

How to print a Hypercube

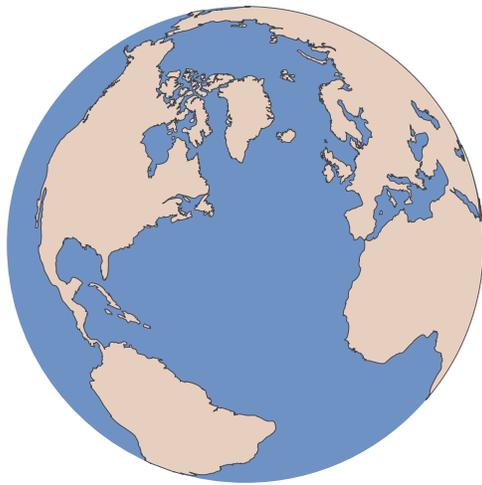
Henry Segerman

One of the things that mathematics is about, perhaps *the* thing that mathematics is about, is trying to make things easier to understand. John von Neumann once said “In mathematics you don’t understand things. You just get used to them.” This is partially true, but we can get a long way towards understanding something new by converting it into something we already know about, and a lot of mathematics consists of *tricks* that make one thing look like another. Maybe the two things aren’t exactly the same, but if we can start from what we already know and work out the differences between the old and the new, we can understand the new.

Another really good way to understand something is to draw a picture of it, to visualise it in some way. Often when we visualise something, we can’t get a perfectly accurate picture, and again we have to use a trick. Perhaps we draw a picture that highlights one feature of an object while ignoring or distorting others. One example of this kind of distortion comes from the problem of drawing a map of the world. The Earth is (roughly) spherical, and perhaps the best way to understand how the continents are arranged on the Earth is to look at a globe. A globe is a very accurate representation of the Earth, the only real difference being a few magnitudes of scale. However, we often want to draw a map in 2-dimensions, and now we run into trouble. The 2-dimensional plane is flat, while the surface of the Earth has curvature, and there’s no way to flatten a sphere onto a plane without introducing some kind of distortion. The most common of map of the Earth uses the Mercator projection. We can see the distortion in the Mercator projection in that features near the North and South poles appear too big on the map in comparison to features near the equator.

David Swart and Bruce Torrence have previously written about mappings from the sphere to the plane in *Math Horizons*, in their article *The Viewable Sphere*. They use various projections from the sphere to the plane to make all-around panoramas; pictures on a 2-dimensional plane of the view in every direction. In this article, we want to use similar ideas to try to visualise objects in 4-dimensional space, for example the 4-dimensional hypercube, or tesseract.

But let’s start with something easier first, and use another one of those understanding tricks: 4-dimensional things are hard to think about, so let’s first look at everything one dimension down. So suppose, as in Edwin Abbott’s classic *Flatland*, a 2-dimensional flatlander living in the plane wants to understand the 3-dimensional cube. We have to use some trick to squish the cube down onto the plane, so that our friend the flatlander can see it with her 2-dimensional senses. How can we do this? Well, any photograph or picture of a cube is a squished cube in the plane. What does a camera do to the geometry of an object? Roughly speaking, a camera puts a plane of film in front of your eye, and marks points on the film where lines between your eye and the object intersect the film. This isn’t quite true because there are lenses involved, your eye is not a single point in space, and other complications. (Although in much of 3-dimensional computer graphics this is essentially what happens in



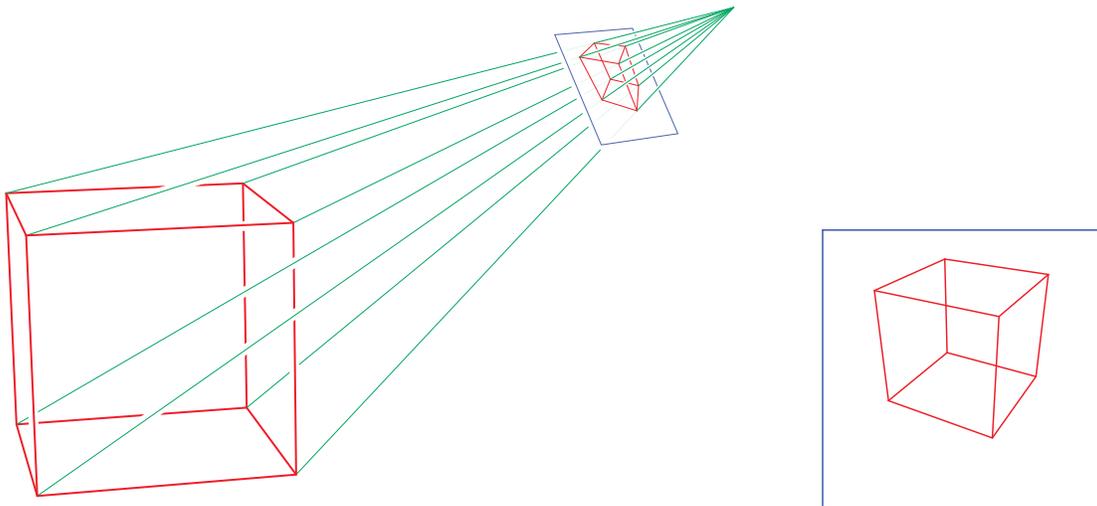
(a) The Earth viewed as a sphere.



(b) The Mercator projection of the Earth

Figure 1

calculating the picture on the screen.) Notice that if we translate the film plane away from the eye but don't rotate it, then the image on the film doesn't change other than scaling up. We could even put the plane behind the object we are looking at, as long as we think about the mapping in terms of the intersection of lines with a plane, rather than rays of light going from the object to our eye!



(a) How to take a photograph of a cube.

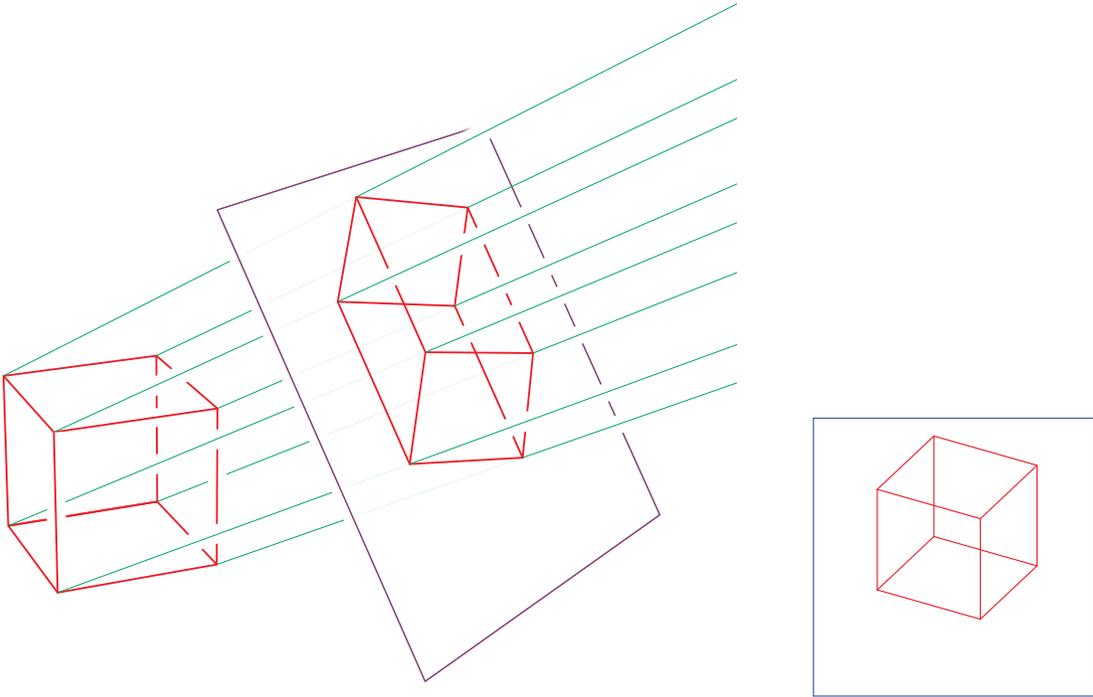
(b) The resulting image.

Figure 2

Although the resulting *perspective projection* of a cube looks very natural to us 3-dimensional viewers, it would be quite strange to a 2-dimensional flatlander. We tell our friend the flat-

lander that all of the angles at the corners of the cube are the same, but they would say that our 2-dimensional picture has all different angles (and it does!). We tell them that the edges of the cube never cross through each other, but they can see two places where they do in our picture. Well, these are some of the flaws we expect in taking a perspective projection from a 3-dimensional object to the 2-dimensional plane.

Another thing we could do is move our eye far away from the cube, so the lines from the cube to our eye become closer and closer to parallel. In the limit, when our eye is infinitely far away from the cube, the lines of sight are parallel and we get an *orthogonal projection*. This has some advantages: lines that are parallel on the cube in 3-dimensions are still parallel on the 2-dimensional film, and lines that are parallel are all the same length as each other, just as they are on the cube. We still get intersections between the edges, and we still get some strange angles.



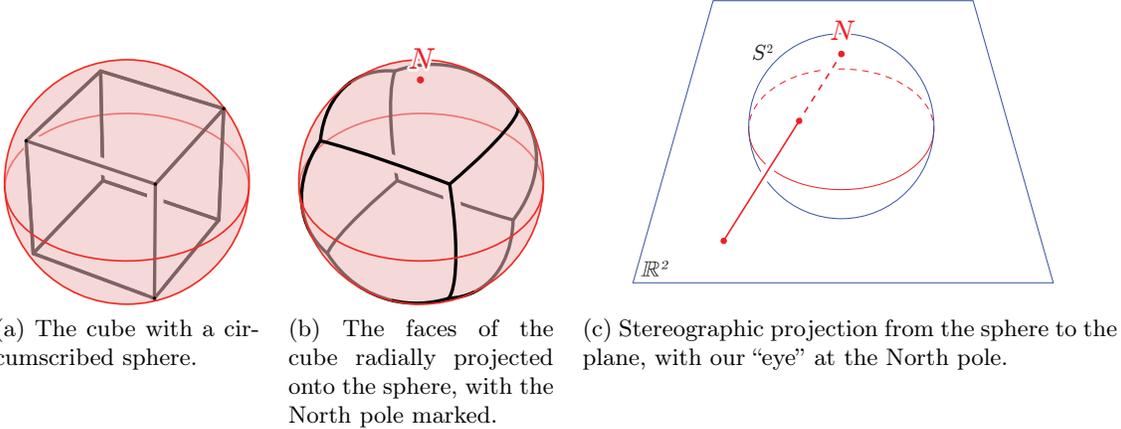
(a) How to take an “orthogonal photograph” of a cube.

(b) The resulting image.

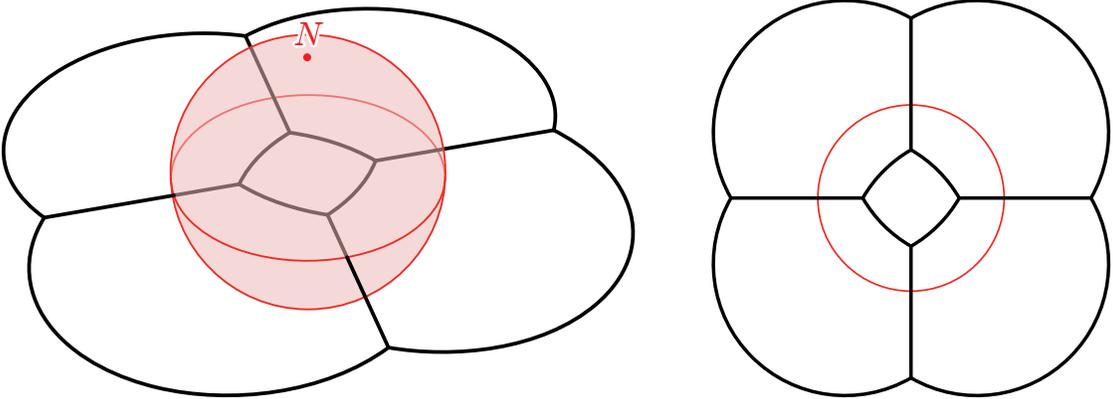
Figure 3

Let’s try something else. Rather than moving our eye away from the cube, let’s move it towards the cube. If we peer in at it from one of its square faces, we should get a picture that doesn’t have intersections between the edges. In order to do something about the angles, we will use another trick that looks a little odd at first. Put a sphere around the cube, so that the corners of the cube all lie on the sphere. Then, “blow air into the cube”, inflating it like a beach ball so that its edges and faces now lie in the sphere. More precisely, project the cube radially outwards onto the sphere, mapping a point of a face or edge to the intersection between the sphere and a ray based at the center of the sphere. This is like the perspective projection from before, with our eye at the center of the sphere, but with the film being the sphere, rather than a plane. Now that we have our “beach ball cube”, take a perspective projection, but with

our eye so close to the cube that it is *on the sphere itself*. This kind of perspective projection, thought of as a map from the sphere to the plane, is called *stereographic projection*.



(a) The cube with a circumscribed sphere. (b) The faces of the cube radially projected onto the sphere, with the North pole marked. (c) Stereographic projection from the sphere to the plane, with our “eye” at the North pole.



(d) Stereographic projection of our “beach ball cube” pattern to the plane. (e) The resulting image, with the equator of the sphere marked.

Figure 4: A cube, radially projected to the circumscribing sphere, then stereographically projected to the plane. Since we can move the film plane towards or away from our eye without changing the image other than by scaling, we have chosen to place the film plane so it passes through the center of the sphere.

What was the point of all of that? First, we have removed the intersections between edges. In fact, stereographic projection is a *bijective* map from the sphere (minus the North pole) to the plane. This means that any pattern on the sphere is transferred to the plane with no possibility of any parts of the pattern crashing into any others. In addition, notice that all of the angles between edges are the same! They are all at 120 degrees to each other. In fact, stereographic projection is a *conformal* map, which means that it preserves angles. So the 120 degree angles we see in the image are the same as the 120 degree angles we see on the “beach ball cube”. Our flatlander friend might complain that these ‘lines’ aren’t straight any more, and that the angles between faces were supposed to be 90 degrees on a cube, but, well, you can’t have everything.

We can also see the conformality of stereographic projection when we apply it to the Earth. The shapes of all of the coastlines are locally accurate, with no shearing or bending, although the larger scale distortion of scale becomes very extreme as we look further North on the globe!

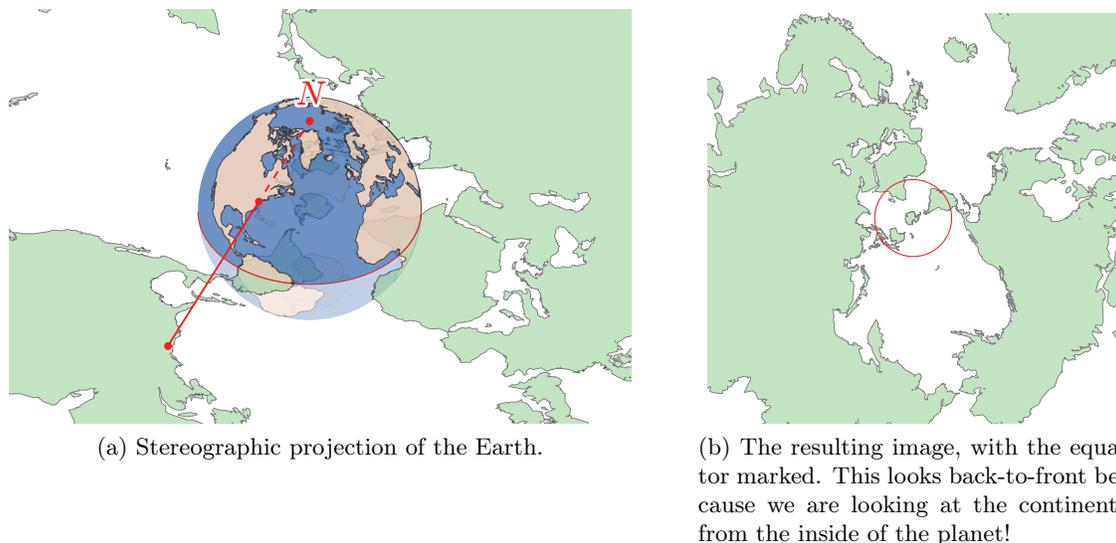


Figure 5

Ok, so now we want to visualise the hypercube. Just as you can build a square by starting with two line segments, moving them apart and connecting the ends together, and you can make a cube by starting with two squares, moving them apart and connecting the corners together, you can make (the edges of) a hypercube by starting with two cubes, moving them apart and connecting the corners together. The problem is that we want all of the edges meeting at a corner to be at right angles to each other, and there aren't enough dimensions in our 3-dimensional world to do this for the hypercube. (In the figure, there aren't even enough dimensions to do the cube, since the page is 2-dimensional!) However, this isn't a problem if we do our construction in 4-dimensional space.

The hypercube has 16 vertices. If we put them at $(\pm 1/2, \pm 1/2, \pm 1/2, \pm 1/2)$, then they all sit on the unit hypersphere in 4-dimensional space, since $1/2^2 + 1/2^2 + 1/2^2 + 1/2^2 = 1$. So we can do exactly what we did with the cube: Radially project the boundary of the hypercube onto the hypersphere, then stereographically project from the hypersphere to 3-dimensional space. The result is shown in Figure 7a. The 3-dimensional version of stereographic projection is also a bijection and is also conformal. Having the edges miss each other isn't so hard for a 3-dimensional picture of a 4-dimensional object; an orthogonal or perspective projection of the hypercube could have this property. However, we can also draw in the 2-dimensional faces (shown as translucent films), and then we would have edges crashing through faces unless we use stereographic projection.

The edges are thickened up in this picture. Although it looks as if the outer edges are thicker than the inner edges, this is just a trick of perspective: the edges are all the same thickness on the unit hypersphere in 4-dimensional space, and the outer edges look bigger only because they are closer to the "eye" of the projection!

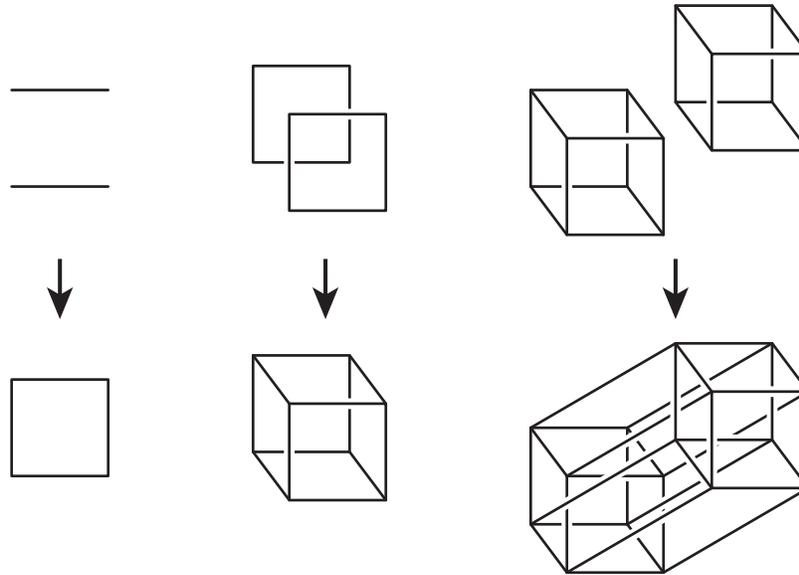
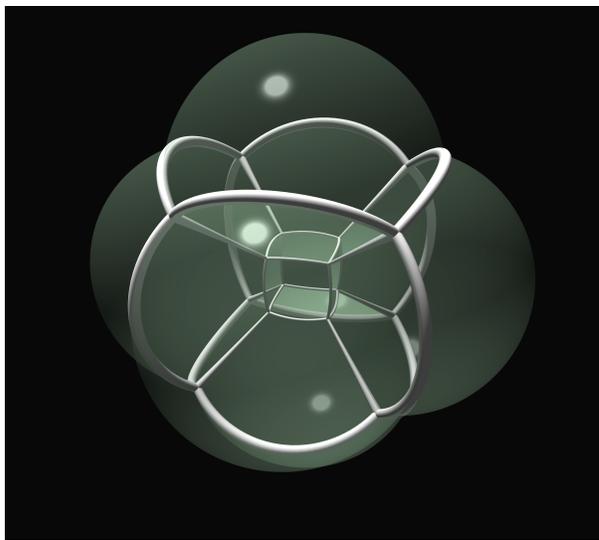


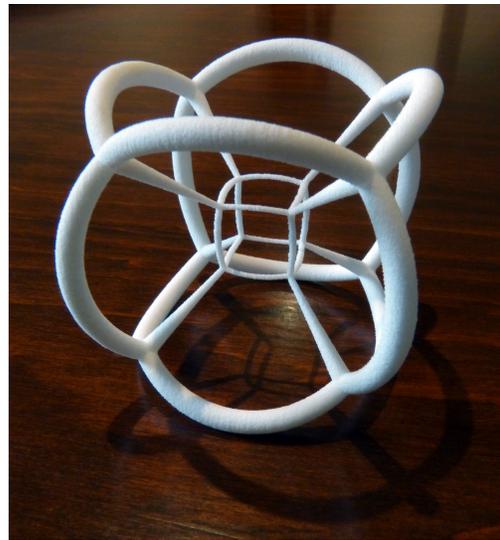
Figure 6: We can build an n -dimensional cube by taking two copies of the $(n - 1)$ -dimensional cube and connecting corresponding corners together.

There’s one more thing we want to do, which is to actually *print* a “picture” of the hypercube. By which we mean 3D print a model, because if we want the best possible visualisation of the hypercube, why lose two dimensions when we could lose only one? Sadly, the photograph of the 3D printed model shown in Figure 7b does lose two dimensions after all. For now at least, *Math Horizons* only uses 2D printing technology.

The cube is just one of the regular 3-dimensional polyhedra, the others being the tetrahedron, the octahedron, the dodecahedron and the icosahedron. Similarly, there are other regular 4-dimensional polytopes than the hypercube. The hypercube is also called the “8-cell”, because it has 8 cubical cells (the square has 4 sides, the cube has 6 faces, the hypercube has 8 cells). One of our favourite regular 4-dimensional polytopes is called the 120-cell, which has 120 dodecahedral cells. We can print a picture of the 120-cell in exactly the same way as we did for the hypercube. Figure 8 actually only shows half of the 120-cell, cut along the “equatorial” sphere half way between the North pole and the South pole of the unit hypersphere. We can get the other half by “reflecting” across the equatorial sphere, which looks like Figure 9. However, since the other half gets very close to the “eye” at the North pole, the objects near the North pole get very large (and so very expensive to print!), just as in Figure 5b.

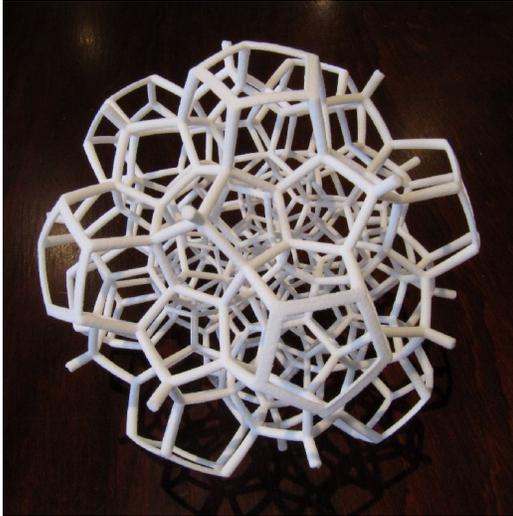


(a) Radial then stereographic projection of the hypercube.

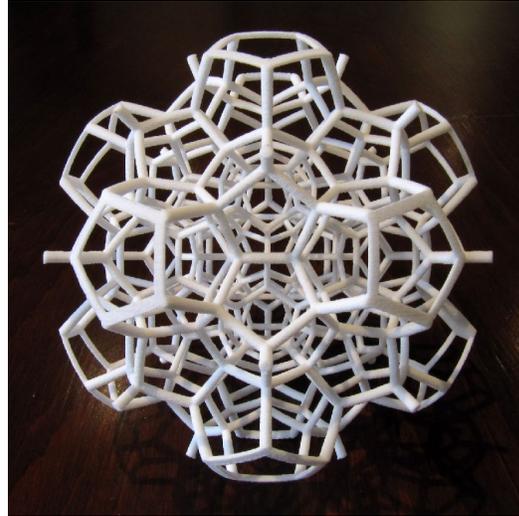


(b) A 3D printed model of the hypercube.

Figure 7



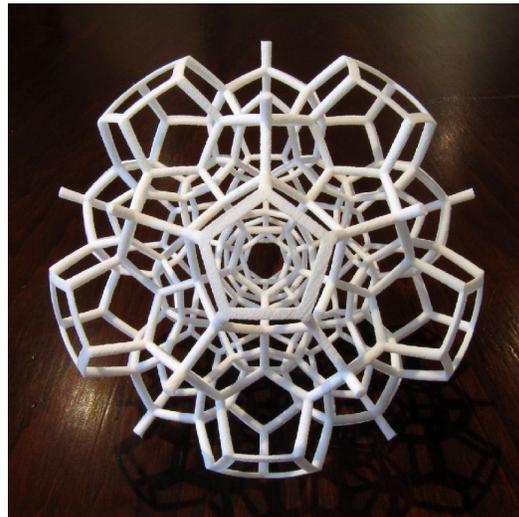
(a) A generic viewpoint.



(b) A 2-fold symmetry axis.



(c) A 3-fold symmetry axis.



(d) A 5-fold symmetry axis.

Figure 8: Half of a 120-Cell. The 3D prints in this article were designed with Saul Schleimer and printed using Shapeways.com.

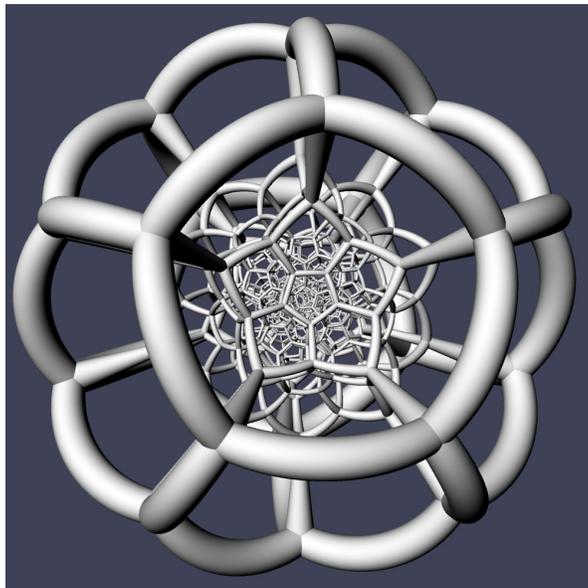


Figure 9: All of a 120-cell.