Modular Construction of Symmetrical Knots

C. H. Séquin, W. Brandon, J. Liu
CS Division, University of California, Berkeley

Abstract

The goal is to construct highly symmetrical models of mathematical knots from a single modular component that can be fabricated easily on inexpensive 3D printers or by injection molding. The "elbow" module, or *Λ-module,* is a piece of tubing that bends through a fixed given angle. To keep control over the torsional angle between subsequent modules, the tube cross section is not a circle but a regular polygon. We discuss possible optimization algorithms that allow the design of such knots with minimal amounts of deviation from the geometrical parameters imposed by the given modular component.

1. Mathematical Knots

Mathematical knots are fascinating objects. They are closed loops that are defined only at the topological level – by the crossing of different branches of the overall loop in 3D-space (Fig.1a), or by the connectivity of the complement space. A particular knot can take on many different geometrical shapes, and there is still no computer program that can unambiguously prove that two different looking versions of the same mathematical knot are indeed the same.

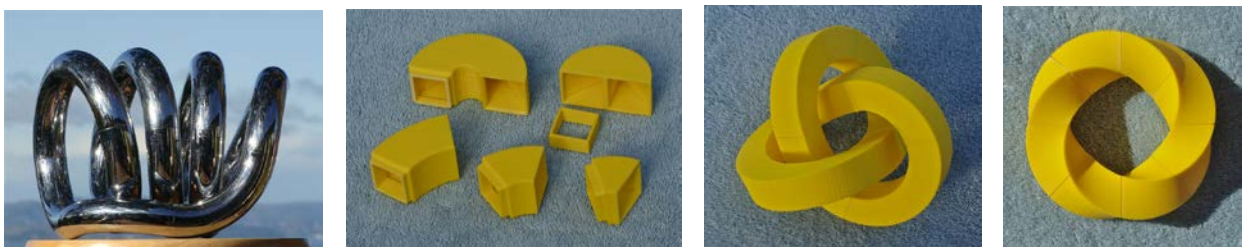
(a) (b) (c) (d) (e)

Figure 1: *Knot 10_{19}; (b) knot construction from tubular elements [9]; (c) Zawitz sculpture [11]; (d) bronze cast of "Torus knot 5,3" [4]; (e) "Music of the Sphere" (Trefoil or knot 3_1) [1].*

Physical realizations of mathematical knots, assembled from tubular elements (Fig.1b,c) or cast in metal (Fig.1d,e) can also result in artistically pleasing objects. Some large-scale knot sculptures cast in bronze (Fig.1e) [1] are expensive and take a lead-time of many months from design to the time when one can enjoy them. Thus, for quick experimentation and exploration of mathematical knots, it is desirable to have a set of snap-together parts, with which one can quickly construct physical models of arbitrary mathematical knots. Project LEGO-Knots [5] used some generic "snap-together" parts [6], supplemented with a few custom-made parts. A small set of different components (Fig.2b) – inspired by the LEGO system – has been designed to form most of the geometrical features found in sculptural knot models (Fig.2c). Different modules have different lengths and bend through different angles. The cross section of these tubular elements is not just a circle, but is a square or an equilateral triangle; this allows more interesting sculptural models to be constructed. These prismatic cross sections make it possible to form twisted structures (Fig.2d); for this reason, different modules come with different amounts of twist. However, often an additional custom-part had to be designed to guarantee a smooth closure of a particular knot.

This paper focusses on a different approach that relies on just a single generic module from which all possible knots are supposed to be constructed. One such system is represented by the "Museum Tangles" by Richard Zawitz (Fig.2a) [12]. They are composed of 18 tubular elements that bend through 90 degrees.

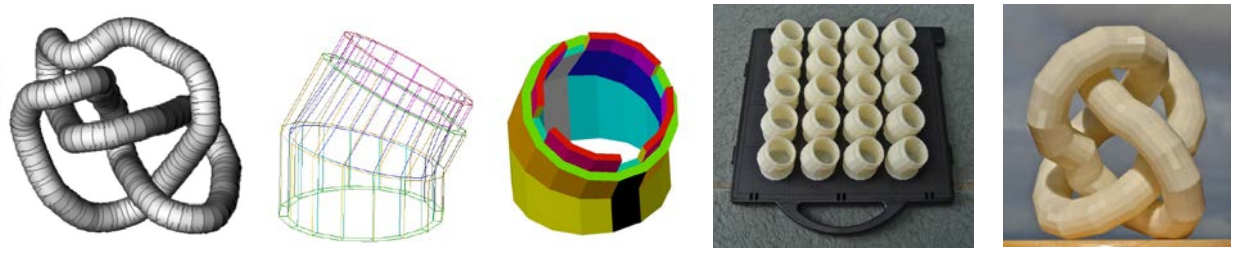
As packaged, the *Tangles* form a single un-knotted loop, which mathematicians call the *Un-knot* or the *Trivial knot*. Some of these tangles can be pried apart, and several of them can then be snapped together to form more complex, truly knotted structures. Another system has been proposed by Zawidzki et al. [10]. Their basic module is a rather short "wedge" that bends only a little bit. This makes it easy to form "organic" looking worms of possibly high knottedness (Fig.3a). Their proposal does not discuss how one might guarantee perfect closure of the loop or how to achieve some desired overall symmetry.



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

Figure 2: *(a) Zawitz "Museum Tangle" [12]; (b) LEGO Knot modules; (c) trefoil; (d) twisted loop.*

This paper focuses on knot model construction from just a single module that could readily be mass-produced by injection molding, but which can also be easily be fabricated on a low-end 3D-printer (Fig.3d), since it needs no supporting scaffolding, which then must be removed manually. In our current project, we want to form knots of high symmetry with far fewer modules than what would be needed by the *Pipe-Z System* [10] (Fig.3a). After some experiments, we have focused on an "elbow" module, which we also call the *Λ–Module*, consisting of two short tubular segments joined at an angle of 30° (Fig.3b,c). With a bending angle, β, of only 30°, rather than 90°, we are able to construct smoothly bending knot models without excessive meandering wiggles. With only this single *Λ–Module* available, the resulting, closed, possibly knotted, tubular loop is completely specified by the torsional angles (the rotation around the cylinder axis) by which two adjoining modules fit together. To allow an artist to follow precisely the given design information, we have designed our *Λ–Module*, with a cross section in the shape of a regular 16-gon, rather than a circle, which then offers 16 discrete torsion angles, separated by 22.5°. The value 16 is large enough to offer sufficient flexibility in constructing pleasing looking knots, yet coarse enough, so that a designer can readily chose the desired torsion angle from the 16 "clickable", "legal" rotational positions. The resulting component is shown in Figure 3c. Its outer tube diameter is 34mm and the axis length of each leg is 15mm. The tightest toroidal loop, composed of 12 modules, has an inner tunnel with a diameter of 70mm.



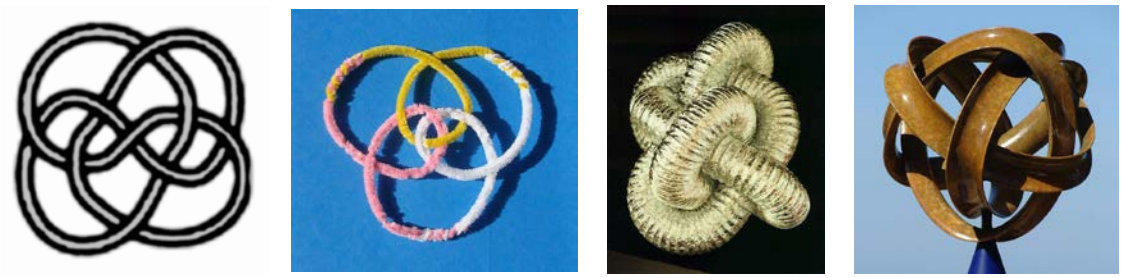
(a) (b) (c) (d) (e)

Figure 3: *(a) Figure-8 knot formed by the "Pipe-Z" system [10]; (b,c) our new "Λ-Module"; (d) 3D-prints of 20 Λ-Modules; (e) Figure-8 knot formed from 40 Λ-Modules.*

2. High-Level Design Issues

A knot specification like *knot* 3_1 (a trefoil) or *knot* 4_1 (the figure-8 knot) just defines the knot topology, but says nothing about the geometry of a particular knot realization. The users must decide what kind of knot model they would like to obtain; whether they want a relatively flat, "2.5-dimensional" configuration that resembles the diagrams (Fig.4a,b) in the knot tables [3] or whether they prefer a more space-filling,

"spherical" realization, which might form the basis for a design of a constructivist 3D sculpture (Fig.4c,d). The user must also specify explicitly the degree of geometrical symmetry they would like to obtain. The diagrams in the knot tables often do not show the highest degree of symmetry that is inherent in a particular knot. As an example, the *Chinese Button Knot* (*knot* 9_{40}) can be realized with dihedral (D_3)-symmetry of order 6 (Fig.4b,c,d), but the standard depiction (Fig.4a) gives no hint of this. The user must also indicate how loose or how tight a representation is desired.

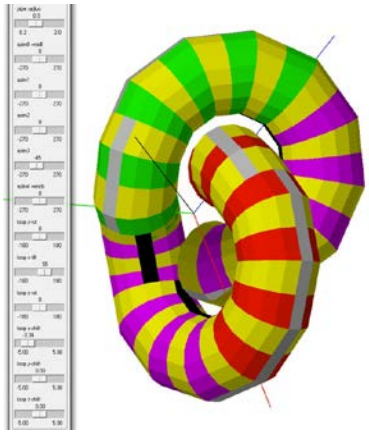


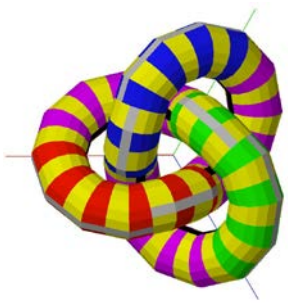
(a) (b) (c) (d)

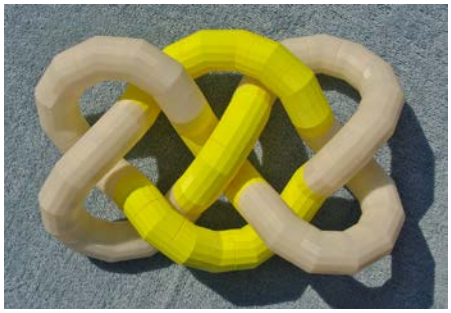
Figure 4: *Chinese Button Knot: (a) diagram for* 9_{40} *in Knot table; (b) another layout of this knot; (c) a tight 3D configuration; (d) an artistic realization of this knot. (Artwork by C. H. Séquin).*

We have created a couple of experimental CAD tools to assist a designer in realizing a desired knot based on a single modular component. In our CAD systems, the *Λ-module* is specified by its *bending angle*, β, (deviation from a straight line, in degrees), the *leg-length*, *L*, of the two tubular segments emerging from the central node of the module, the *tube-diameter*, *TD*, of the two legs, and the *number of discrete torsional angles* allowed, (e.g., defined by the 16-gonal cross section at the tube joints).

Our first CAD tool to design such modular knots was an interactive graphics tool implemented with Berkeley SLIDE [8], a graphic modeling system that allows parametric, procedural specifications. The designer sets all independent torsion angle values with a corresponding number of interactive sliders (grey field on the left). The modeling system then constructs a corresponding branch by assembling consecutive modules with the specified torsion angles. If the overall symmetry is of type D_n, then the constructed branch will already have C_2 symmetry. A set of *n* copies of this branch are then placed on *n* appropriately spaced, radial C_2 symmetry axes, where the designer can move these branches radially along those axes and rotate the branches around them in order to obtain the best match at the joints where two of these branches come together. Knots with only five or six independent torsion angle values can typically be found in a couple of hours of trial and error experimentation [6]. Examples of knots that have been constructed with this interactive graphics tool are shown in Figure 5.







(a) Trefoil, 33 modules (b) Cinquefoil, 50 modules (c) Knot 7_4, 64 modules

Figure 5*: Successful modular knot constructions.*

More complex knots, where a larger number of torsion angles need to be set, require more assistance from the CAD tool. The focus of this paper is the development of an optimization system that assists the user in finding a good set of torsion angles for all the joints to produce a knot model (Fig.3e) close to what the user may have had in mind, when imagining the knot being formed from a flexible hose or drier duct (Fig.4c).

3. Looking for Optimal Solutions

The users describe the desired knot they aim for with a polyline or a B-spline embedded in 3D space. The number of vertices on the poly-line, or the number of sampling points on the B-spline, indicate the number of modules the user plans to use for the knot construction. A first adjustment is a uniform scaling operation that assures that all the distances at skewed branch crossings exceed the tube diameter, (*TD=34mm)*, of the available module. At the proper scale, the total length of the knot curve is then evaluated. Dividing this length by (2*L*=30mm) yields a lower bound for the number of modules that could realize this particular knot geometry. Since we want the final knot to exhibit maximal symmetry, we choose the number of modules to be a multiple of *n* for a knot with C_n or D_n symmetry. The *n* C_2-symmetry axes must then pass through some module centers or through the joints between two modules.

It is useful to do one more test in this initial planning stage: All the bending angles between pairs of adjacent segments of the sampled B-spline curve are checked. If any of these bending angles exceed some upper bound, e.g., 2*30°, then the designer will be advised to make the starting curve "rounder."

We now let the vertices of the polyline represent the center points of our given modules. In a physically realizable solution, these vertices must be spaced from their neighbors by a distance of 2*L*, given by the length, (*L=15mm)*, of the legs of the tubular module. Using greedy optimization, based on gradient descent, the distances between neighboring vertices in the poly-line can be pushed towards 2*L*, and the bending angle, β, (the deviation from a straight-line tube) at each vertex is forced to approach 30°. If the desired knot curve has some relatively straight branches, then in these branches, the solution will have to introduce some zig-zag or helical sweep paths, so that the angles between subsequent segments in the poly-line are always 150°.

If our *Λ-Module* had a circular cross section then a greedy optimization with only two types of error terms can often deliver a solution with no local deviations from the ideal geometrical characteristics. One of the two terms is a *total length error*, *TLE,* obtained by summing the squares of the differences of any individual segment length from the ideal length of 2*L=30mm*:

Total Length Error: $TLE = \sum (length_i - 2L)^2$.

The other error term is the deviations of the actual bending angles at each vertex (the angle formed by three consecutive points on the polyline) from the specified value β (β=30° in our examples):

Total Bending Error: $TBE = \sum (bend\text{-}angle_i - \beta)^2$.

There is some ambiguity as to how these two error terms, *TLE* and *TBE*, should be combined into a single overall penalty value that then can be minimized. What are equally "bad" (noticeable) deviations in these two categories? By physically manipulating assemblies of our *Λ-Module*, we determined that a practical error threshold for visibility of any geometrical deviation from a perfect assembly is about one degree for the bending angles and half a millimeter for module separations. Such deviations would be barely noticeable for our *Λ-Modules* fabricated on inexpensive 3D-printers. We assign both of these error thresholds a penalty value of 1.0. The assigned penalties are increased quadratically with increasing deviation. All the local penalties are summed into a global penalty value, which is then divided by the number of error terms being evaluated; in the optimization above, this is equal to twice the number of modules. This results in an averaged penalty (error) measure that is independent of knot complexity.

Given that we want to find a perfectly symmetrical solution with C_n or D_n symmetry, the number of individual torsion angles that need to be determined is reduced by a factor of *n*, or by 2*n*, respectively. However, this also adds more constraints and reduces the number of possible solutions. These symmetry

constraints are captured in a hierarchical description of the knot poly-line, given by n or $2n$ identical pieces; therefore, the computational effort of finding the optimal vertex positions is also reduced by the same factor.

But, a physically realizable solution may not always exist. If we wanted to make a toroidal loop with only eleven components, then this cannot be realized with a bending angle β of 30°. In this case the optimization would result in the same slanted gap between all 11 modules, and the final penalty would be:

Penalty of 11-module torus = $11 * (30/11)^2 / 22 = 3.719$ (penalty units).

We also need to check, whether the final "wiggly" knot axis has no skewed branch crossings with a separation of less than the tube diameter, *TD*, of the module. It is even possible that the resulting polyline now has a different knotted-ness, because the optimization process may have caused some inadvertent branch crossings that could have changed the knot type. Both these concerns can be addressed during the optimization process itself by introducing an additional error term that increases dramatically as two branch elements approach each other to a distance of *TD*.

4. Discrete Torsion Angles

Any given solution coming from the above optimization process, will be presented in the form of a list of torsion angles. These angles may take on arbitrary values. But, how would a user realize a torsion angle of, say, 142.7° between two subsequent nodules? This is why we have given our module a cross section of a regular 16-gon, with 16 precisely defined torsion angles to choose from. A 16-gonal cross section can be robustly realized on inexpensive printers, and it also is reasonably convenient when assembling a particular knot model, giving just the list of torsion angles. To make it even easier to apply the correct torsion angle when joining together two *Λ–Modules*, the flanges needed to make the tubular joints are notched to give a ready indication of some particular torsion angles, such as 0°, ±90°, or 180° (Fig.3b).

Suppose we want to construct a trefoil knot with perfect D_3-symmetry from 30 *Λ-Modules* with 16 "legal" torsion angles. We may first form a branch of 10 modules by adding 5 modules on either side of a central joint, maintaining perfect C_2-symmetry. This symmetrical branch is entirely specified by 5 discrete torsion angles, 4 of which are applied pairwise symmetrically on either side of the central joint. The question then is, what set of 5 angles will define a *branch*, where three copies of it can be assembled into a trefoil knot with a low enough error value at the three joints between those *branches*; the penalties at all other joints are zero by construction. This now changes the optimization domain from a smooth, contiguous solution space into a discrete set of grid points, where gradient descent can no longer be used.

However, this approach, which was inherent in our first graphical modeling tool [6], is not optimal! In an actual, physical realization, errors will naturally distribute themselves over all the 30 joints in the knot model. At every joint, there could be a small axial misalignment and a little bit of torsional deviation from one of the 16 admissible angles – hopefully rendering all of these deviations almost invisible.

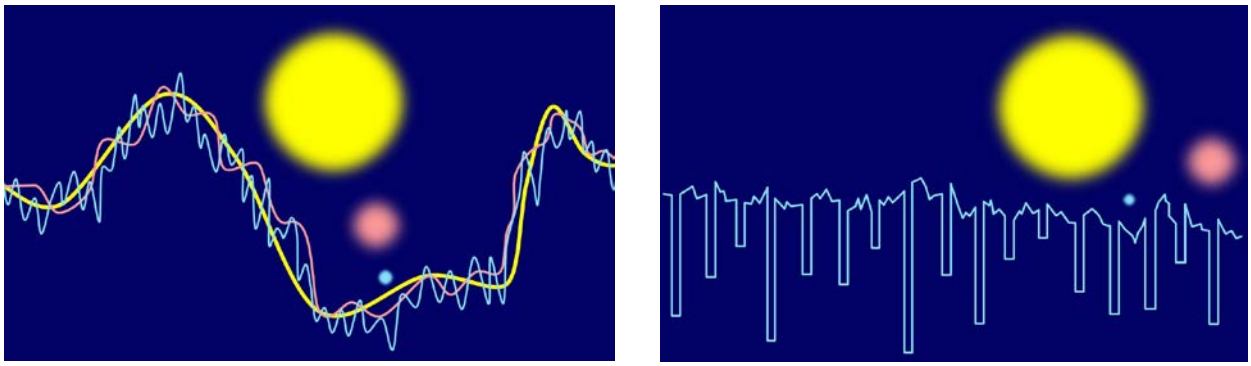
Thus, to evaluate the true quality of any discrete choice of torsion angles, we need to perform an additional gradient descent optimization. In this process, the actual torsion angles will deviate from their ideal discrete values, and additional small errors will build up in the bending angles and module separations in order to reduce the errors at the three branch junctions. In the overall penalty function, we equate a 1° deviation in a torsion angle with a 1° deviation in bending angle. It is then among these optimized, "nearly discrete" solutions that we need to find the best possible candidates.

5. (Unsuccessful) Optimization in a Discrete Solution Space

Two often-used techniques in the domain of discrete optimization are *Genetic Algorithms* [2] and *Simulated Annealing* [7] Genetic Algorithms are particularly useful when the entity to be optimized has a limited set of discrete characteristics that can be optimized more or less independently of one another. Examples might be the eyes, ears, and noses in an animal, which together contribute to an animal's survivability. In our knots, all torsion angles are equally important, and they all work "in concert" with one another. Except for random "mutations," Genetic Algorithms do not offer a useful approach for this particular design problem.

Simulated Annealing is a technique that is widely used in the placement and routing of integrated circuits (IC). A given set of components, and all the required wire connections between them, need to be placed as tightly as possible on a small IC chip. The algorithm tries to improve the current solution, by re-locating and re-orienting a few components at a time. We cannot expect that there always exists a possible swap of a few components that will improve the current solution. Therefore, the algorithm, particularly in the early phases of the optimization, will accept moves that lead to slightly worse results, hoping that from this new position, there are other moves that will result in an overall improvement. The accepted amount of additional error in a particular move is equated with a "temperature" of the system, where higher temperatures are synonymous with larger uncertainties of what constitutes a "good" move. This generic optimization paradigm seemed most suited for our problem domain, where we try to make the best possible changes to our discrete "legal" torsion angles. However, almost all "small" digital moves, i.e., changing one torsion angle by one discrete "click," resulted in a large increase in the overall error value, and we have not been able to find a strategy that would guide us reliably and efficiently towards "the best" solution.

For many types of problems, such as placement and routing of components on an IC chip, Simulated Annealing [7] works well, because there is some correlation between similar moves and the benefit they yield with respect to the overall error function. This corresponds to a solution space (depicted for just one dimension) that looks like Figure 6a. Rolling around a "big fuzzy ball" (corresponding to a high temperature) in this landscape would let it find the deepest large (yellow) valley. Then, as the temperature is reduced gradually, the diameter of the ball shrinks, and it will start to explore the finer (pink) structure found in this large valley. Eventually the ball will be small enough (blue) to fall into one of the little crannies at the bottom of the big yellow basin.



(a) (b)

Figure 6: *Solution space profiles: (a) a well-behaved function; (b) difficult "golf-course" profile.*

To understand whether the solution space of our design problem has the appropriate structure for Simulated Annealing to work, we have studied the solution spaces of trefoil knots assembled from 30 or 33 *Λ-Modules.* For this simple, highly symmetrical knot with only five independently selectable torsion angles, we could perform an exhaustive search. For each combination of discrete torsion angles, we performed a quick analysis to see whether it would have a chance to produce the desired trefoil knot; this eliminated most of the one million possible combinations. The optimization process was then run only for these promising, "constructible" combinations (about 4300). This allowed us to study the relationships between "digital neighbors", i.e., knots that differ by only one or two small changes in the discrete torsion angles.

The results were rather surprising and illuminating: There are only very few digital neighbor pairs where both of them have low error values. This indicates that the structure of the solution space looks more like Figure 6b, which could be characterizes as a "devil's golf course" – a rugged surface with isolated pockets of different depth. In such a solution space, there is no good strategy to move from one good pocket to another one. One good solution offers no indication that there might be other good solutions nearby (in the digital sense). Thus, searching for the best solution in the purely digital domain is <u>not</u> a good approach.

6. A More Promising Approach

Recently we have started to pursue a different approach, in which we retain more information from the contiguous solution space of the initial polyline with arbitrary torsion angles. We now investigate how this information can guide us to digital combinations that are more likely to deliver good solutions.

One approach might be to let the designer make small changes to the initial knot curve, perhaps by changing some parameters of the defining B-spline. Perhaps, the initial B-spline representations for the knot was too smooth. The final polyline of any realized modular knot has many wiggles and undulations, since it must bend through 30° at every module center. In a lucky situation, these undulations may be exploited to route one branch nicely around an adjacent branch. But, to find such opportunities, the initial knot curve would have to rely on several more control points, and it would be difficult for the designers to add such details into their initial knot geometry specification. Thus, we found no good strategy to guide the designers in how to make small random changes to the initial knot curve, hoping to stumble across the right kind of undulations that would then result in a modular knot with a low overall error value.

Another option might be to shift the initial sampling points along the B-spline. But, an arbitrary shift in the sampling positions would break the strict symmetry of the desired knot geometry, and we disallow this operation. However, there is a second sampling option that maintains the chosen symmetry: Any possible C_2-symmetry axis must either pass through the center of a module or through the junction between two of them. Thus, we should evaluate the original knot curve with twice as many sample points as the number of modules we plan to use and then separately evaluate and optimize the polylines resulting from the even samples as well as the from the odd numbered ones.

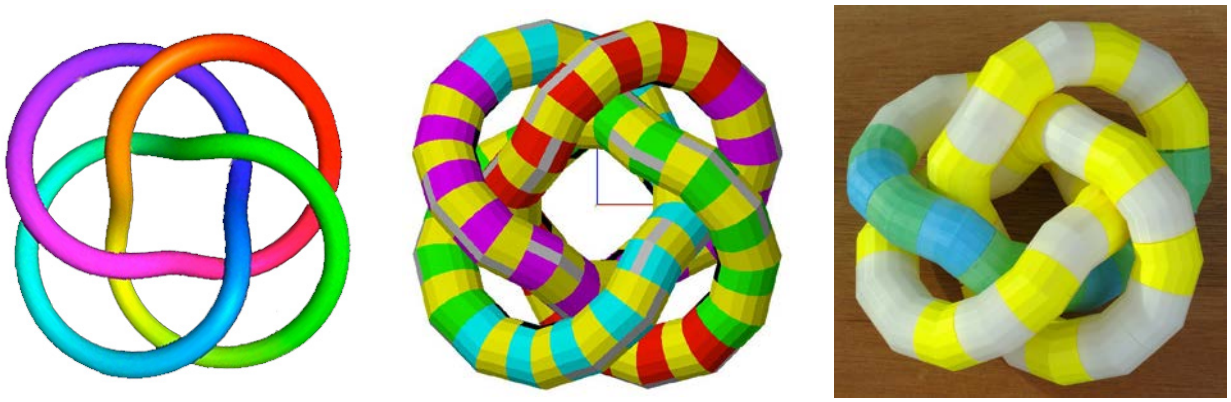
In the purely digital realm, a knot design with A individually choosable torsion angles, that maintain the specified overall symmetry, offers 3^A digital neighbors where individual angles differ by no more than ±1 "clicks" from the initial digital combination. For A=10 this gives us 59,049 options to explore, – with no indication which ones of these may be more promising to offer a low overall penalty.

If we remember how any particular discrete torsion angle was chosen by rounding the initial analog angle, we only have to focus on two options. Each analog angle has two closest legal values, which can be reached by rounding either up or down. Thus, we may limit ourselves to evaluate only 2^A combinations. For A=10 this is 1024 and thus equivalent to a speed-up of more than 50 times!

But, we can be even more selective, since we have some indication as to which options may be more promising. If one of the torsion angles of the initial polyline is close to half-way between its two nearest "legal" angles, might it not be likely that rounding to the other, slightly farther away legal torsion angle may yield a solution just as good, or perhaps even better? Thus, we evaluate every analog torsion angle for its potential to be rounded in the alternate direction. Any rounding operation is associated with a local penalty equal to the square of the angle change needed to reach a "legal" value. With A independent torsion angles to be set, we might evaluate all 2^A combinations of different rounding directions for their initial costs associated with rounding in one or the other direction. We may then choose the 100 "least expensive" combinations and perform gradient descent into the corresponding local optimum. Among the solutions with low enough overall penalties, the designer may then choose one, either because it has the lowest penalty value or because it is most aesthetically pleasing, – perhaps because it has the fewest gratuitous undulations, or the most uniform clearances between skewed crossings of adjacent branches.

Based on some limited early results, it seems that small, tight knots offer the most challenges. Larger, more complex knots, while taking more computation time for each optimization run, actually are more often giving good results with low average penalty. A larger number of modules offers more junctions, where a little deviation from the ideal geometry can be introduced to achieve the overall proper closure of the knot loop. Physical implementations with more modules result in more flexible assemblies, which can more easily be deformed to get the last junction to match up properly.

Figure 7 shows the example of a (4,3) torus knot with D_4 symmetry. It is composed of 48 *Λ-Modules*. The six torsion angles in one of the replicated 6-module branches are: 0°, 90°, -67.5°, 0°, -45°, 22.5°, respectively.



(a) (b) (c)

Figure 7: *(4,3)-torus-knot: (a) knot curve; (b) modular CAD model; (c) physical realization.*

7. Conclusions

At the moment, this is work in progress, and the three authors are the sole users of this emerging system. A few different optimization approaches have been tried and rejected. We are now focusing on a more promising approach that seems to yield more readily solutions with low error values. A broader evaluation of its performance statistics has been started. Once all algorithmic issues have been fully resolved, our next goal is to provide this system with a decent user interface, so that designers without any computer-science training can easily design modular mathematical links and knots with high complexity and high symmetry.

References

[1] B. Collins, D. Lynn, S. Reinmuth, C. Séquin. "Realization of Two New Large-Scale Sculptures." FASE 2012, – http://people.eecs.berkeley.edu/~sequin/PAPERS/2012_FASE_NewSculptures.pdf

[2] Genetic Algorithms. – https://en.wikipedia.org/wiki/Genetic_algorithm

[3] Rolfsen Knot Table. – http://katlas.org/wiki/The_Rolfsen_Knot_Table

[4] C. H. Séquin. "Torus-Knot_5,3." -- http://people.eecs.berkeley.edu/~sequin/ART/Science_Sculpture/

[5] C. H. Séquin. "Lego Knots." *Bridges Conf. Proc*. pp 261–270 (2014) – http://archive.bridgesmathart.org/2014/bridges2014-261.html

[6] C. H. Séquin. "Reconfigurable Snap-Together Sculpting." Computer-Aided Design and Applications, (2015). – http://people.eecs.berkeley.edu/~sequin/PAPERS/2015_CAD-A_SnapSculpt.pdf

[7] Simulated Annealing. – https://en.wikipedia.org/wiki/Simulated_annealing

[8] J. Smith. "SLIDE design environment." (2003). – http://www.cs.berkeley.edu/~ug/slide/

[9] F. Smullin. "Analytic Constructivism: Computer-Aided Design and Construction of Tubular Sculptures." Luncheon Presentation, 18th Desigri Automation Conference, Nashville, TN, (June 30, 1981).

[10] M. Zawidzki, K. Nishinari. "Modular Pipe-Z System for 3D Knots." Jour. Geometry and Graphics, pp 81-87, Vol.17 (2013).

[11] R. X. Zawitz. "Tangle Creations." – https://www.tanglecreations.com/pages/richard-x-zawitz

[12] R. X. Zawitz. "Museum Tangle." – https://www.tanglecreations.com/search?type=product&q=museum+tangle