Regular 3D Polygonal Circuits of Constant Torsion

Tom Verhoeff *
Faculty of Mathematics and CS
Eindhoven University of Technology
Den Dolech 2
5612 AZ Eindhoven, Netherlands
Email: T.Verhoeff@tue.nl

Koos Verhoeff
Valkenswaard, Netherlands

Abstract

We explore a special class of regular 3D polygonal circuits, that is, of regular non-planar polygons. In these circuits, all segments (edges) have the same length, all corner angles are equal, and all torsion angles have the same (absolute) value. We also show some artwork based on these constant-torsion circuits.

1 Introduction

Consider the polygonal circuit[1] in Figure 1. Although not immediately obvious from the pictures, this is a highly regular 3D structure. It consists of 16 *equal-length segments*. The 16 *angles between adjacent segments* are all *equal* (viz. to 49.94°). We call these the **corner angles** or **joint angles**. Finally, the 16 *dihedral angles between adjacent corner-spanning planes* are *equal* (viz. to 90°). Figure 2 shows these planes explicitly. We call the latter angles the **torsion angles**, as explained further below.

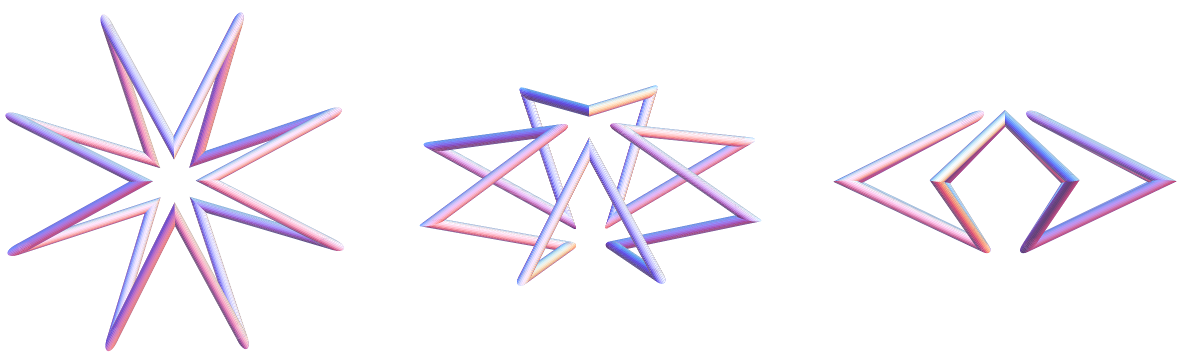


Figure 1: *Regular 3D polygonal circuit of constant torsion (top view, halfway view, side view)*

In this article, we explore 3D non-planar polygons[2] having constant segment lengths, constant corner angles, and constant torsion angles. We refer to these polygons briefly as **regular constant-torsion polygons**. Section 2 explains why these structures raised our interest. In Section 3, we precisely define this class of polygons, using 3D turtle graphics, and in Section 4 we provide some theoretical analysis results. We address the issue of finding such constant-torsion polygons in Section 5. In Section 6 we show some artwork that involves constant-torsion polygons. Section 7 concludes the article.

[1]The circuit's line segments are infinitely thin, but they have been thickened for the sake of visualization.
[2]Traditionally, a polygon is a planar figure; we also use the term polygon for arbitrary (possibly non-planar) polygonal circuits.

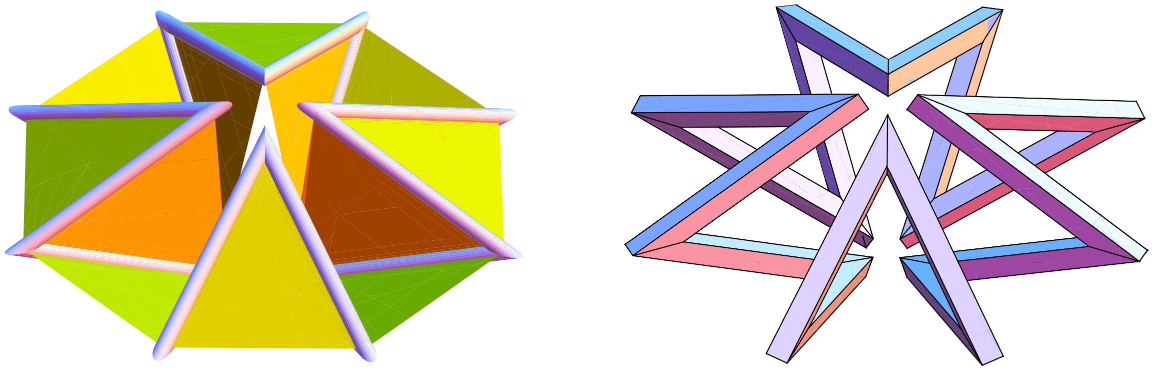


Figure 2: *Polygon of Figure 1 with corner-spanning planes (left) and mitered with a square beam (right)*

2 Motivation

In a polygonal circuit, the line segments can be thickened to beams that all have the same cross section and that have the segments as center line. It is most elegant when the edges of adjacent beams nicely match, yielding (regular) miter joints. Figure 2 (right) shows the polygonal circuit of Figure 1 with its line segments thickened to beams having a square cross section. Here, the beam edges match properly at all miter joints.

When a polygonal circuit is thickened to a beam, it is always possible to have beam edges match properly at all joints *but one*. Hence, open polygonal chains are not so interesting, because they pose no mitering challenge. As explained in [2] and demonstrated in [3], it depends on the geometric details of the circuit, whether the final joint will match as well. Traveling along the circuit, you accumulate a certain amount of *torsion*. The final joint matches properly, if and only if the total amount of torsion is a symmetry of the cross section. Note that the total torsion of a planar polygon is always zero.

A mitering artist has an interest in circuits that can be "mitered all the way round", that is, circuits whose total torsion corresponds to the cross section. There are several techniques available to obtain such circuits:

1. Exploit some freedom in the choice of the vertices to tweak the circuit such that an appropriate total amount of torsion is obtained. For aesthetic reasons, certain relationships between the vertices may need to be kept invariant, such as the knot type. This typically leads to beams with *ad hoc* dimensions (lengths, bevel angles, and torsion angles), which complicates the actual construction of artwork.
2. Restrict the circuit's vertices and segments to a pre-defined 3D-embedded graph with suitable torsion properties. Two special cases are regular polyhedrons and 3D lattices. For example, take a Hamilton path on a cuboctahedron, or a trefoil knot in the cubic lattice.
3. Restrict the torsion changes at each joint. This generalizes the preceding technique, by retaining what is valuable (torsion control) and dropping the (sometimes overly restrictive) pre-defined graph.

   In Figure 2 (right), the miter joints match all the way round, because the beam cross section is a square and the accumulated torsion at each joint changes in steps of 90° (remember that adjacent corner-spanning planes are perpendicular).

The second technique can be compared to walking in a city along the streets. Given the road map, it is easy to make a closed tour, but there is little freedom. The third technique resembles walking in the desert, with a compass and no landmarks. There is more freedom, but then it is a challenge to walk a closed tour.

By the way, [2] also presents *skew miter joints* as a way of making more interesting objects based on lattices, such as the face-centered cubic lattice. However, we will not employ skew miter joints in this article. Here, we pursue the third technique with classic miter joints.

3 Definitions

We will define the circuits of interest using a 3D variant of *turtle graphics* [1]. Turtle graphics were introduced by Seymour Papert as a simple way of producing graphics by a robot turtle, and later by a virtual turtle in the Logo programming language. It is based on self-relative operations, rather than absolute coordinates. At any moment, the turtle has a position and a heading. The turtle walks in the plane and can be instructed to activate (lower) or deactivate (raise) its pen; to move forward or back a given distance; to rotate left or right through a given angle, around its center. This way, many figures can be described through simple programs that control the operation of the turtle.

Our 3D variant of turtle graphics [4] involves a flying turtle whose **state** is defined by its **position** and **attitude** in 3D space. The turtle's attitude is determined by its **heading** (a vector) and its **normal** (a vector perpendicular to the heading, defining the relative up direction). The plane that contains the heading and is perpendicular to the normal is called the turtle's **base plane**.

The turtle starts in the origin heading along the positive x-axis and with the positive z-axis as normal (see Figure 3, left), so that the (x, y)-plane is its base plane. For our purposes, the pen is always active. The flying turtle obeys these commands:

$Move(d)$ moves distance d in the direction of the current heading;

$Turn(\varphi)$ turns clockwise about the current normal by angle φ, changing the heading but not the normal;

$Roll(\psi)$ rolls clockwise about the current heading by angle ψ, changing the normal but not the heading.

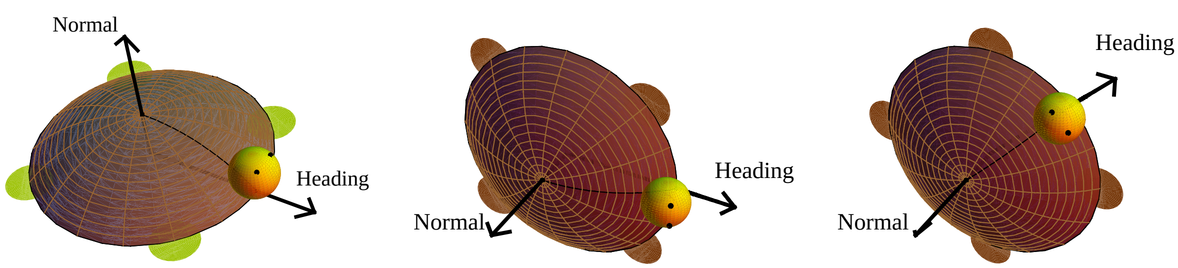


Figure 3: *3D flying turtle in initial state (left); after* $Roll(90^\circ)$ *(middle); after* $Roll(90^\circ)$, $Turn(45^\circ)$ *(right)*

By suitably combining *Turn* and *Roll* commands, the turtle can head in any direction and produce 3D paths.

We define $Segment(d, \psi, \varphi)$ as the sequence of the three commands $Move(d)$, $Roll(\psi)$, $Turn(\varphi)$, where d, ψ, and φ are given parameter values. A sequence of *Segment* commands describes a 3D polygonal path.

The **torsion angle** of an internal segment in a directed polygonal path is defined as the directed dihedral angle between the plane spanned by the segment and its predecessor segment, and the plane spanned by the segment and its successor segment. Note that the torsion angles are the same as the roll angles in paths produced by a sequence of *Segment* commands.

We call a turtle program, typically consisting of *Segment* commands, **closed** when it returns the turtle to its initial position, and we call it **properly closed** when it returns the turtle to its initial *state*. That is, in order to be properly closed, not only should the turtle's final position equal its initial position, but also its final attitude must equal its initial attitude.

A **polygonal path of constant torsion**, or CT path in short, is a path P produced by a sequence of $Segment(d_i, \psi_i, \varphi_i)$ commands that all have the same value for $|\psi_i| = \psi$, with all $d_i > 0$ and $\varphi_i \neq 0$ modulo 180°. Path P is called **regular**, when all $d_i = d$ and $\varphi_i = \varphi$ for $0 < \varphi < 180^\circ$. P is called a polygonal circuit of constant torsion, or briefly a **CT polygon**, when it is a *properly* closed CT path. A *regular CT polygon* is completely determined by the constants d, ψ, and φ, and the sequence of roll signs for ψ. All its segments have the same length, all corner angles are $180^\circ - \varphi$, and all torsion angles are $\pm\psi$.

The regular CT polygon of Figure 1 has $\varphi = 130.06°$, $\psi = 90°$, and ψ-signs ++−−++−−++−−++−−. With $\psi = 90°$, $\varphi = 120°$, and sign sequence +−−++−−++−−++− one obtains a circuit (see Figure 4, left), but the angle at the origin is clearly less than 60° (the final and initial heading are not equal, and the torsions of the two segments at the origin is much less than 90°. The turtle program is closed but not properly closed.

Figure 4: *CT path that is not properly closed (left); regular CT polygon with self-intersection (right)*

Taking $\psi = 90°$, $\varphi = 112.456°$, and the sign sequence +−+−+−+−+− yields a regular CT polygon that exhibits self-intersection (see Figure 4, right). Even if the abstract (infinitely thin) circuit does not self-intersect, this may still happen when thickening the segments (too much).

4 Analysis

In this section, we present some theorems about 3D turtle geometry (we are not aware of a good reference). Two turtle programs are called **congruent** when they trace out congruent paths, and **path-equivalent** when they trace out identical paths. We call them **equivalent**, denoted by $\equiv$, when they are path-equivalent *and* lead to the same *final state* (especially, the same final attitude).

Every turtle program defines a 3D polygonal path. Every 3D polygonal path starting in the origin with its first segment aligned along the positive x-axis can be described by a suitable sequence of *Segment* commands with $d > 0$, $0 < |\varphi| \leq 180°$, and $0 \leq |\psi| \leq 180°$.

We mention a few basic properties of 3D turtle commands. Adjacent *Move* and *Roll* commands commute:

$$Move(d),\ Roll(\psi) \quad \equiv \quad Roll(\psi),\ Move(d)$$

Note that, in general, the other pairs do not commute.

Every program consisting of *Turn* and *Roll* commands only, i.e. without *Move* commands, has an equivalent program of the form $Roll(\psi_1),\ Turn(\varphi),\ Roll(\psi_2)$.

Concerning adjacent *Segment* commands, we have

$$Segment(d_1,\psi_1,\varphi_1),\ Segment(d_2,\psi_2,\varphi_2) \quad \equiv \quad Segment(d_1,\psi_1 \pm 180°,-\varphi_1),\ Segment(d_2,\psi_2 \pm 180°,\varphi_2)$$

This way, all turn angles φ_i in a properly closed turtle program consisting of *Segment* commands can be made positive, by compensating 180° in both adjacent roll angles. A *Segment* program with all turn angles positive is said to be in **normal form**. This is especially interesting for properly closed circuits that have $\psi_i = \pm 90°$, because in that case $\psi_i \pm 180°$ has the same absolute value as ψ_i, and hence the normal form is a CT polygon.

For a properly closed turtle program consisting of *Segment* commands, the total amount of torsion (as needed to determine closure of the mitering [2]) equals the sum of all roll angles (taking signs into account).

Consequently, if the roll angle of each segment is a symmetry of the cross section, then the circuit can be constructed with miter joints that match all the way round. For instance, with $\psi = \pm 90^\circ$ a square beam will always work.

As observed in the preceding section, a closed turtle program (final position = initial position), in general, need not be properly closed (also, final attitude = initial attitude). However, for every closed turtle program there exists a congruent properly closed turtle program consisting of *Segment* commands. Note that the *Roll* and *Turn* at the (end of the) last segment are not relevant for closure. They can always be adjusted to roll the turtle such that the last turn can make the final heading coincide with the initial heading. An extra *Roll* may be needed to align the final normal with the initial normal. This extra *Roll* command can then be merged into the first *Segment* command, thereby effectively rotating the entire path about the first segment, to bring its last segment in the (x, y)-plane.

Note that when a closed constant-torsion program is made properly closed by the preceding transformation, the resulting congruent program possibly no longer is a constant-torsion program! In that case, its path is (rightfully) not a CT polygon (cf. Figure 4, left).

By definition, the distance between adjacent vertices in a regular CT polygon is constant. Furthermore, because all corner angles are equal as well, we also have that distances between vertices that are *two segments apart* are equal. But it is even the case that distances between vertices that are *three segments apart* are equal, because all torsion angles are equal in absolute value.

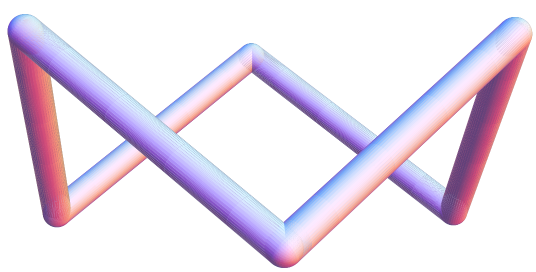
The special case $\psi = 0$ modulo 180° yields planar figures. The regular CT polygons with $\psi = 0$ are the classic regular (planar) polygons. In order to close, φ must divide a multiple of 360°, in which case closure is proper as well. When $\psi = 180^\circ$, constant-torsion paths will zig-zag and never close.

In the remainder, we mostly restrict ourselves to regular CT polygons with $\psi = 90^\circ$.

5 Constructions

We already exhibited some regular CT polygons. Can all regular CT polygons be characterized? What combinations of corner and torsion angles, and sign sequences produce regular CT polygons?

There are some obvious regular sign patterns to try. For instance, a zig-zag obtained by repeating +−. Figure 5 (left) shows +−+−+−, which happens to close properly for $\psi = \varphi = 90^\circ$ (it is a path on a cube). In general, these zig-zags are Hamilton paths on antiprisms (including tetrahedron and octahedron, which have $\psi \neq 90^\circ$). When allowing self-intersection, they traverse the diagonals on the vertical faces of prisms (cf. Figure 4, right). In this family the vertices lie in two planes (layers).



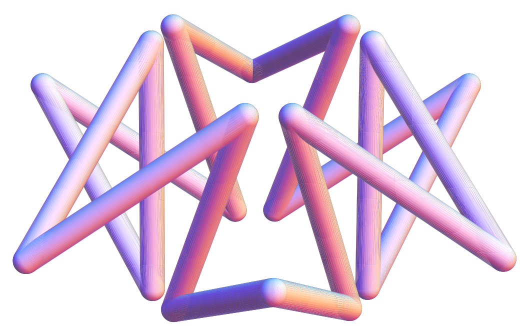


Figure 5: *Regular CT polygons with sign patterns +−+−+− (left) and +++−−−+++−−−+++−−− (right)*

There is also a three-layer family constructed by repeating ++−−, of which Figure 1 shows an example. The vertices now lie in three layers. Although it is possible to give a formula to determine the corresponding φ for given ψ, we have used numerical approximation techniques. Figure 6 shows a plot of the distance between initial and final position of the turtle for the regular CT paths with $\psi = 90°$, signs ++−−++−−++−−++−−, and φ ranging from 0 to 180°. There are three solutions for φ, where the distance is zero, that is, where closure occurs, viz. around 50°, around 130° and at 180° (in the latter case, the path degenerates because all segments align). In fact, it came as a delightful surprise when we first discovered that this path closed properly for non-trivial φ.

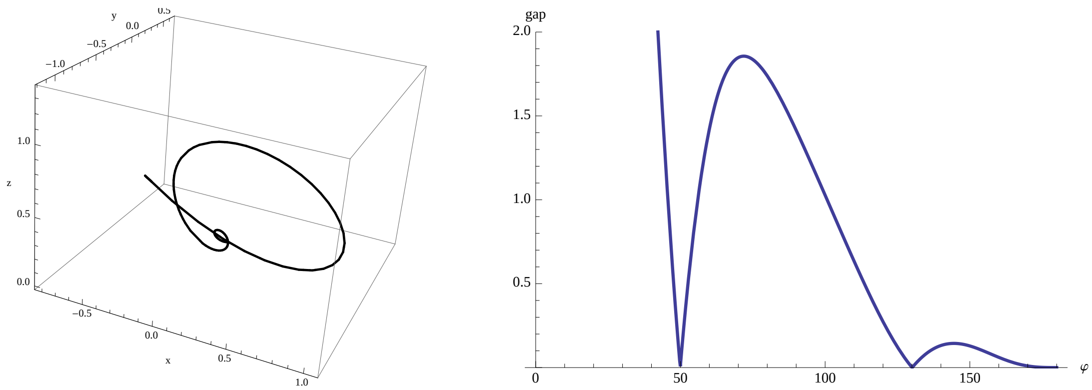


Figure 6: *Final position (left) and gap between initial and final position (right) for torsion signs ++−−++−−++−−++−− as function of corner angle* φ

A four-layer family is obtained by repeating +++−−−; see Figure 5 (right) for an example (it also appears in Figure 8, left). More complicated patterns also exist.

So far we have no clear insight into what patterns work. We only have many examples, found by exhaustive checking of short sign patterns, and interactive exploration of promising longer sign patterns. It is interesting to note that among all the examples, there are none that have a Möbius twist or that are knotted.

6 Artwork

The pieces of artwork shown in this section are all based on regular constant-torsion polygons with torsion angles $\psi = \pm 90°$. The beams have a square cross section, which makes it relatively easy to cut the beams appropriately, and it ensures that all joints are matching classic miter joints. A nice property of these beams is that they are all congruent (identical or mirror images).

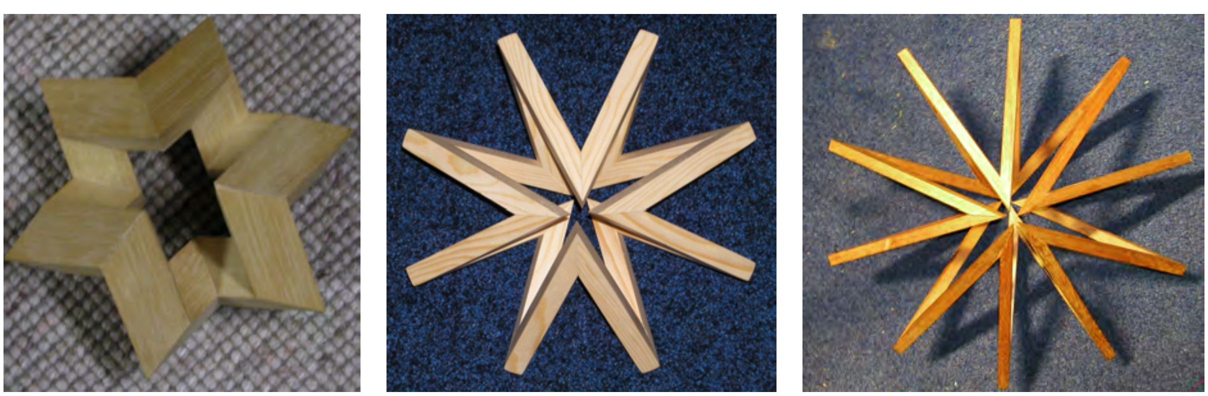


Figure 7: *Artworks based on regular CT polygons with three layers (12, 16, 20 segments, wood)*

Figure 7 presents three instances of the three-layer family described in the preceding section. The one in the middle is also featured in Figures 1 and 2. Their roll sign patterns consist, respectively, of three, four, and five repetitions of the sequence ++−−. The corresponding corner angles $180^\circ - \varphi$ are 90°, 49.94°, and 38.17° respectively. Note that the bevel angles are half the corner angle, and thus quite acute. For the object on the left, the cross section was rotated over 45°, so that the faces of adjacent beams do not lie flush at the joints, in contrast to the other two.

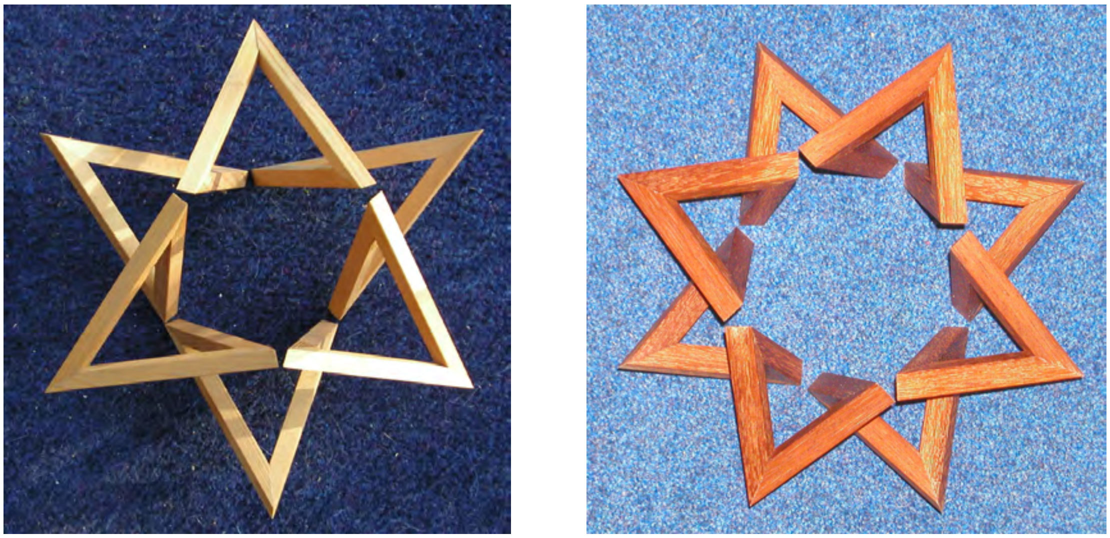


Figure 8: *Artworks based on regular CT polygons with four layers (18, 24 segments, wood)*

The two objects in Figure 8 are instances of the family of four-layer regular CT polygons. Their roll sign patterns consist, respectively, of three and four repetitions of the sequence +++−−−. The corresponding corner angles $180^\circ - \varphi$ are 40.99° and 52.85° respectively.

The object in Figure 9 is based on three repetitions of ++−++−−+−− with corner angle 48.62°. This sign pattern also closes properly with $\varphi = 90^\circ$, in which case it stays in the cubic lattice.

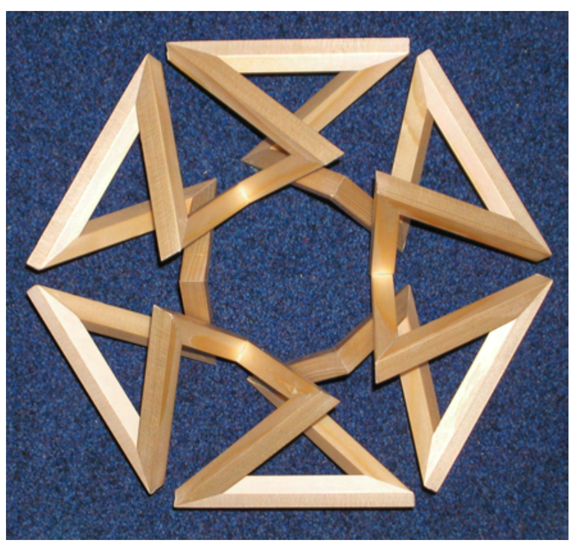


Figure 9: *Artwork based on regular CT polygon with signs* $(++-++--+--)^3$ *(30 segments, wood)*

The object in Figure 10 uses four repetitions of +++−−++−−− and a corner angle of 44.43°. It is less symmetric than the others, since it lacks an up-down symmetry, as the pictures corroborate.

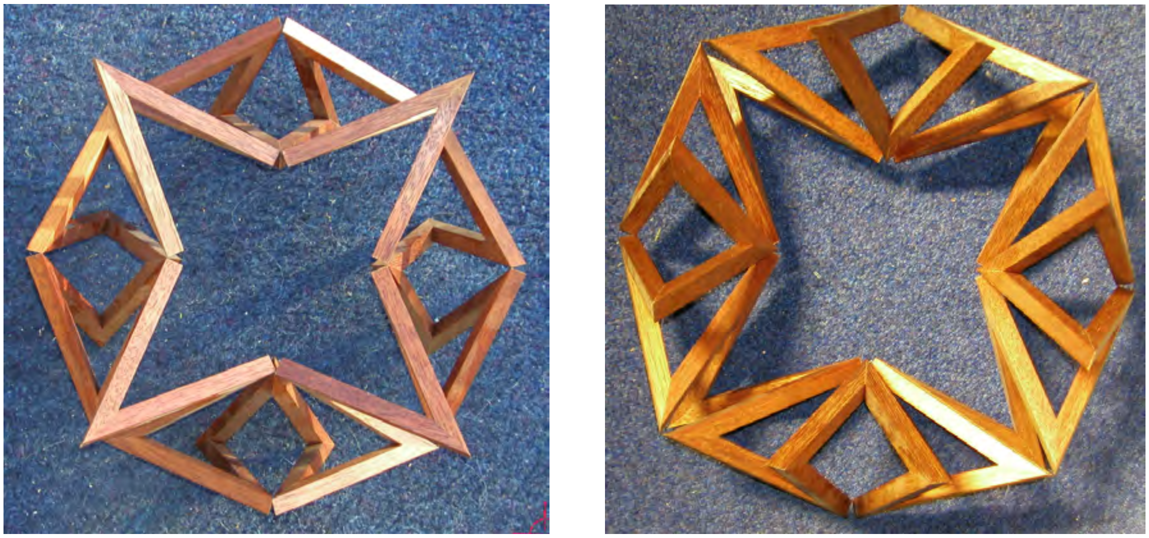


Figure 10: *Artwork based on regular CT polygon with signs* $(\texttt{+++--++---})^4$ *(40 segments, wood)*

7 Conclusion

We have defined and explored regular constant-torsion polygons, as a fruitful inspiration for artwork. The advantage of using constant-torsion circuits is that it gives the mitering artist control over the total amount of torsion, and hence over making all miter joints match properly. In the case of constant-torsion polygons, whose torsion is $\pm 90^\circ$ per segment, it is convenient to construct such artwork with beams having a square cross section: there are only two mirror-image pieces. Using regular CT polygons does raise the problem of finding appropriate angles and torsion signs to ensure that the polygonal chain properly closes onto itself.

We have exhibited several infinite families of regular CT polygons, but the characterization of all regular CT polygons is still an open problem. In our searches, no knots were encountered, nor regular CT polygons that have a Möbius twist. The use of skew miter joints and (skew) fold joints (cf. [2]) in constructing regular CT polygons also offers interesting opportunities for future research.

Acknowledgments All artwork and pictures were made by the second author. The illustrations were made with Mathematica and OmniGraffle. We would like to acknowledge the helpful feedback from the reviewers.

References

[1] Seymour Papert, *Turtle Graphics*, `http://en.wikipedia.org/wiki/Turtle_graphics` (accessed 13 January 2009)

[2] Tom Verhoeff, Koos Verhoeff. "The Mathematics of Mitering and Its Artful Application", *Bridges Leeuwarden: Mathematical Connections in Art, Music, and Science, Proceedings of the Eleventh Annual Bridges Conference, in The Netherlands*, pp. 225–234, July 2008.

[3] Tom Verhoeff. "Mitering a Closed 3D Path". From *The Wolfram Demonstrations Project*, `http://demonstrations.wolfram.com/MiteringAClosed3DPath/` (accessed 13 January 2009).

[4] Tom Verhoeff. "3D Flying Pipe-laying Turtle". From *The Wolfram Demonstrations Project*, `http://demonstrations.wolfram.com/3DFlyingPipeLayingTurtle/` (accessed 30 January 2009).