

Dodecahedral Pentafoil Tangle

Brandon Wang

Yasuhide Okamoto, Jonathan Ko



Figure 1: The reference tangle¹

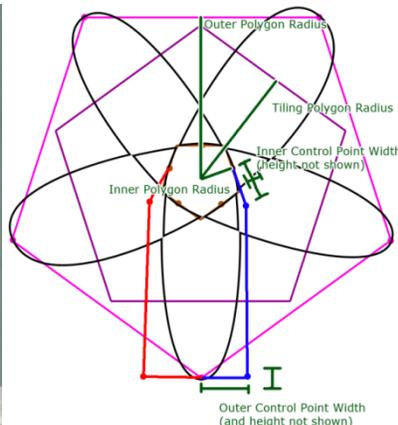


Figure 2: Representation of a knot²

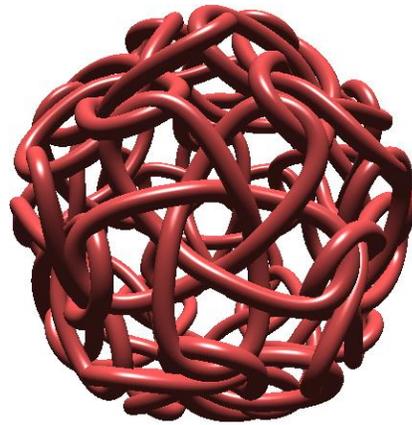


Figure 3: Our generated tangle²

The “Dodecahedral Pentafoil Tangle” is a spherical object of interlocking knots, seen in Figure 1. To create a procedural representation of the tangle, we first take note of the reference object’s symmetry and underlying shape. Each knot is a pentafoil knot, consisting of five protrusions. Knots are placed on faces of a regular dodecahedron, aligning the protrusions of the knot with the sides of the dodecahedron. The protrusions lock with each other by extending past the side of its host pentagon.

Once we know the repeating and underlying structure of the tangle, we can create a single pentafoil and repeatedly tile it over a dodecahedron. The instanced pentafoil can be modified, and the resulting tangle will be uniformly changed on each face.

To create the pentafoil knot, we realize that it is a regular shape, with 5-fold symmetry. Because of its curved nature, we choose a geometrically continuous b-spline to represent the shape. Aligning the knot to the regular polygon, we can start to define a set of control points to construct the b-spline. An inner pentagon is also used to define the inner control points, as the knot contains a gap in the center. From two reference pentagons, we can create control points from uniform distances corresponding to each polygon. This allows us to create the b-spline of a single protrusion, and have the b-spline rotated five times to create the pentafoil knot, as seen in Figure 2. Because we rotate a single b-spline, we can create an infinite number of protrusions in our knot, but we cannot create a Platonic solid with more than five sides per face to hold our knot.

A modified version of the spring 2011 CS184 AS8 framework allows us to procedurally generate a tangle. Figure 3 was generated using our method of b-spline interaction. We define three radii for controlling the three reference pentagons’ shape, and two sets of two-dimensional vectors to represent the distance from a pentagon vertex to a b-spline control point, in a tangential plane with the vertex. These variables are user controlled and can be manipulated at will to interactively and uniformly change the tangle’s shape. Further examples can be seen online².

¹ http://www.cs.berkeley.edu/~sequin/SCULPTS/CHS_bronzes/PentafoilTangle/

² <http://www.brandonwang.net/class/cs285/as1/>