www.cad-journal.net
Computer-Aided Design
AND APPLICATIONS

Cable-Knots and Knotted Braids

Carlo H. Séquin

University of California, Berkeley, sequin@berkeley.edu

Abstract. Abstract geometrical sculptures are generated by sophisticated sweeps along the curves of mathematical knots. First, a knot is laid out as an aesthetically pleasing 3D space curve, and an arbitrary cross-sectional profile is swept along that curve. The complexity of the underlying knot is enhanced by running a "multi-strand cable" along the knot curve and twisting this cable in such a way that different strands connect into other strands in a cyclical manner, thus forming a single closed loop, representing a much more complex mathematical knot. The complexity of this knot can be further enhanced by locally intertwining neighboring strands, by giving the cable a significantly higher amount of twist, or by braiding the strands of the cable. Some of the resulting CAD models have been turned into sculptural maquettes using layered fabrication on inexpensive 3D printers.

Keywords: Geometric Sculptures, Mathematical Knots, Sweep Generator, 3D-Prints.
DOI: https://doi.org/10.14733/cadaps.2024.510-521

1 INTRODUCTION

For the last four decades, mathematical knots served for me and my students as inspiration and as a framework to design abstract geometrical sculptures [7]. In a project called "The Beauty of Knots" [6], we start with a knot from the mathematical knot tables [4] and geometrically deform that closed-loop curve in 3D space to take on a pleasing shape, often with some degree of rotational symmetry. A simple profile in the shape of some regular n-gon, a simple crescent, or a small star shape is then swept along this curve. This may result in some pleasing shapes that can then be realized as small models fabricated on an inexpensive 3D printer (Figure 2(a,b)), or in some bronze sculptures (Figures 2(c), 7(a)).

In the current development, we greatly enhance the complexity of the underlying mathematical knot by following the basic knot curve with a *multi-strand cable*, and by interlinking the strands in different ways.

2 KNOT SCULPTURE DESIGN

The design process typically starts with a 2D knot diagram taken form the mathematical knot table [4] or sketched freehand by the designer. This diagram is deformed into a pleasing planar layout, possibly with some simple rotational symmetry. This can be done conveniently by suitably placing control points for a cubic B-spline, thus assuring overall G1- (tangent) and G2- (curvature) continuity. A symmetrical trefoil can easily be done with just 9 control points (Fig. 1a). The 2-lobe D_2-symmetrical version of knot_3_1 may employ 16 control points to yield good control in the helically intertwined central region (Fig. 1b), and the Knot_5_2 can be modeled with just 12 control points (Fig. 1c). Some of the coordinates of these control points can be set as variable parameters that later can be adjusted interactively with on-screen sliders, while the resulting overall shape is being displayed.

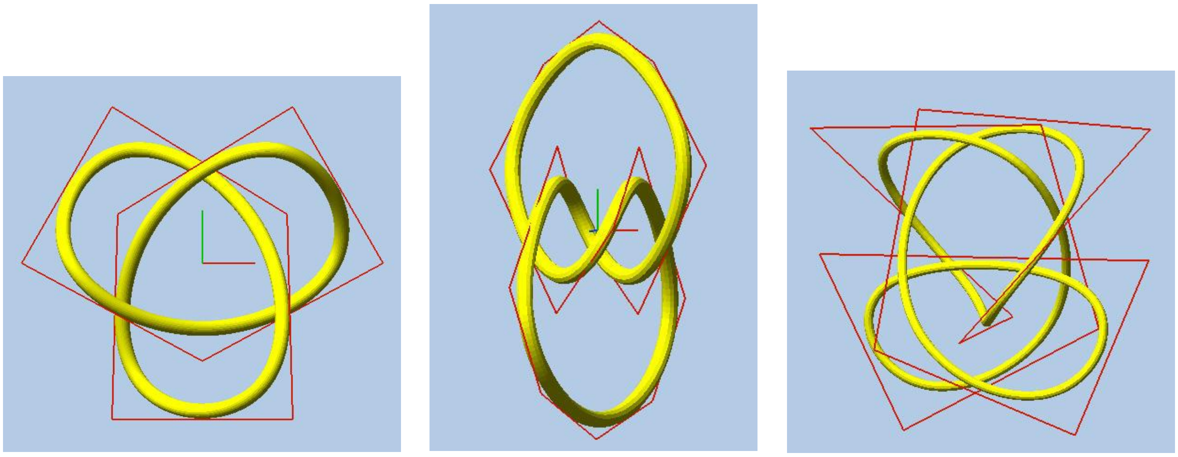

Figure 1: Simple knot curves: (a) D_3 Trefoil, (b) D_2 Trefoil, (c) Knot_5_2.

The next step is to add a 3D component. For instance, one might add a couple of control points at every curve-crossing, so that one of the two crossing lines can be raised in *z*, while the other one is being pushed down correspondingly. Again, the amount of this 3D offset may be introduced as an adjustable parameter. The resulting curve path is displayed with an interactive graphical "crystal ball" interface, and all sliders are fine tuned to obtain a pleasing 3D space curve.

Finally, a cross-sectional profile must be chosen, to be swept along the knot curve, possibly with some twisting and changing in scale. There are two modes in which this profile can be applied to the knot curve.

The initial sweep may start out with *minimal torsion*, where the cross-section is pushed forward while keeping the locally occurring torsion to zero. For arbitrary space curves, this will typically result in a sweep that does not naturally close onto itself but will exhibit some azimuthal rotation difference around the local tangent at the point of closure. This can be fixed by adjusting an overall *twist* parameter. This azimuthal adjustment can be further fine-tuned by introducing additional "control-points," where the local azimuthal rotation can be adjusted.

The other approach is using an *intrinsic* mode. Here, the *x*-axis of the defined 2D profile always points away from the local center of curvature. This mode automatically generates nice saddle surfaces where the knot curve forms curved lobes (e.g., top of Figure 2(a)). The disadvantage is that in the neighborhood of a nearly planar inflection point, the cross section will make 180° azimuthal flip, which may locally pinch down the generated sweep into an "hour-glass" configuration. The designer still has the freedom to make local adjustments to the shape and orientation of the cross-section. Figure 2 shows examples of sculptural maquettes designed with the method described above.

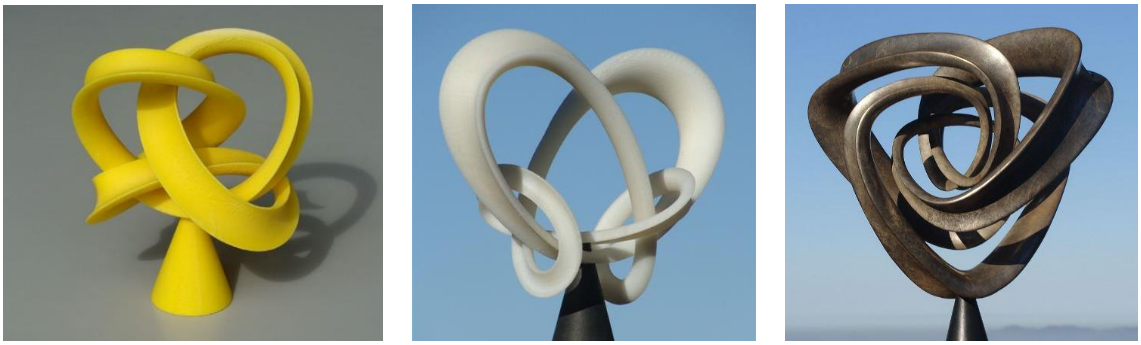

Figure 2: "The Beauty of Knots": (a) Knot_5_2, (b) Knot_7_7, (c) Torus-Knot(3,5).

2.1 The CAD Environment

Most of the design experiments described below have been carried out in the context of some home-brewed CAD systems. Over the last 30 years, this design environment has evolved from Berkeley SLIDE (Scene-Language for Interactive Design Environments) [9] to NOME (Non-Orientable Manifold Editor) [5] and to JIPCAD (Joint Interactive & Programmable CAD) [1]. They all allow to start with some short program fragments to define a sweep curve and/or its cross-sectional profile. The programs typically have a set of parameters that can be adjusted interactively through on-screen sliders, while the overall resulting geometry is displayed.

SLIDE started out as the design environment used in the Computer Graphics and Modelling course CS 184 at UC Berkeley. NOME and JIPCAD gradually added the capability to add and/or modify geometry in an interactive setting on a graphics screen. The main challenge in developing this editing capability was in finding ways to integrate these manually added enhancement into the original program that describes the complete sculpture geometry [5].

All these programs have powerful sweep constructs that allow arbitrary cross-sections to be swept along any space curve in 3D space. Moreover, at various user-defined "control-points" along the sweep curve, the given cross-section can be scaled non-uniformly in *x* and *y*; and it can also be rotated around the local tangent of the sweep curve. In a further enhancement, called *sweepmorph*, cross-sections are allowed to change their shapes more dramatically, with the main restriction that all the cross-sectional profiles have to be specified with the same number of points around their perimeters. These sweep generators were the main tools used in defining the cable-knots and the knotted braids.

JIPCAD files do not look much different from the other CAD files. They introduce vertices that define faces, which are combined into meshes, which then may be grouped into parts geometries. The latter can then be instantiated multiple times in different positions and orientations, thus exploiting any symmetries in the final model. The strength of JIPCAD lies in its built-in generators, in particular the sweep generator. A detail description of the operation of JIPCAD is given in reference [1].

The final geometry of all the knot sculptures is a boundary representation of a long, cylindrical tube closing onto itself. As such, it could readily be subjected to a subdivision process to obtain an arbitrarily smooth surface. However, it is much simpler to obtain a more finely tessellated surface by increasing two relevant geometrical parameters in the sweep generator: the number of radial segments specifying the swept cross-section, and the number of longitudinal steps taken along the knot curve.

JIPCAD, like most other geometrical CAD tools, can readily output the final boundary representation as an .STL-file, which is accepted by most 3D-printers. However, it takes a bit of experience and judgement to decide how to orient the part on the 3D-printer in order to obtain the best support scaffolding that robustly supports all branches of a braid, and is still not too difficult to remove, if manual support removal is required.

3 INCREASING KNOT COMPLEXITY

Once a pleasing symmetrical knot layout has been found, the complexity of the knot can readily be increased by running a *multi-strand cable* (Figure 3(a)) along the knot curve. The cable is twisting in such a way that the cable strands connect into other strands in a cyclical manner (Figure 3(b)) and thus form again a single, closed loop, representing a much more complex mathematical knot.

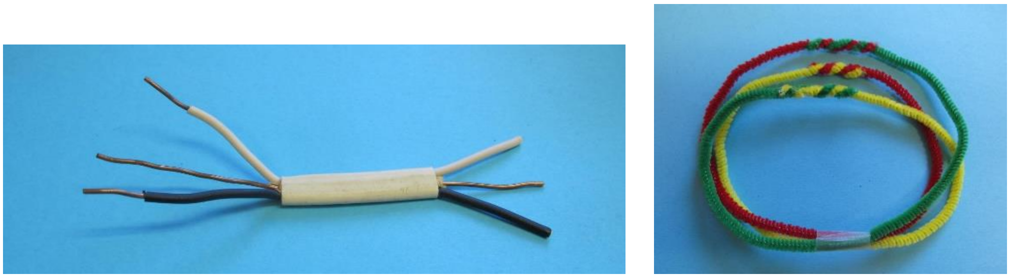

Figure 3: (a) A 3-strand *cable*. (b) Closing a twisted cable into a single-strand loop.

3.1 Cable Un-Knots

To demonstrate the concept of a cable-knot, we run a 3-strand cable along a simple circular un-knot and give the cable a 120° twist before joining its ends. This results in a classical *1_3_Torus Knot* (Figures 4(a,b)). Mathematically, this is still an un-knot, but now with a more interesting geometry.

We can further modify the geometry by warping the shape of the underlying donut around which the torus knot has been wound. Figures 4(c,d) show two views of the modified, warped geometry. The shown 3D-print models are about six inches tall.

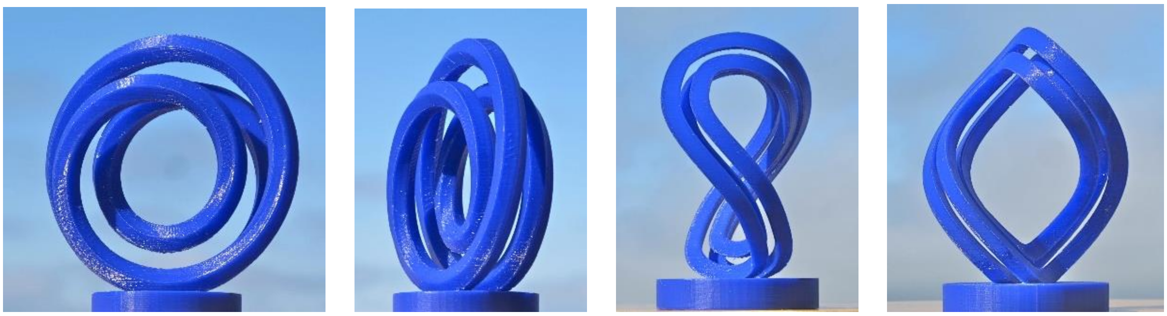

Figure 4: Un-knots: (a,b) A simple torus knot, front and side; (c,d) warped version of the same knot.

3.2 Trefoil Knots

More interesting sculptures result when we let the cable itself form a real, non-trivial knot. The simplest such knot is the Trefoil, a 3-crossing knot, also known as Knot_3_1. Figure 5(a) exhibits its classical layout with 3-fold rotational D_3-symmetry. If a 3-strand cable follows this curve and exhibits a 120°-twist before closing, a more complicated knot results (Figures 5(b,c)). Wherever the basic Trefoil exhibits a simple crossing, we now find two groups of three strands each crossing over one another, thus generating 3×9 crossings. The 120°-twist of the cable adds two more crossings, since one strand needs to cross over the two other ones. Thus, overall, we end up with a 29-crossing knot. Figure 5(b) shows a CAD model that exhibits the cyclical connectivity. Figure 5(c) depicts a 3D-print model.

To generate such 3-strand cable-knots, we could run a special sweep along the trefoil curve, using a compound cross-sectional profile that comprises three separate patches with the chosen cross-section for the strand, – in this case, an equilateral triangle (Figure 5(d), top). We then apply a 120°-twist to the cable as a whole. But there is another way to achieve the same result. We can use a simpler cross-section profile with just a single, offset strand-profile (Figure 5(d), bottom) and run this profile three times along the basic trefoil curve with an overall twist of 3×120° = 360°. The azimuth rotation of the cable has been chosen in such a way that two strand-lobes touch the cylindrical base of the sculpture.

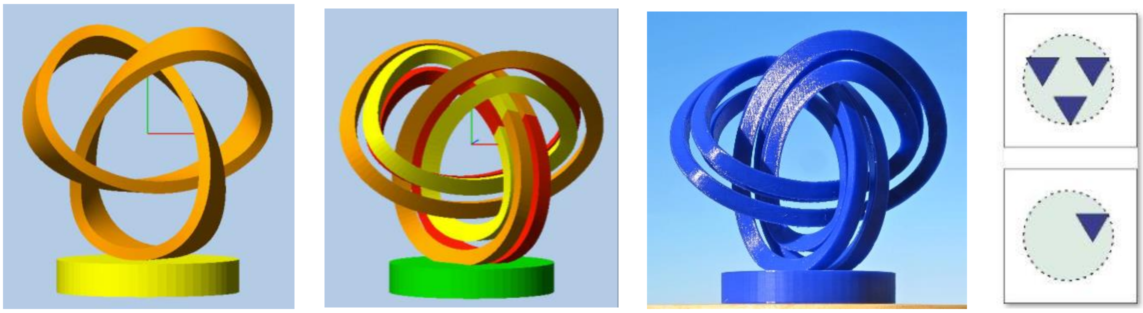

Figure 5: (a) Trefoil knot. (b) CAD model of a 3-strand Trefoil cable-knot. (c) A 6-inch-tall 3D-print. (d) Two options of defining the cross-sectional profile to be swept along the knot curve.

There is a different layout of the Trefoil knot that exhibits overall D_2-symmetry with three mutually orthogonal 2-fold rotation axes (Figure 6(a)). Running a twisted 3-strand cable along this curve results in the 2-lobe Trefoil cable-knot shown in Figures 6(b,c).

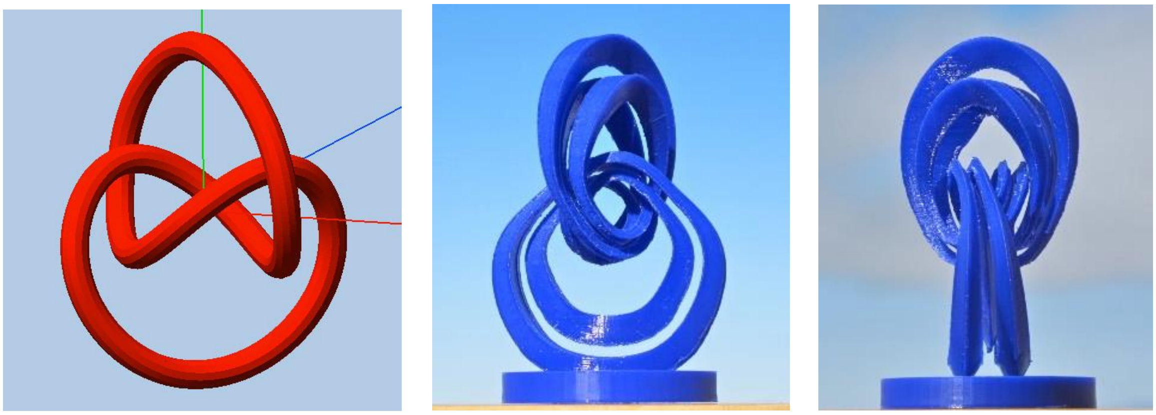

Figure 6: (a) Trefoil knot with D_2-symmetry. (b,c) A “2-lobe” version of a Trefoil cable-knot, two views.

3.3 Figure-8 Knots

The next more complicated knot after the trefoil is the 4-crossing Figure-8 knot. This knot has a pleasing, D_2-symmetrical layout (Figure 7(a)) with two lobes to stand on, and two more lobes stretching towards the sky. Running a 3-strand cable along this knot curve results in the Figure-8 cable-knot shown in Figures 7(b,c). Again, the azimuth rotation of the cable has been adjusted so that two strands in each of the bottom lobes connect to the supporting base.

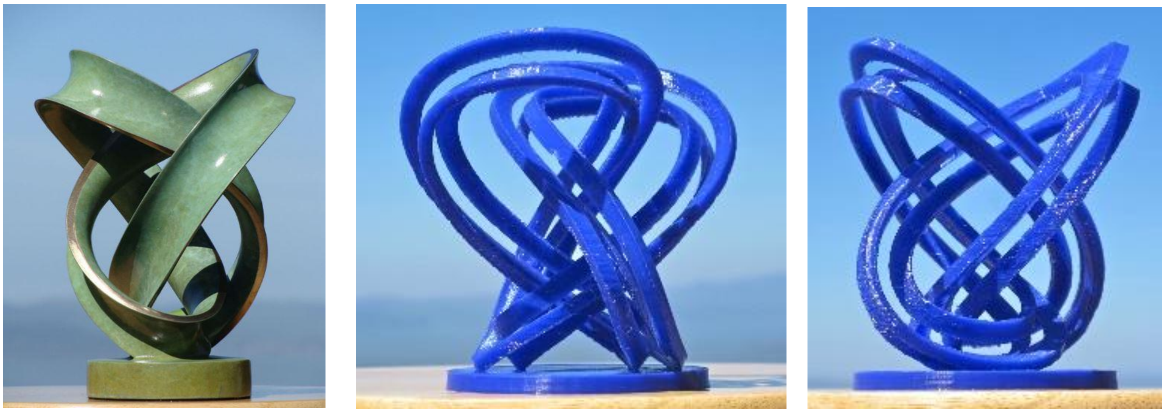

Figure 7: (a) 4-lobe layout of the Figure-8 knot. (b,c) 3-strand Figure-8 cable-knot, two views.

Just as for the Trefoil knot, there exists a D_2-symmetrical, 2-lobe layout of the Figure-8 knot (Figure 8(a)). A corresponding 3-strand Figure-8 cable-knot is shown in Figures 8(b,c) from two different viewpoints.

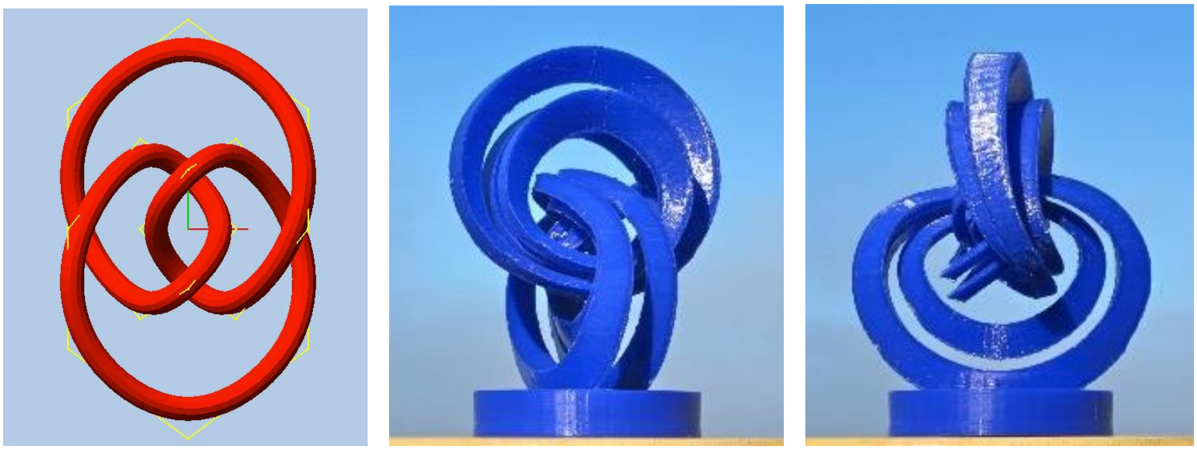

Figure 8: (a) 2-lobe layout of the Figure-8 knot. (b,c) 2-lobe variant of the Figure-8 cable-knot.

4 FURTHER ENHANCEMENT OF KNOT COMPLEXITY:

The complexity of these sculptural maquettes and of their underlying knots can be further enhanced in several different ways.

First, we can simply use cables with more strands. Figure 9(a) shows a 4-strand Figure-8 cable-knot that follows a knot curve that is more typical of the diagrams that one finds in a table of mathematical knots [4]. A corresponding 7-inch-tall maquette made on a 3D-printer is shown in Figure 9(b).

Alternatively, we can stick with a 3-strand cable, but let the three strands be more heavily intertwined by giving the cable a substantial amount of additional twist, – while still making sure that the three strands connect to one another in a cyclical manner. Figure 10 shows the result for a 3-strand cable-Trefoil with about ten full turns of additional cable-twist.

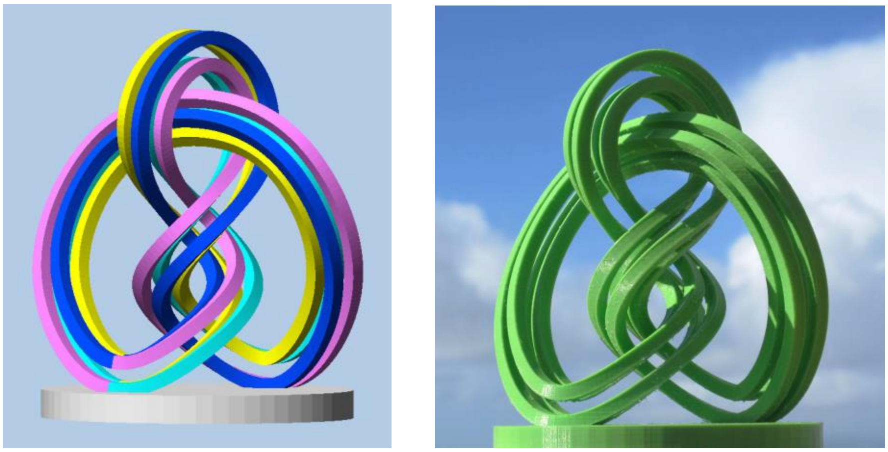

Figure 9: (a) 4-strand Figure-8 cable-knot with 90° twist. (b) Corresponding 3D-print.

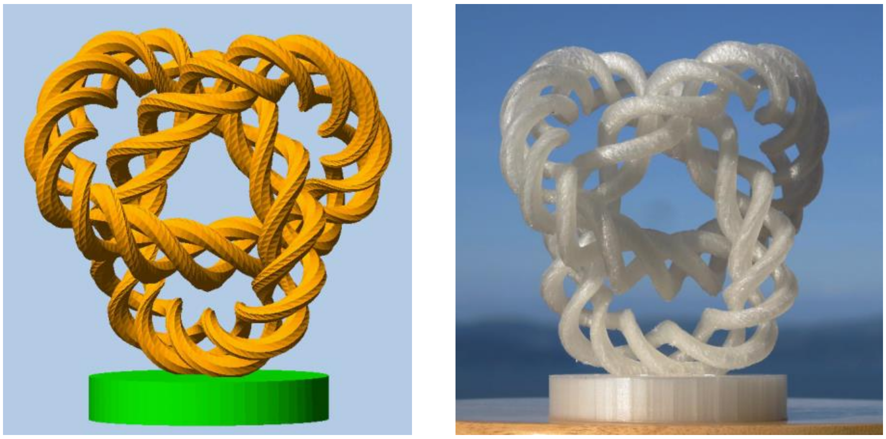

Figure 10: 3-strand Trefoil cable-knot with 3480° twist: (a) CAD model; (b) FDM 3D-print.

5 BRAIDING

One further step to increase knot complexity is to let the multiple strands in the cable form a braid. Figure 11(a) shows a classical 3-strand braid, and Figure 11(b) displays a more complex, flat, ribbon-like braid with six strands. If such a flat braid is wrapped around a cylinder and closes onto itself, it forms a *Turk's Head* knot. Such knots, made of rawhide or latigo leather, are often used by cowboys to form decorative handles for switches and other gear used around horses [3],[2]. They are characterized by two numbers: *p*, the number of *parts* (or strands, or thongs), and *b*, the number of *bights* (or bulges) along the edges of the braided ribbon.

Within the constraints of our goal to obtain a single mathematical knot (Figure 12(a)), we must pay attention to the width (*p*) and length (*b*) of the braid making sure that it joins to form a single closed-loop strand. In general, it requires that *p* and *b* are mutually prime. If they have a common divisor, like the common factor of 2 used in Figure 12(b), we obtain a link with two individual loops, shown in pink and yellow.

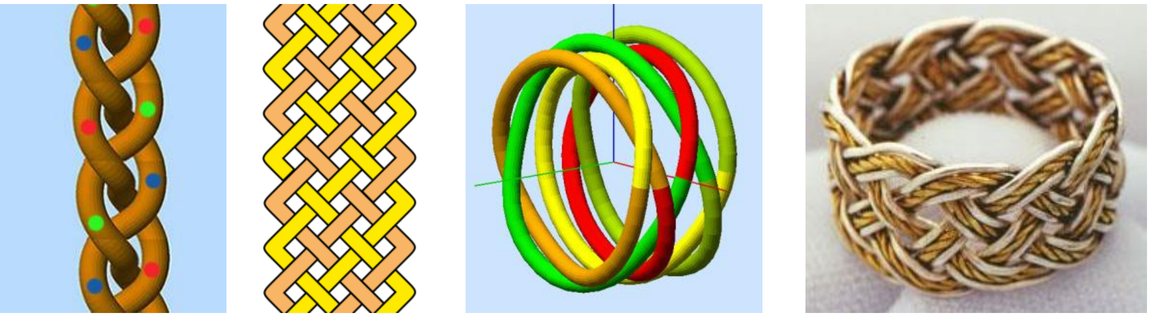

Figure. 11: Braids: (a) *p*=3; (b) *p*=6. Turk's Head knots: (c) *p*=5, *b*=6; (d) *p*=5, *b*=11.

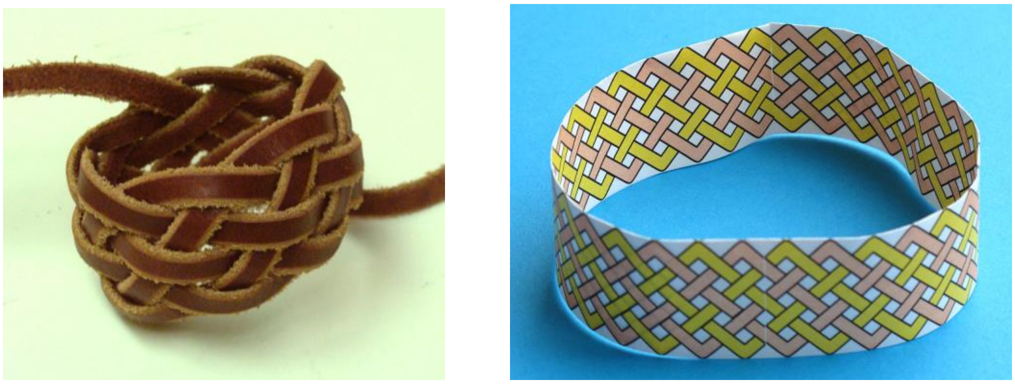

Figure. 12: (a) Single-strand latigo leather knot; *p*=6, *b*=7. (b) Turk's Head link with 2-loops; *p*=6, *b*=32.

The strands in these braids form a simple *Tabby-weave* pattern, where every strand adheres to a strict *over–under–over–under* sequence. Of course, more complicated multi-strand braids could be used to obtain geometrically more intriguing patterns. But our focus in the next sections is to apply braids to more interesting mathematical knots.

5.1 Twisted Turk's Head Knots

As a first step towards more interesting knot structures, we can give the Turk's Head braid a Möbius twist (Figure 13(a)). To end up with a single closed-loop strand, we might think that we can simply reduce the length of the loop to 31 bights, which is mutually prime with 6. But if we take a 2-colored braid of width 6 and length 31 and join its ends after a 180° flip, we obtain a nice color-consistent match, showing that this is again a 2-thong Möbius braid (Figure 13(a)).

The closure of a Möbius band reverses the left-to-right pattern of the strands; this must be compensated for: Half a braid-period also reverses the strand pattern. Thus, to make a single-thong knot, we may select two mutually prime numbers and then subtract or add a half-period set of bights (i.e., *p*/2 for simple Tabby-weave braids). We then find that for 6-part braids all even numbers of bights work – except for multiples of six. Figure 13(b) displays a twisted Turk's Head knot in which the single thong has been given a gradual rainbow-coloring that traverses through the whole hue circle.

Figure 13(c) depicts a triply-twisted Möbius braid, woven from a single latigo leather thong by Lorenzo Larrucea [8].

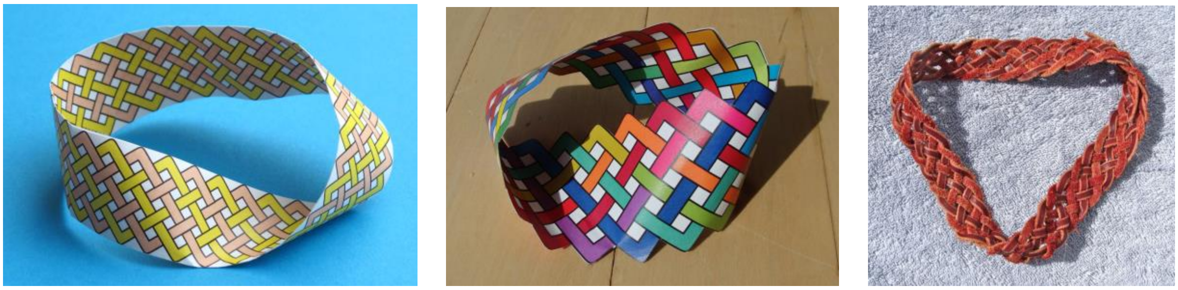

Figure 13: Möbius-twisted Turk's Head knots: (a) 2-loops, *p*=6, *b*=31; (b) single thong, *p*=6, *b*=16; (c) triply-twisted Möbius braid made from latigo leather, *p*=6, *b*=34.

5.2 Knotted Braids

Making CAD models of braids that flow along more complex knot curves poses new design challenges. Typically, a set of points that define the path of an individual strand can be generated by a small program that captures the motion of a strand in subsequent cross-sectional slices as they move along the knot curve. For the simple 3-strand braid (Figure 11(a)), the strand performs a repeated figure-8 motion as shown in Figure 14(a). This motion can be captured in a coarse manner with as few as 6 vertices (red polyline), or more smoothly with many more vertices (yellow B-spline curve). If we sweep such a cross section along the knot curve, we obtain a "multi-tube worm," and the sweep paths for individual strands will lie in the surface of this worm. More complex braids, such as the one shown in Figure 11(b) require a cross-sectional motion diagram with more "tubes" (Figure 14(b)).

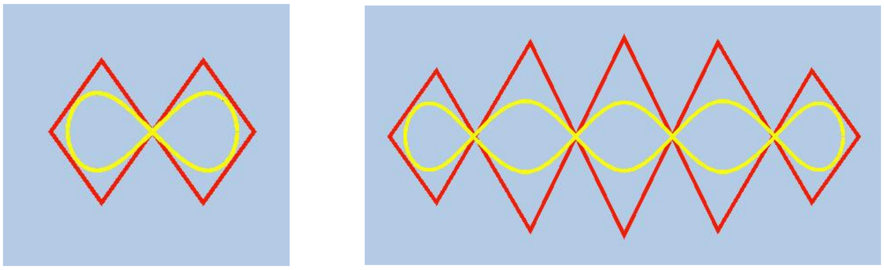

Figure 14: Cross-sectional movement of strands: (a) for 3-strand braid; (b) for 6-strand braid.

Our JIPCAD environment [1] makes the definition of these sweep paths particularly easy, since it allows a carefully chosen subset of vertices in the "multi-tube worm" surface to be connected into a polyline that defines the desired sweep path for a strand. In the simplest definition of the two cases shown in Figure 14, that worm surface forms a rectangular mesh that is 6 or 12 segments wide and as long as the number of samples we choose to place along the knot curve. JIPCAD automatically assigns names to the mesh vertices in the form of "*mesh_y_x*". For a coarse, 3-strand braid (Figure 14(a)) *x* runs from 0 through 5, and *y* runs from 1 through the numbers of samples taken along the knot curve. The sweep path for the strand is then given by a "diagonal" line in this worm surface mesh, starting with: "*mesh_1_0, mesh_2_1, mesh_3_2, mesh_4_3, mesh_5_4, mesh_6_5, mesh_7_0, mesh_8_1, …*" This coarse description of the cross-sectional motion-path is sufficient to define the geometry of the strand's sweep-path since this coarse polyline can serve as a control polygon for a cubic B-spline that allows to generate smoothly curving sweeps for each of the three strands. All three strands follow that same type of diagonal paths in this rectangular mesh, but with different phase-offsets between parameters *y* and *x*. To guarantee that the braid closes into a single-loop strand, the considerations of primality, discussed in the previous section, apply again.

Now we can apply this process for creating a 3-strand braid to a simple Trefoil knot, placing 9+2/3 braid periods (58 steps) along the knot curve. This length gives us the desired cyclical connectivity between the three strands. The transition from one strand into another is shown above the center of Figure 15(a).

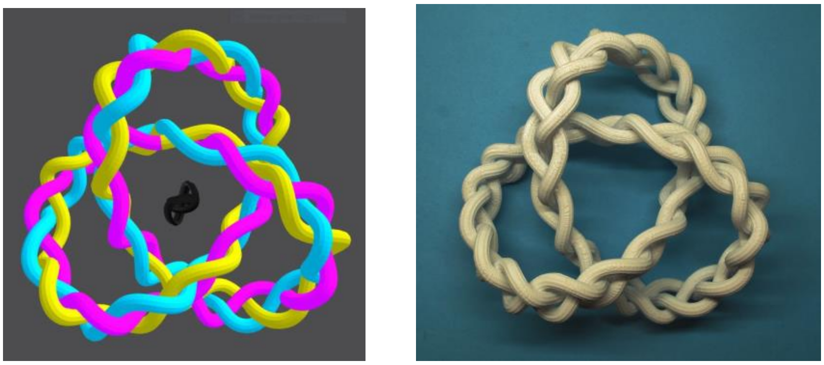

Figure 15: Trefoil knot with a tight 3-strand braiding: (a) CAD model; (b) 3D-print.

As a second example we use Knot_5_2 (Figure 2(a)) and follow it with different braids. Figure 16 shows the result when using a simple 3-strand braid.

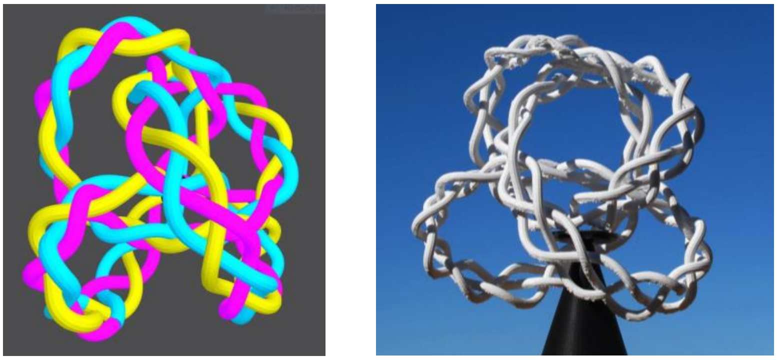

Figure 16: Knot_5_2 with a 3-strand braid: (a) CAD-model; (b) 3D-print.

Figure 17 uses two different kinds of braids, first a flat 4-strand ribbon (Figure 17(a)), and another variant where the four strands are bundled into a more cylindrical cable (Figure 17(b)).

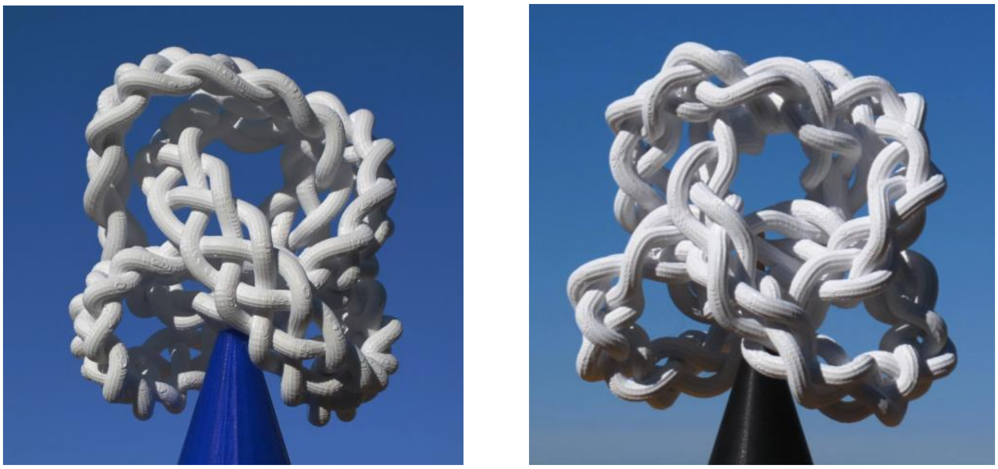

Figure 17: (a) Knot_5_2: (a) with a flat 4-strand braid; (b) with a cylindrical 4-strand braid.

5.3 Braided Braids

A plausible question arises: What will happen if we apply the braid-generation process in a recursive manner to the strands of an already braided structure? Figure 18(a) shows the concept: On the left is a 3-tube worm that forms the surface that could accommodate a 4-strand braid. Next to it is one of the four strands shown in yellow, but now with a 2-tube cross-section that can serve as a carrier for a 3-strand braid. And further to the right are the 3 sweep paths for such a braid.

Figure 18(b) depicts the completed braided braid with all four strands of the major braid shown in the colors: yellow, blue, red, and green, as well as the four completely instantiated minor braids.

In principle there is no reason why the recursive braiding could not be carried out through more than just two recursion steps. However, the current capabilities of 3D-printing render this approach of limited practicality.

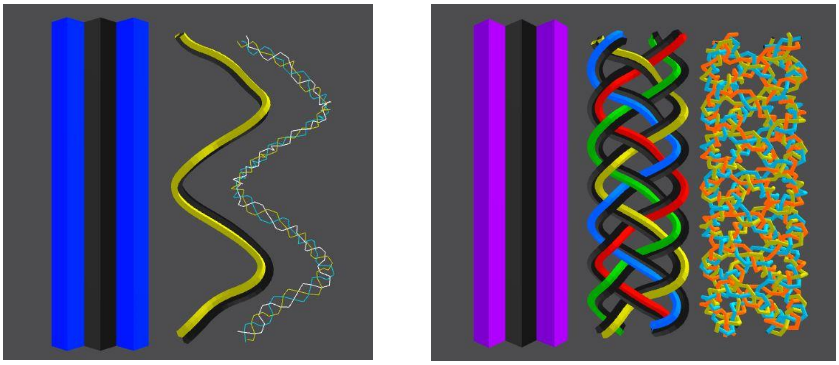

Figure 18: Braided braids: (a) a single strand of the major braid (b) the complete major braid, as well as the four minor braids fully instantiated.

6 CONCLUSION AND FUTURE WORK

Cable-knots and their highly twisted or braided variants offer a rich playground for geometrical sculptures. First there are several topological parameters to choose from: the type of knot selected to place the cable, the number of strands, the braid type, and the amount of twist applied when closing the cable loop.

In this study the focus has been on the basic topology and overall structure of the resulting sculptural forms. For this purpose, in most examples the individual strands have a simple triangular or circular cross-section. When building a real sculpture, additional geometrical issues must be addressed. The shape of the strand and its azimuthal orientation need to be carefully selected. In Figures5(b,c) the triangular strand has been oriented so that one of its sides lies flat on the cylindrical pedestal. On the other hand, in Figures 7(b,c) the triangular strand penetrates into the base with one of its corners. Braiding the cable-strands adds another set of parameters and possible patterns.

Furthermore, the cross-sectional profile need not remain constant along the whole sweep. As illustrated in Figure 2(c) and Figure 8(b,c), the profile has been reduced in those parts of the sweep that pass through the tightly linked center of the sculpture, and it has been enlarged in the outer parts of the lobes.

Combining such geometrical fine-tuning with the almost limitless possibilities of starting with a cable-knot with enhanced complexity, resulting from heavily twisting or braiding the cable strands, gives sculpture designers an inexhaustible set of possibilities.

Carlo H. Séquin, https://orcid.org/0000-0002-0060-2162

REFERENCES

[1] Fan, R.: Joining Interactive Graphics and Procedural Modeling for Precise Free-Form Designs, (EECS-2021-125) 2021. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-125.pdf

[2] Grant, B.: Encyclopedia of Rawhide and Leather Braiding, Cornell Maritime Press, Centreville, MD (1972).

[3] Graumont, R.; J. Hensel, J.: Encyclopedia of Knots and Fancy Rope Work, Cornell Maritime Press, (1939).

[4] Knot Table: http://katlas.org/wiki/The_Rolfsen_Knot_Table

[5] Séquin, C. H.; Chen, T.: Combining Procedural Modeling and Interactive Graphical Editing for the Design of Abstract Geometrical Sculptures, FASE_SMI, Geometry Summit, Vancouver BC, June 21, 2019. http://people.eecs.berkeley.edu/~sequin/PAPERS/2019_FASE_NOME.pdf

[6] Séquin, C. H.: The Beauty of Knots, Gathering for Gardner G4G9, Atlanta GA, March 26, 2010. http://people.eecs.berkeley.edu/~sequin/PAPERS/2010_G4G9_Knots.pdf

[7] Séquin, C. H.: Splitting Tori, Knots, and Möbius Bands, Bridges Conf. Proc., pp 211-218, Banff, Canada, July 31, 2005. http://people.eecs.berkeley.edu/~sequin/PAPERS/Banff05_SplitTori.pdf

[8] Séquin, C. H.; Larrucea, L.: Introducing the Möbius-Twisted Turk's Head Knot, Bridges Conf. Proc., 335-338, Baltimore, July 29-August 2, 2015. http://people.eecs.berkeley.edu/~sequin/PAPERS/2015_Bridges_Moebius-Knot.pdf

[9] Smith, J.: SLIDE design environment, On-line, 2003. https://people.eecs.berkeley.edu/~ug/slide/