Spline Knots and Their Control Polygons With Differing Knottedness

Carlo H. Séquin

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-152
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-152.html

November 1, 2009

Acknowledgement

This work is being supported in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF award #CCR-0225610 (ITR)).

Spline Knots and Their Control Polygons With Differing Knottedness

Carlo H. Séquin
CS Division, University of California, Berkeley
E-mail: sequin@cs.berkeley.edu

Abstract

Spline knots based on Bézier curves or B-splines can exhibit a knot type that is different from that exhibited by its control polygon, i.e., the spline and its control polygon are not ambient isotopic. By forming composite knots from suitably designed building blocks the difference in knottedness of the two 1-manifolds can be made arbitrarily large.

1. Introduction

To mathematicians, a knot is a closed, non-self-intersecting curve with a specific embedding in Euclidean 3D-space $\mathbf{R}^3$. When such a knot is described as a B-spline or as a composite Bézier curve for analysis or for detailed geometrical manipulation, it is highly convenient and efficient, if most of the calculations and operations can be done using just the linear segments of the control polygon. However there may be substantial topological differences between the curve itself and its control polygon. We know that Bézier curves and B-splines are variation diminishing, *i.e.*, any straight line in 2D, or any plane in 3D, cannot intersect the curve more often than it does the control polygon. But this still permits the control polygon to have self-intersections or loops that are not exhibited by the spline curve itself. On the other hand, as the control polygon is subdivided into multiple consecutive segments, it moves closer to the spline curve. Any non-self-intersecting Bézier curve with regular parameterization will, *after* sufficiently many subdivisions, have a control polygon that is also non-selfintersecting [5]. However, two non-self-intersecting, closed curves can still have very different embeddings within $\mathbf{R}^3$. So, it is of interest to consider when a spline curve and its control polygon have the same embedding. This is the purview of knot theory, where non-self-intersecting, closed curves are classified and tabulated according to their embeddings [1]. The topology of the space surrounding the knot, in particular the relation of *ambient isotopy*, is used to partition knots into equivalent subclasses. Ambient isotopy is also the appropriate topological equivalence relation for many practical applications in geometric modeling, visualization, and animation [3].

For a rich class of composite Bézier curves, ambient isotopic equivalence with sufficiently subdivided control polygons has been shown by Moore et al [4]. They have shown that for non-self-intersecting Bézier curves with a regular parametrization there exists a positive integer m such that the control polygons of the m-th or higher subdivision of the curve are all ambient isotopic to the Bézier curve. However, that work did not provide any explicit examples where a composite Bézier curve and its control polygon were not ambient isotopic even while they were both non-self-intersecting. This report constructs and analyses a few illuminating examples.

Three basic constructions are shown: First, we build unknots with control polygons of arbitrary high knottedness (Section 3). Then we construct unknotted control polygons (Section 4) that generate either arbitrarily knotted composite splines (Fig.6), or, alternatively, two separate spline loops with an arbitrarily high linking number [1], even though the corresponding control polygons do not interlink at all (Fig.5).

2. A Simple Unknot With a Knotted Control Polygon

We first construct an example involving a composite of two simple cubic Bézier curves. Curve A (yellow) has the (red) control polygon P_0, P_1, P_2, P_3, and Curve B (blue) has the (green) control polygon P3, P4, P5, P0, with the respective coordinate values:

P_0: (-6, -6, 12), P_1: (4, 1, -1), P_2: (-4, 1, 1), P_3: (6, -6, -12), P_4: (1, 2, 4), P_5: (-1, 2, -4).

Figure 1a shows these curves parallel projected onto the *x*-*y*-plane.

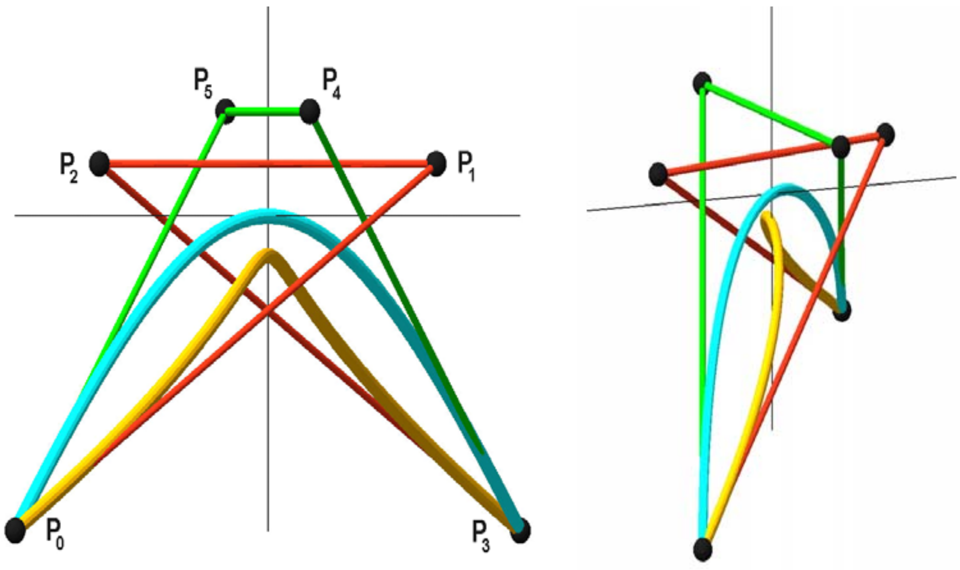

Figure 1: Two cubic Bézier curves with their control polygons. (a) The composite of the two curves (blue and yellow) form the unknot, while their control polygons (red and green) form a trefoil knot. (b) A view of this 3D configuration rotated around the y-axis.

We now analyze the resulting knots formed by the composite Bézier curve and their control polygons. A paper discussing this configuration with more mathematical rigor has been submitted [2]. In the projection of Figure 1a, the Bézier curves themselves (yellow and blue) form a closed loop with no intersections; thus this is clearly the unknot or tivial knot.

The projection of the knot formed by the control polygon (Fig.1a) exhibits 5 crossings. But the two crossings formed by line segment P_0,P_1 can readily be eliminated with a Reidemeister move of Type II [1] by stretching the line P_0,P_1 so that it loops around the outside of vertex P_3. This leaves a knot projection with three alternating (over-under-over…) crossings. The fact that this sequence is indeed alternating can be visualized by rotating the whole configuration around the *y*-axis (Fig.1b). It also can be proved rigorously by calculating the linearly interpolated *x*- and *y*-values for the three intersections, and by interpolating correspondingly the *z*-values of the line segments involved. Once we have convinced ourselves that the control polygon forms a closed loop with three alternating crossings, we know that this is a trefoil knot, listed as Knot 3_1 in the knot tables [1].

Some readers may be bothered by the lack of smooth C^2-continuity at the joints of the two Bézier curves. Of course, topological analysis only cares about connectivity (C^0-continuity) and not about smoothness of the curves. But we can readily use the combined 6-stick control polygon to define a cubic B-spline. The result is shown in Figure 2. The geometry of the control polygon has not changed at all, thus it still forms a trefoil. The B-spline curve exhibits a single crossing in the projection onto the *x*-*y*-plane shown in Figure 2a. The small loop associated with it can be removed with a Type I Reidemeister move [1], which un-twists the small loop next to this crossing. Thus we now have a perfectly smooth realization of the unknot with a control polygon that still forms a 3-crossing knot.

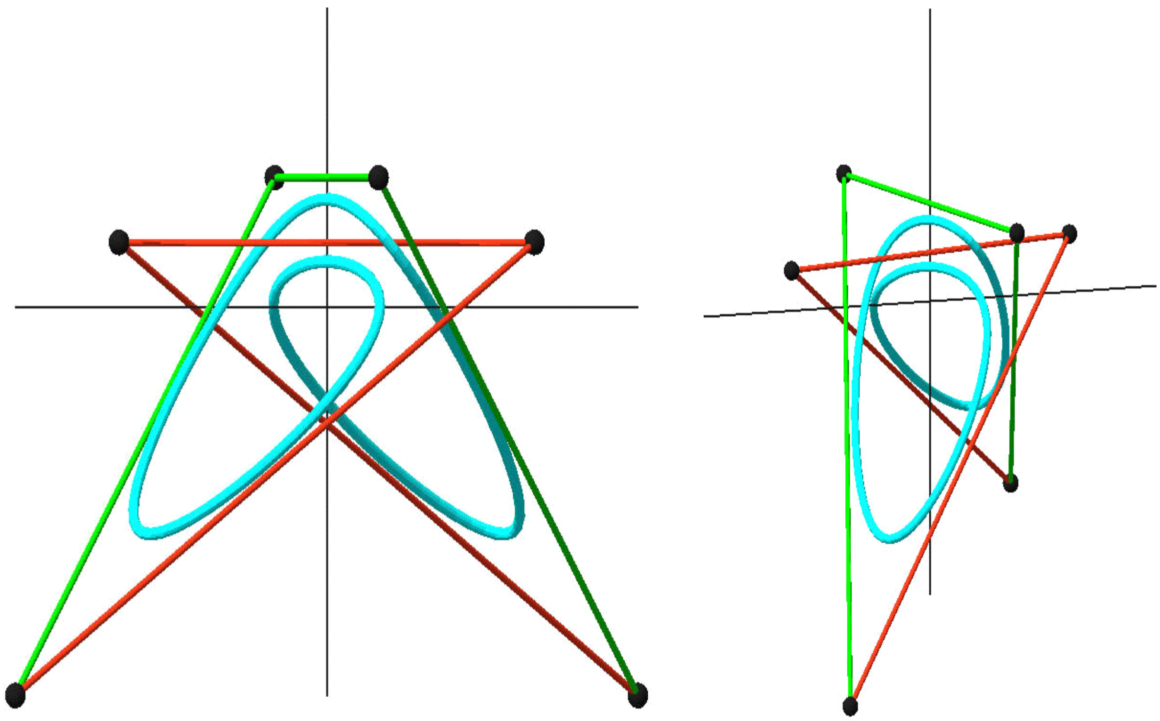

Figure 2: A C^2-continuous B-spline curve with its 6-segment control polygon.
(a) A projection into the x-y-plane. (b) A rotated view of this 3D configuration.

3. Control Polygons of Arbitrary Knottedness

We can now use this configuration to form an unknot with a control polygon of arbitrary knottedness [2]. For that purpose we split and separate the two Bézier curves and their control polygons from Figure 1 at point P_0 by a small amount that does not change the orientation of any of the crossings in this figure. Now we can connect *n* instances of this construct into a pin-wheel configuration as shown in Figure 3 for the case of $n = 6$. The concatenation of the Bézier curves is still a crossing-free loop and thus forms the unknot. At the same time, the concatenation of the control polygons forms a composite knot with a crossing number of $3n$. Thus we can construct a knot with an unlimited degree of knottedness by simply increasing *n*.

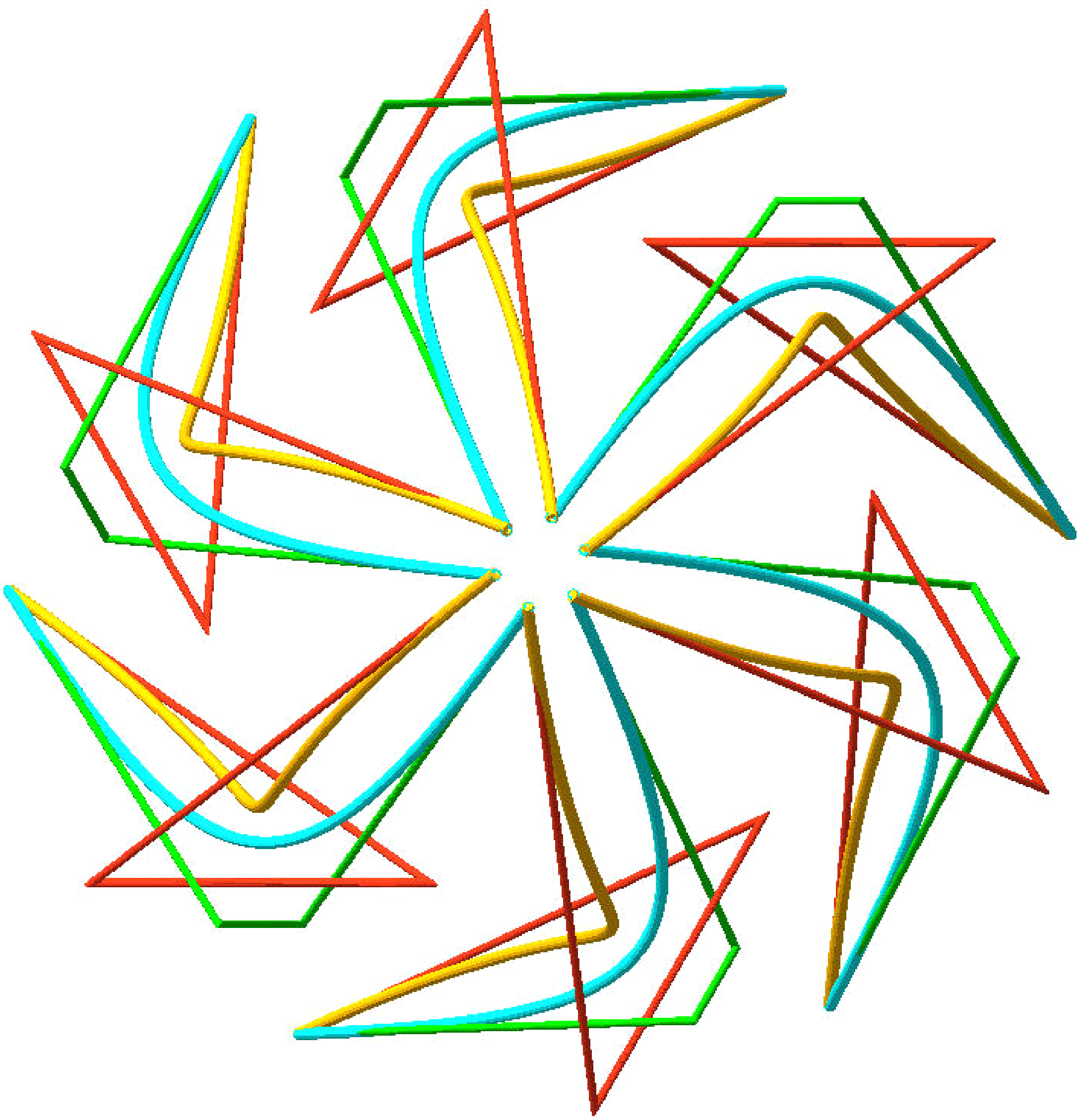

Figure 3: Composite of six building blocks from Figure 1.

4. Un-knotted Control Polygons

The above construction and analysis raises an intriguing complementary question: Can an un-knotted control polygon form a knotted spline curve? Our first intuitive reaction might be to answer this question in the negative. We know that B-splines and Bézier curves obey the *convex hull property*: The spline curve segments lie always within the convex hull of the associated control polygon. These splines are also *variation diminishing*, which means that the spline curve can never have more "wiggles" than the corresponding control polygon; or, stated more formally, any plane that cuts through the control polygon q times, can cut the spline at most q times. – However the interlinking of two curves is a more subtle property!

Figure 4a shows a simple cubic Bézier segment. Of course it lies totally inside the convex hull (thin green lines) of its control polygon. However, the two pink-colored areas denote regions where the spline curve is lying "outside" some of the control segments. This gives us the opportunity to entangle two curve segments without also entangling the two control polygons (Fig.4b). While the spline segments and their control polygons are nearly planar, the assembly of the two segments is made "very 3-dimensional" by placing two instances of Figure 4a in planes that are perpendicular to one another. With proper affine scaling we can arrange it that the four end-points of the curves fall onto the corners of a square. This yields a nice modular building block that allows us an easy construction of composite knots or links: We can place several such units around a regular n-sided prism with n square faces (Fig.5 and 6).

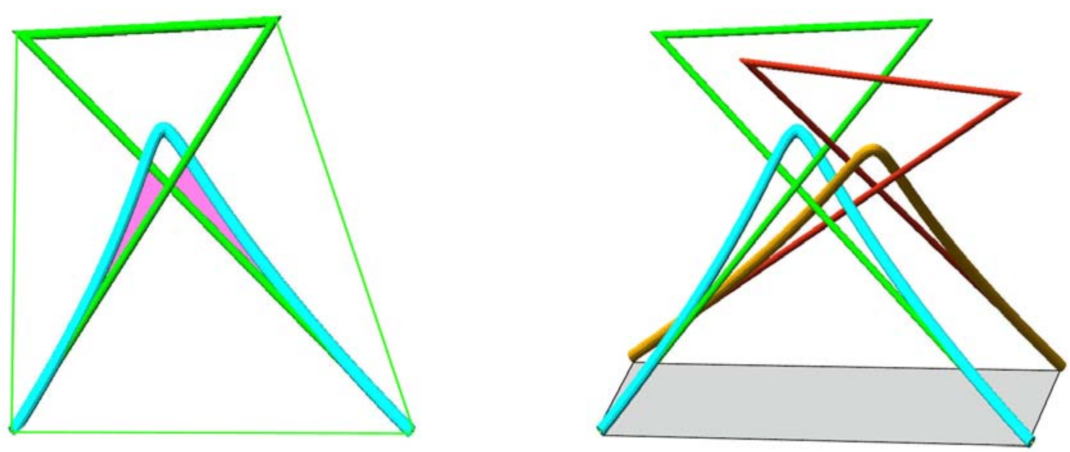

Figure 4: Cubic Bézier curves with their control polygons. (a) The (blue) Bezier curve is entirely inside the convex hull, but there are two regions where the curve is "outside" its control polygon. (b) Two such curve pieces can be "entangled" without having entangled control polygons.

In a first construction (Fig.5a) we place three building blocks as shown in Figure 4b around a 3-sided prism by simply rotating the three instances around the *z*-axis in increments of 120° and then fusing the ends of the curves (Fig.5b). Both the green and the red control polygon form separate unknots, and so do the yellow and blue spline curves. But while the two control polygon loops are not linked at all – the green-green junctions lie all above the red-red junctions (Fig.5c) – the spline loops form a linked configuration with 6 alternating crossings (Fig.5d).

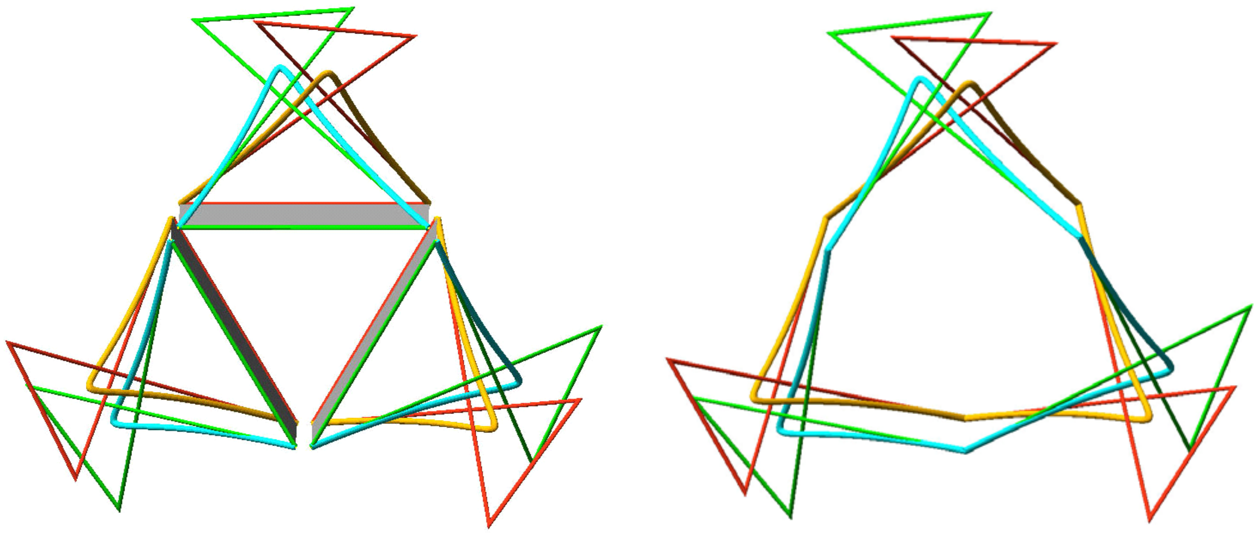

Figure 5: (a) Three of the constructs shown in Figure 4b are arranged around a 3-sided prism, (b) The same construction shown with the ends of the curves merged.

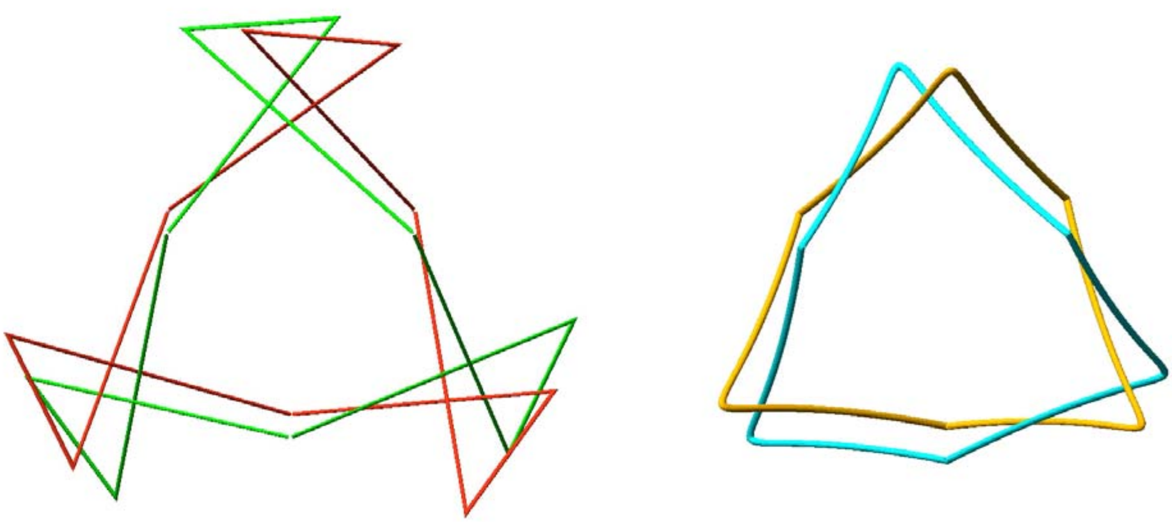

Figure 5 (cont.): (c) The three pairs of control polygons shown by themselves. (d) The resulting Link 6^2_1 composed of two interlinked unknots.

As a second example we build a non-trivial spline knot with an un-knotted control polygon. With this aim, we place four of the constructs of Figure 4b around a 4-sided prism (cube). But this time we rotate the bottom unit 90° around the *y*-axis, so that the partial loops formed by three red and three green control polygons are connected into a single loop. However, this loop is still the unknot. On the other hand, the spline segments, which are now also connected into a single loop, form the 8-crossing knot of type 8_1.

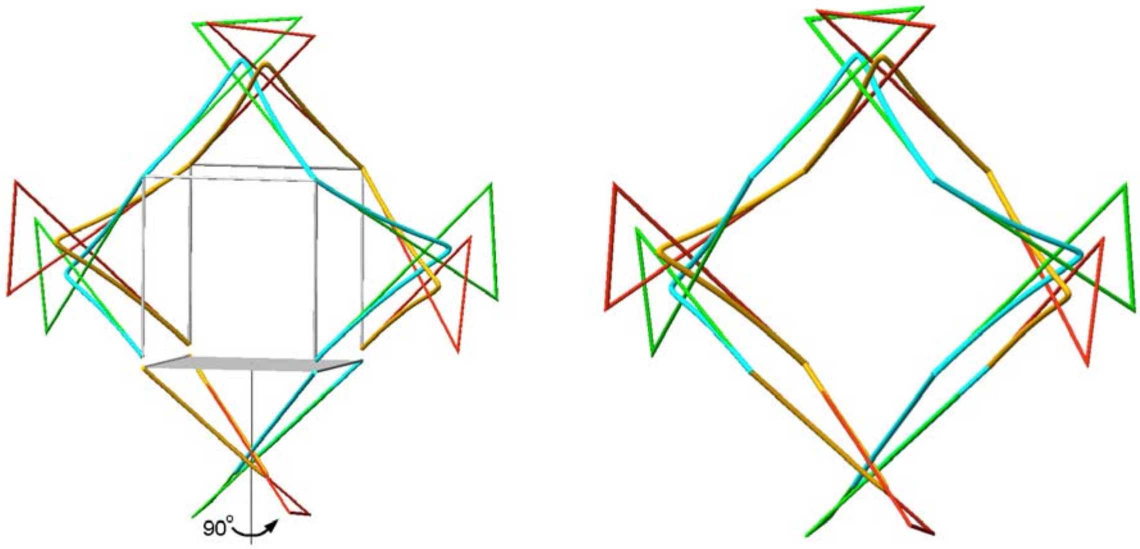

Figure 6: (a) Four of the constructs shown in Figure 4b are arranged around a 4-sided prism, with the bottom unit rotated by 90° around the y-axis. (b) Same arrangement with fused curve ends.

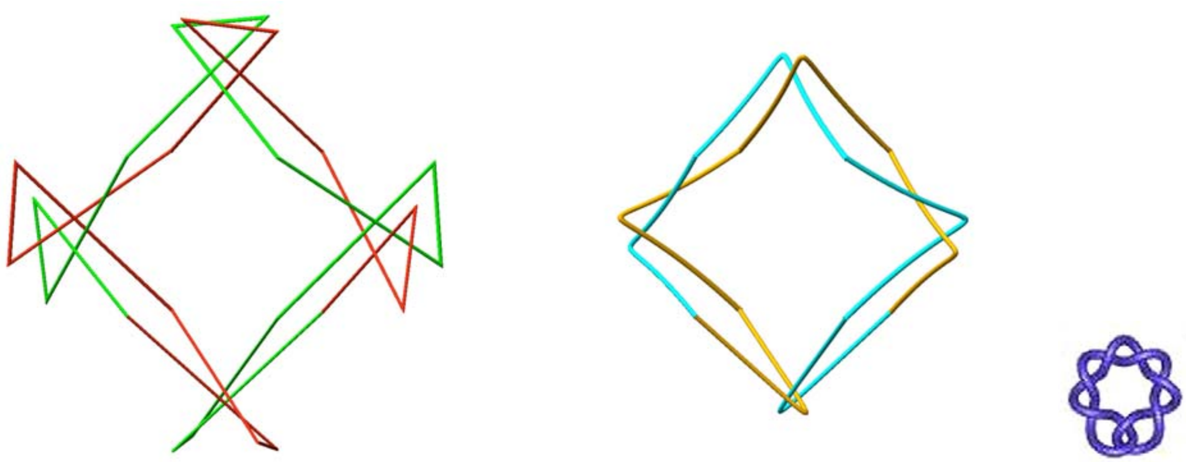

Figure 6 (cont.): (c) The four pairs of control polygons shown by themselves. (d) The resulting knot of type 8_1.

Acknowledgements

Stimulating technical discussions with Tom Peters are gratefully acknowledged.

This work is being supported in part by the Center for Hybrid and Embedded Software Systems (CHESS) at UC Berkeley, which receives support from the National Science Foundation (NSF award #CCR-0225610 (ITR)).

References

[1] C. Adams, *The Knot Book*, W. H. Freeman and Co, New York, 1994.

[2] J. Bisceglio, T. J. Peters, J. A. Roulier, C. H. Séquin, *UnknotsWith Highly Knotted Control Polygons*, submitted to CAGD, October 2009.

[3] K. E. Jordan, L. E. Miller, E. L. F. Moore, T. J. Peters, A. C. Russell, *Modeling time and topology for animation and visualization with examples on parametric geometry*, Theoretical Computer Science 405 (2008) 41–49.

[4] E. L. F. Moore, T. J. Peters, J. A. Roulier, *Preserving computational topology by subdivision of quadratic and cubic Bézier curves*, Computing 79 (2) (2007) 317–323.

[5] M. Neagu, E. Calcoen, B. Lacolle, *Bézier curves: Topological convergence of the control polygon*, in: T. Lyche, L. L. Schumaker (Eds.), *Mathematical Methods in CAGD*, Oslo 2000, Vanderbilt University Press, Nashville, 2000, pp. 347–354.