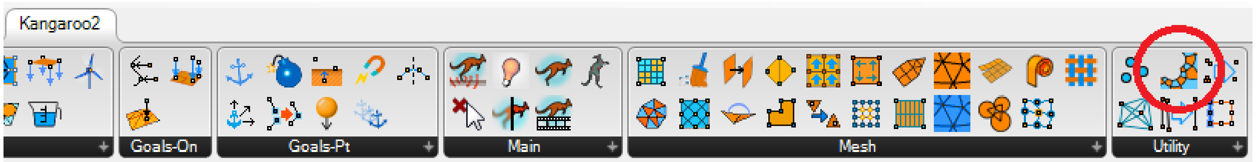
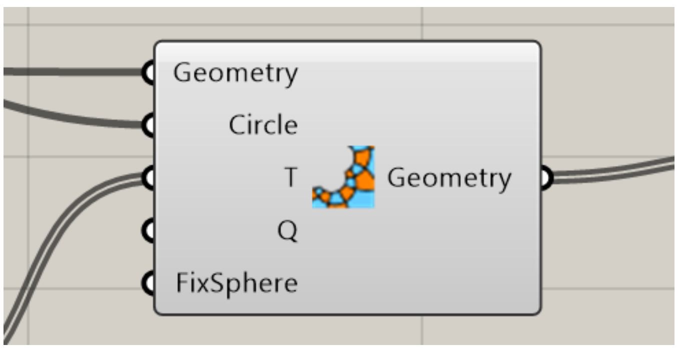
Möbius Transformations

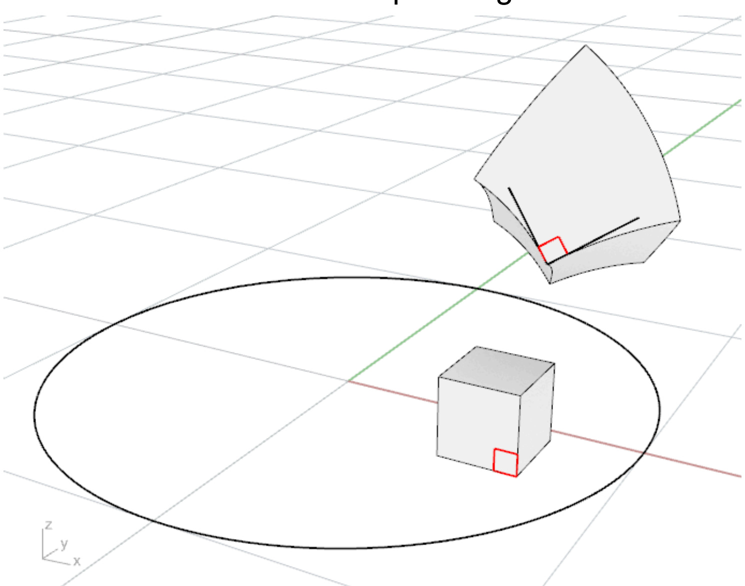
The Möbius Transformation component can be found under the Utility tab of Kangaroo2 (in versions 2.5 and up)



This deforms the geometry input to Geometry(G) using a particular mathematical transformation named after August Ferdinand Möbius.
It works on breps, trimmed and untrimmed surfaces, meshes, curves, polylines, lines, points, circles, arcs and planes.



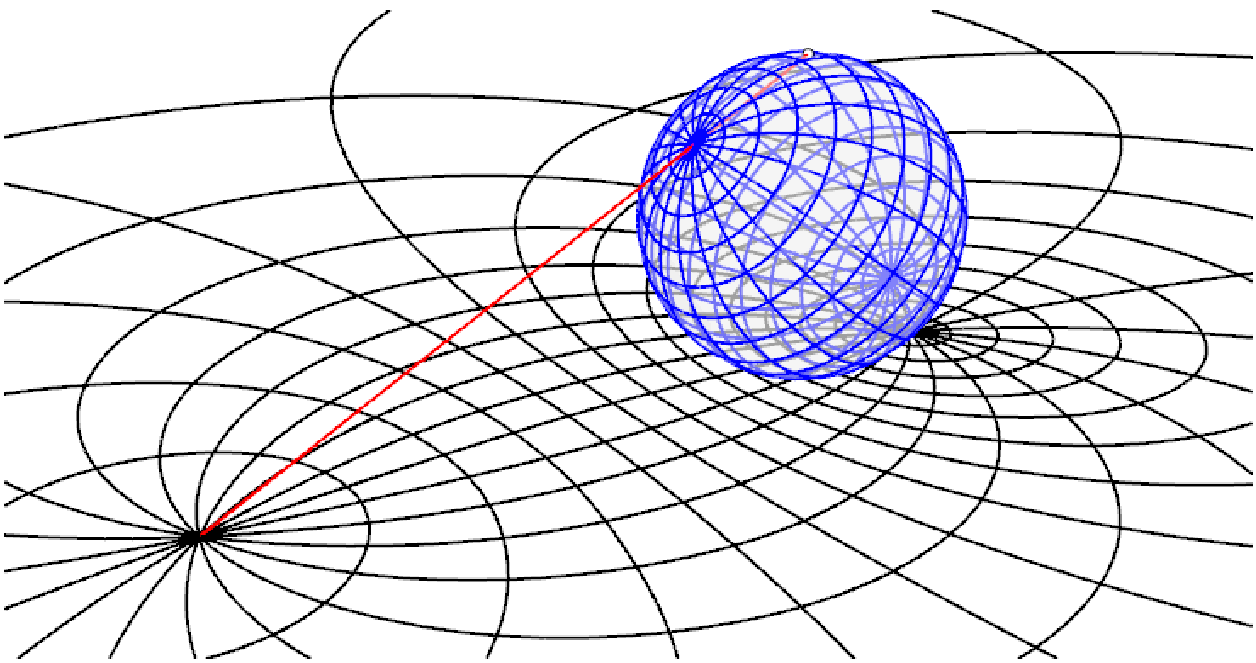
Except for the simple rigid transformations (translation or rotation) and scaling, all other transformations (such as twist, bend or taper) change the angles of the geometry, but Möbius transformations keep all angles the same.



Möbius transformations are also unique among the non trivial transformations in that they preserve circles, arcs and spheres (the Möbius transformation of a circle is another circle etc), if we consider lines as arcs of infinite radius, and planes as spheres of infinite radius.

There are several other useful and well known transformations which can be seen as particular cases of Möbius transformations:

- Inversion of 3d geometry in a sphere
- Inversion of planar geometry in a circle
- Stereographic projection from a sphere to the plane
- Inverse stereographic projection from the plane to the sphere



The Möbius transformation component can also be seen as applying 4 dimensional rotations.
What we normally just call a sphere is sometimes referred to by mathematicians as the *2-sphere*. Even though it 'lives' (or is *embedded*) in familiar 3 dimensional space, it is itself a 2 dimensional manifold.
Similarly, we can consider a *3-sphere* as the set of 4-dimensional points at equal distance from some centre.
(Note that here we are talking about a 4th *spatial* dimension, not taking time as the 4th dimension.)

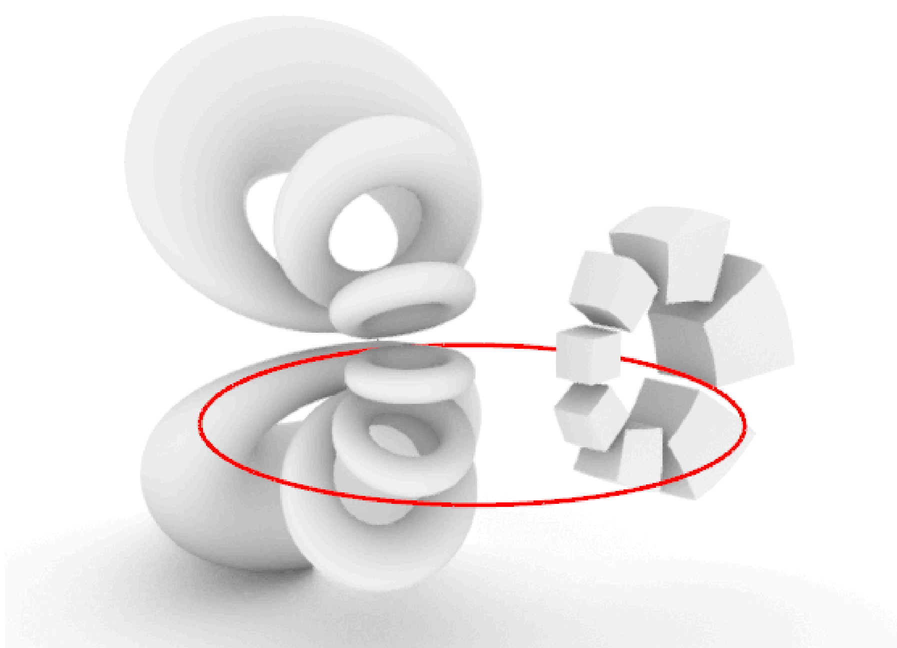
We cannot display 4 dimensional objects directly, but we can use projection to transform them into 3d. In particular here we use stereographic projection because it preserves angles. Normally stereographic projection refers to the mapping from the 2-sphere to the plane, but there is also a higher dimensional version of this, which maps the 3-sphere to *flat* 3-space (ie the familiar Euclidean 3 dimensional space we normally deal with in Rhino). There is also an equivalent inverse mapping to go the other way.

The way the Möbius transformation component works is to:

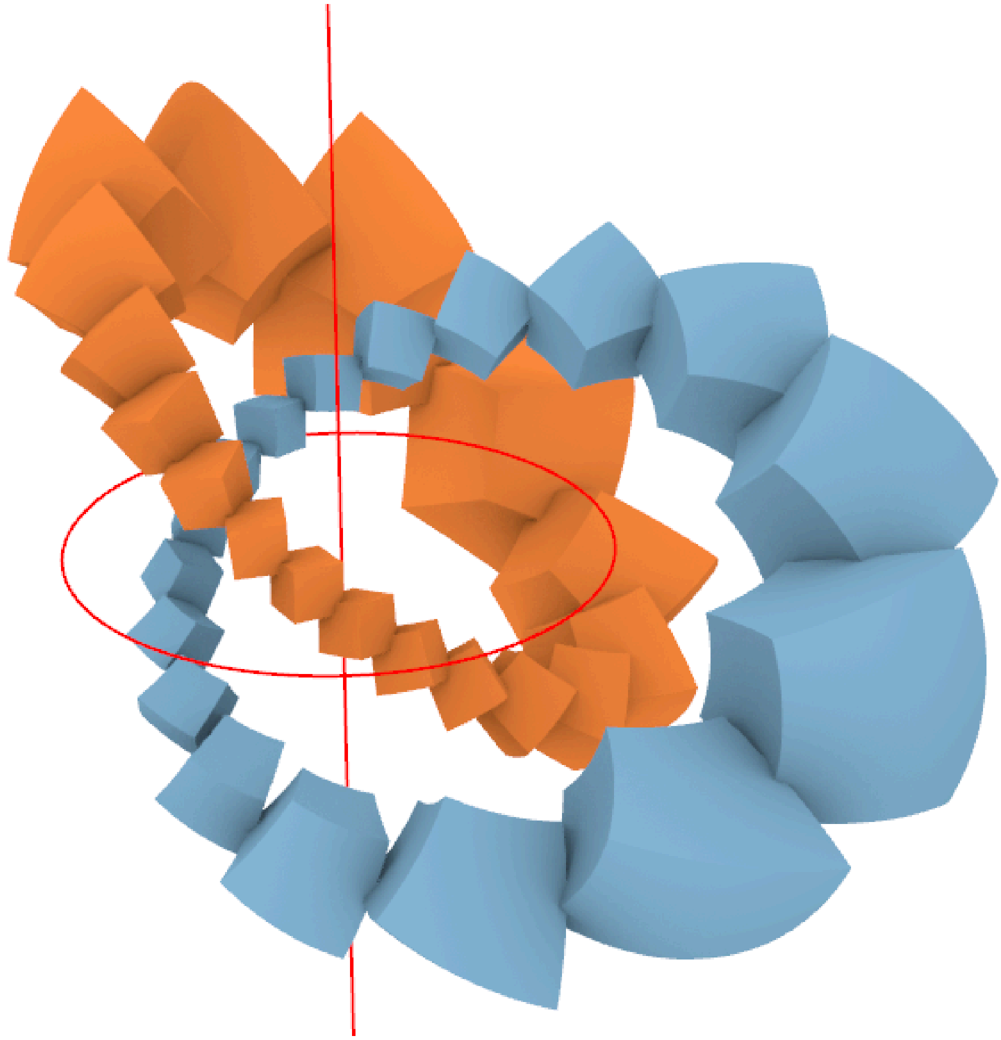
- Take some geometry in flat 3-space
- Inverse stereographically project this geometry to a 3-sphere
- Rotate that 3-sphere in 4-space
- Stereographically project back from the 3-sphere to flat 3-space
- Output the transformed geometry

Just as a rotation of the 2-sphere always leaves 2 points unmoved, rotation of the 3-sphere leaves 2 circles in place. When projected into flat 3-space, one of these circles becomes a straight line. The other fixed circle is the circle you set as the C input for the component. (note that while points on the fixed circles stay on those circles, they can move along them)

The geometric effect in 3d of the process described above can be seen as rolling the space *through* and around the fixed circle, a bit like the motion of a smoke ring.



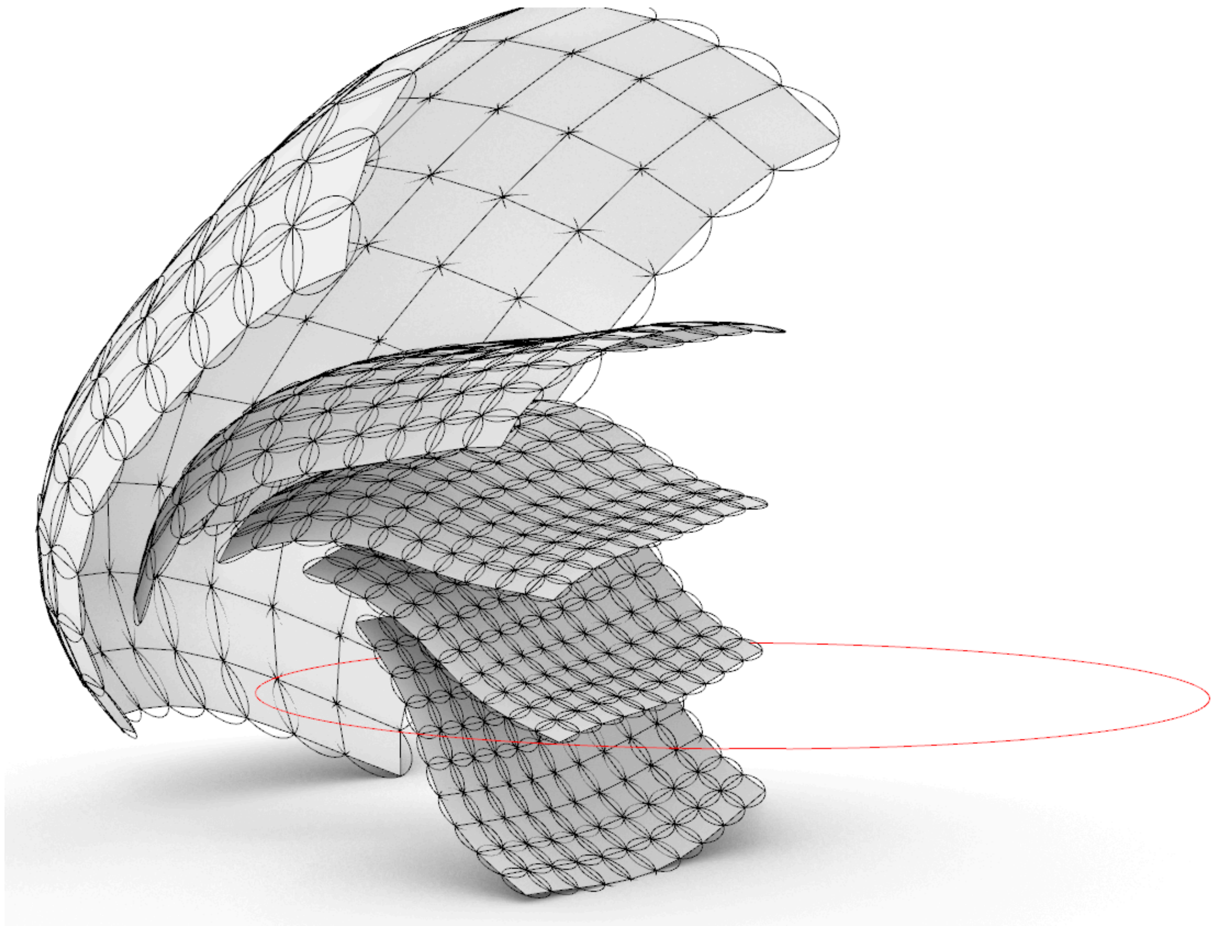
In 4d we have an extra parameter for rotation, and in the component this is controlled with the Q input. This has the effect of simultaneously rotating about the axis line of the circle. Its default setting is 0, but if set to 1, the result is that for each rotation 'through' the circle, we also make one rotation around the axis line.



The T parameter controls the angle of the 4d rotation.
Rotation by 2π brings everything back to its original position.
Rotation by π is equivalent to *inverting in a sphere* with the same radius and centre as the input circle.
Rotating by π/2 is a stereographic projection from the 2-sphere to the plane.
Rotating by -π/2 is an inverse stereographic projection from the plane to the 2-sphere.

Note that with stereographic projection, as we get closer to the top of the sphere, the projected point on the plane gets further away. We say the point at the exact top gets projected to the *point at infinity*. Since we cannot represent this point in Rhino, it gets mapped to the origin.

Because circles are preserved by Möbius transformations, they can be particularly useful when working with other geometry which is based on circles. For instance:
-Circle packings map to circle packings
-Circular meshes (where the vertices of each face lie on a common circle) map to other circular meshes. Since lying on a common circle also implies a common plane, these meshes also have planar faces.

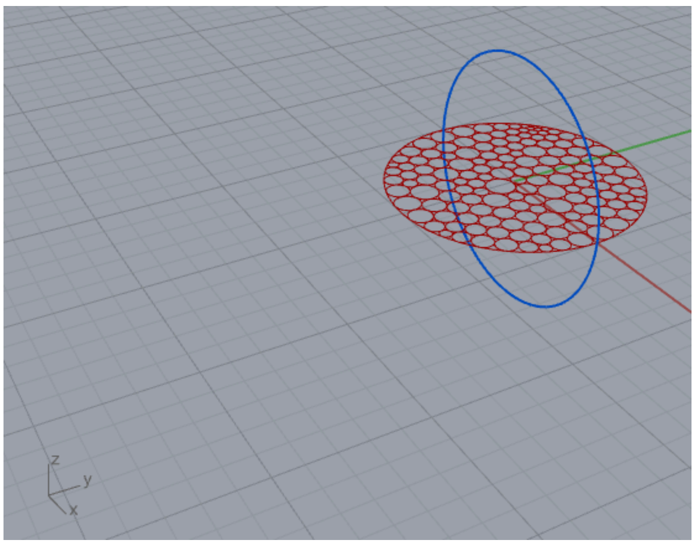


We can also consider the curve swept out by a point as we vary the T input (by inputting a series of values to T). When Q is set to 1, each point traces out a circle this way, and any 2 circles are linked. This is the projection of the *Hopf fibration* (https://vimeo.com/3945328).
We can also input fractional values to Q to generate *torus knots*.

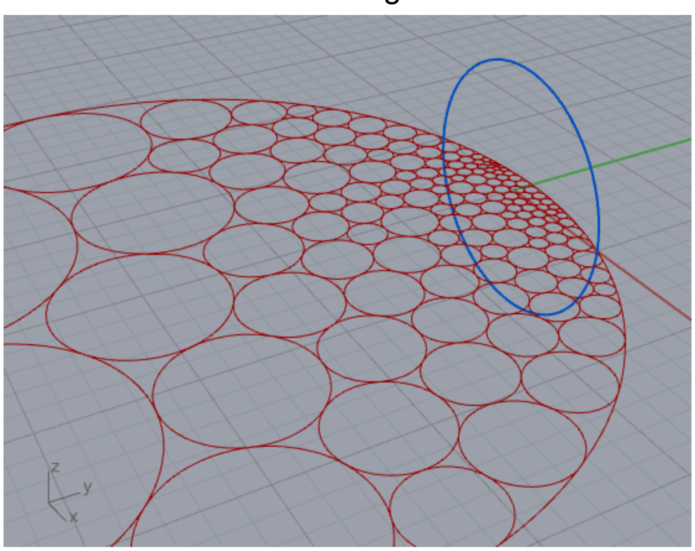
Similarly we can sweep curves through varying values of T to generate surfaces. In this way we can generate surfaces such as the Möbius band with circular boundary (https://vimeo.com/2037835), and the Lawson Klein bottle(https://vimeo.com/2495945), both of which are swept out by circles.

The F(FixSphere) input option, if set to True, moves and scales the geometry so that objects inside the sphere with the same centre and radius as the circle C remain inside that sphere
(as long as the T parameter is between -π/2 and π/2, otherwise the sphere is inverted, and this option keeps everything that was inside the sphere on the outside).

Starting circle packing (generated with the TangentIncircles Kangaroo goal):



Möbius transformation using blue circle as C and with F=False



Möbius transformation with F=True (note how the packing is ‘bunched up’ on one side, but the boundary circle is still the same as the original)