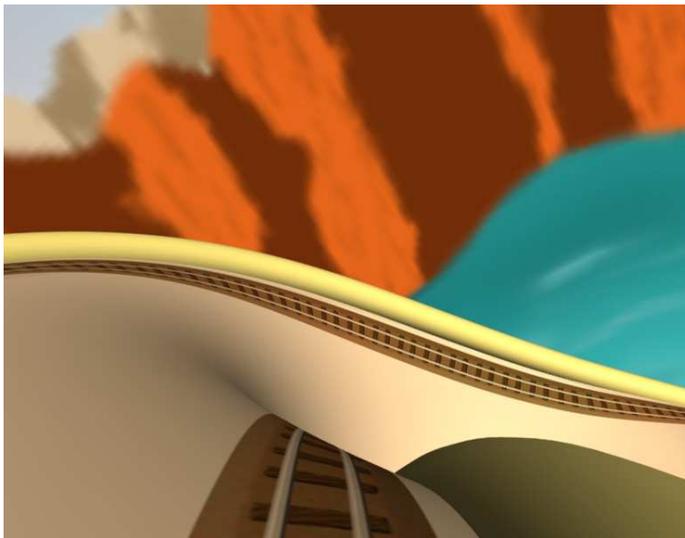


Figure 28: A special effect

SeifertView is available for download from [14]. On this web site we furthermore provide a short and informal introduction to Seifert surfaces, the braid representation, and various options and features of our tool.

VII Discussion

We have presented a method for the visualization of Seifert surfaces, and have introduced closed Seifert surfaces. These surfaces are generated starting from the braid representation; several styles can be used, and by varying parameters the user can produce different versions. Via physically based smoothing, attractive knots can be generated in seconds.

In this field, one answer gives immediately rise to new questions. Some examples are the following. Physically based smoothing leads to attractive surfaces; we would like to have a procedure that gives smooth closed surfaces (see section III). This requires a modified relaxation method with for instance extra outward pressure on the surface.

We are interested in producing minimal genus surfaces for knots and links. Allowing multiple twists does increase the flexibility, but we have not yet found an algorithm to convert a braid representation (or other representation) into this new representation, and also we do not know if this extended braid notation suffices to produce any minimal genus knot. If this can be done for all different knots, tables and overviews of Seifert surfaces can be generated automatically. Another future goal is to create Seifert surfaces from arbitrary given closed loops. That is, the input would be a geometric model of the knot, rather than the braid notation or other symbolic representation.

Another remaining puzzle concerns the morphing of shapes. For instance in Fig. 7, all shapes are isotopic, but we would like to exhibit this via a smooth animation.

Finally, so far we concentrated on visualizing Seifert surfaces, but these are not the only possible surfaces bounded by knots. Also, Seifert surfaces play an important role in computing linking numbers, fluxes, and circulations for space curves. Visualizing these would be helpful in a wide range of applications ranging from knot theory to electromagnetism to fluid dynamics.

Bibliography

- [1] C. Adams. *The Knot Book: An elementary introduction to the mathematical theory of knots*. W.H. Freeman and Company, 1994.
- [2] D. Bar-Natan. www.math.toronto.edu/~drorbn/katlas, 2004.
- [3] E. Catmull and J. Clark. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, 1978.

- [4] E. E. Catmull and R. J. Rom. A class of local interpolating splines. In *Computer Aided Geometric Design*, pages 317–326. Academic Press, 1974.
- [5] G. di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing – Algorithms for the visualization of graphs*. Pearson, 1998.
- [6] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C (2nd Edition)*. Addison-Wesley, 1995.
- [7] P. Frankl and L. Pontrjagin. Ein Knotensatz mit Anwendung auf die Dimensionstheorie. *Math. Annalen*, 102:785–789, 1930.
- [8] L. Kauffman. *On Knots*. Princeton University Press, 1987.
- [9] C. Livingston. *Knot theory*. Math. Assoc. Amer., 1993.
- [10] C. Livingston and J. C. Cha. Knotinfo: Table of knot invariants. www.indiana.edu/~knotinfo/, 2004.
- [11] D. Rolfsen. *Knots and links*. Publish or Perish, 1976.
- [12] R. G. Scharein. *Interactive Topological Drawing*. PhD thesis, Department of Computer Science, The University of British Columbia, 1998.
- [13] H. Seifert. Über das Geschlecht von Knoten. *Math. Annalen*, 110:571–592, 1934.
- [14] J. J. van Wijk. www.win.tue.nl/~vanwijk/seifertview, 2005.
- [15] J. J. van Wijk and A. M. Cohen. Visualization of the genus of knots. In C. Silva, E. Groeller, and H. Rushmeier, editors, *Proceedings IEEE Visualization 2005*, pages 567–574. IEEE Computer Society Press, 2005.

Jarke J. van Wijk

Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands

j.j.v.wijk@tue.nl

<http://www.win.tue.nl/~vanwijk>

Arjeh M. Cohen

Department of Mathematics and Computer Science, Technische Universiteit Eindhoven, The Netherlands

A.M.Cohen@tue.nl

<http://arpeg.nl/arjeh/>