Stereographic Projection

This document describes how to create a sphere model with cutouts placed in such a way that when a point light is placed at the top of the sphere it will cast a shadow that is a regular grid. The resulting work of art is a physical manifestation of a projection of a sphere onto a plane.

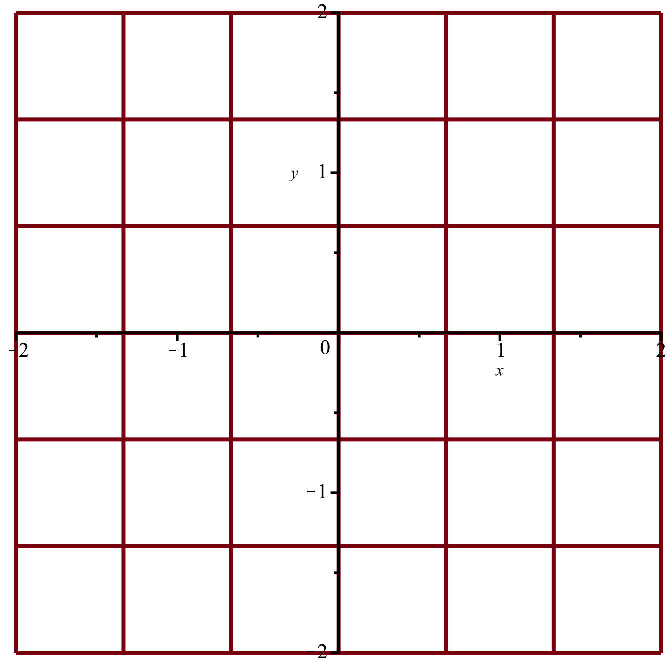
The model in the picture is 6.5" in diameter, and is made out of cedar, with a walnut stand. It was made in two pieces; essentially two bowls, turned on a lathe to hollow and shape it. The hemispheres are then glued into a sphere and then cut-out.

Maple is used to calculate the projection layout, calculating the lengths, widths and intersection points of the arcs on the sphere.

Grid

The goal is to take this grid and project it onto a sphere.

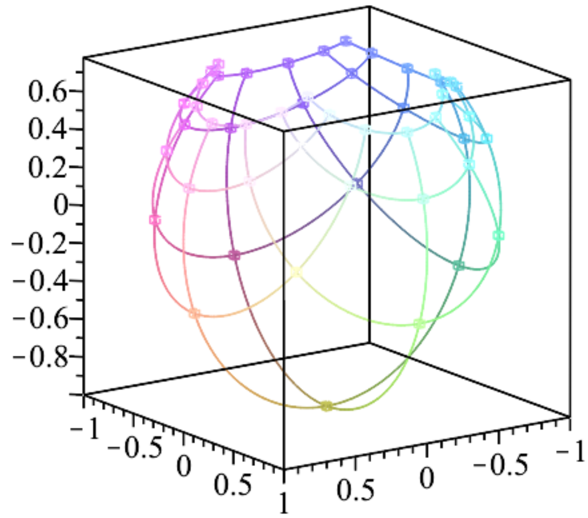
```
> plots:-display([
      (for i from -2 to 2 by 2/3 do
           plots:-implicitplot( x=i, x=-2..2,y=-2..2 ),
           plots:-implicitplot( y=i, x=-2..2,y=-2..2 );
      end do)
  ]);
```




```
> proj := (x,y)->[ 2*x/(1+x^2+y^2), 2*y/(1+x^2+y^2), (-1+x^2+y^2)/
  (1+x^2+y^2)];
```

$$proj := (x, y) \mapsto \left[\frac{2 \cdot x}{1 + x^2 + y^2}, \frac{2 \cdot y}{1 + x^2 + y^2}, \frac{-1 + x^2 + y^2}{1 + x^2 + y^2} \right] \tag{1.1}$$

```
> plots:-display([
      (for i from -2 to 2 by 2/3 do
           plots:-spacecurve( proj(x,i), x=-2..2 ),
           plots:-spacecurve( proj(i,y), y=-2..2 );
      end do),
      plots:-pointplot3d([
        for i from -2 to 2 by 2/3 do
           (for j from -2 to 2 by 2/3 do
               proj(i,j);
           end do);
        end do], symbol=box)
  ]);
```



Physical Grid

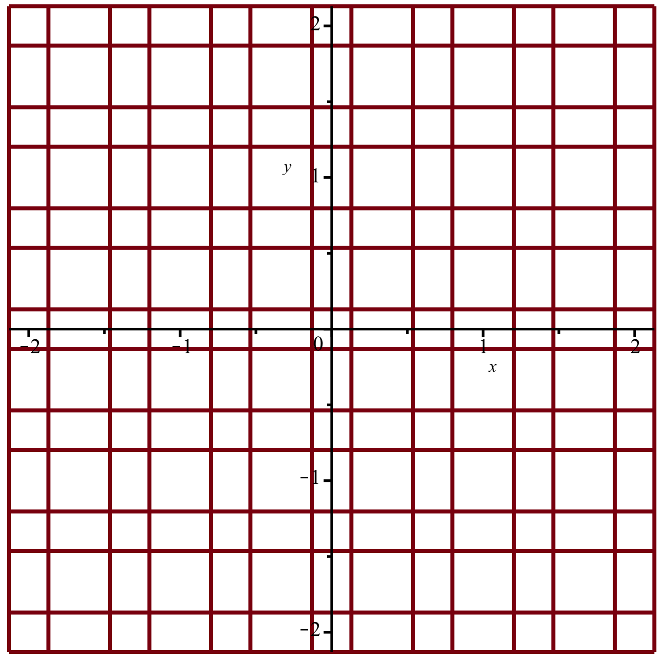
The sketch of the globe lacks thickness. Let's redefine the grid to have some width:

Pick a thickness, t:

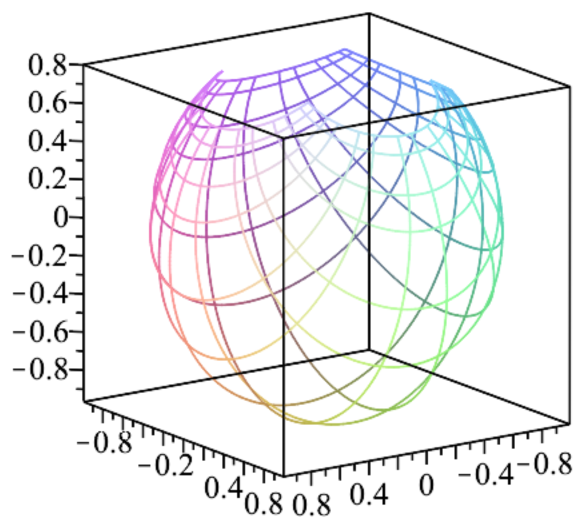
```
> t := .13:   # deviation from center line; thickness = 2*t
```

```
> plots:-display( [
      (for i from -2 to 2 by 2/3 do
           plots:-implicitplot( x=i-t, x=-2-t..2+t,y=-2-t..2+t ),
           plots:-implicitplot( x=i+t, x=-2-t..2+t,y=-2-t..2+t );
      end do),
      (for i from -2 to 2 by 2/3 do
           plots:-implicitplot( y=i-t, x=-2-t..2+t,y=-2-t..2+t ),
           plots:-implicitplot( y=i+t, x=-2-t..2+t,y=-2-t..2+t );
      end do)

  ]);
```



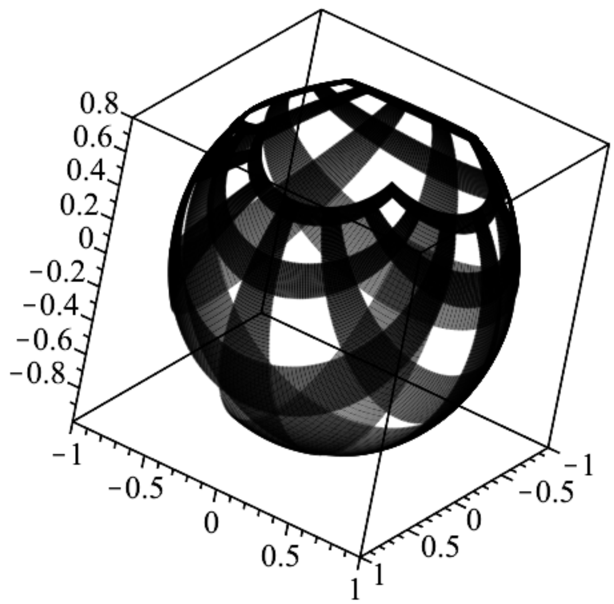

```
> plots:-display([
     (for i from -2 to 2 by 2/3 do
          plots:-spacecurve( proj(x,i-t), x=-2-t..2+t ),
          plots:-spacecurve( proj(x,i+t), x=-2-t..2+t ),
          plots:-spacecurve( proj(i-t,y), y=-2-t..2+t ),
          plots:-spacecurve( proj(i+t,y), y=-2-t..2+t );
     end do)
  ]);
```



Visualize a filled globe by shading in the curves, and revealing the cut-outs.

```
> plots:-display([
     (for i from -2 to 2 by 2/3 do
         (for j from -t to t by t/20. do
```

```
            plots:-spacecurve( proj(x,i+j), x=-2-t..2+t ),
            plots:-spacecurve( proj(i+j,y), y=-2-t..2+t )
        end do)
    end do)
],color=black,transparency=.5);
```



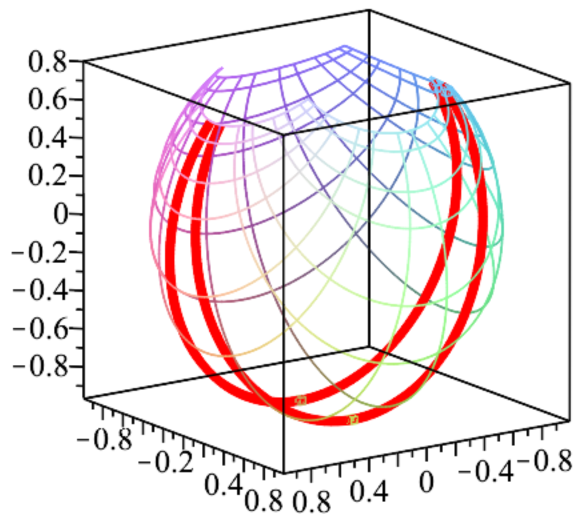
Build Plan

How to build this?
 -- look at individual arcs and measure the lengths to each intersection point

Let's look at a particular set of arcs (in red) ...

```
> plots:-display([
      (for i from -2 to 2 by 2/3 do
           plots:-spacecurve( proj(x,i-t), x=-2-t..2+t, `if`(i=0,op
  ([color=red,thickness=10]),NULL) ),
           plots:-spacecurve( proj(x,i+t), x=-2-t..2+t, `if`(i=0,op
  ([color=red,thickness=10]),NULL) ),
           plots:-spacecurve( proj(i-t,y), y=-2-t..2+t ),
           plots:-spacecurve( proj(i+t,y), y=-2-t..2+t );
     end do),
     plots:-pointplot3d([
       for pt in [[-2,0],[0,0]] do
```

```
                proj(pt[1],pt[2]-t),
                proj(pt[1],pt[2]+t);
          end do], symbol=box)
  ]);
```



The general arclength formula

```
> arclen := (L,v) -> local g; int( sqrt(add(diff(g,lhs(v))^2,g=L)),
  v);
```

$$arclen := (L, v) \to \int \sqrt{add\left((diff(g, lhs(v)))^2, g = L \right)}\, dv \tag{3.1}$$

Arclength of our traces

```
> arclen( proj(x,y), y=a..b );
```

$$\int_a^b \left(\left(\frac{16\, x^2 y^2}{(x^2+y^2+1)^4} + \left(\frac{2}{x^2+y^2+1} - \frac{4\, y^2}{(x^2+y^2+1)^2} \right)^2 + \left(\frac{2\, y}{x^2+y^2+1} - \frac{2\,(x^2+y^2-1)\, y}{(x^2+y^2+1)^2} \right)^2 \right)^{1/2} dy \tag{3.2}$$

Some sample lengths:

```
> arclen( proj(x,-t), x=-2-t..0 );   #edge to middle of bottom band
```

$$2.238417445 \tag{3.3}$$

```
> arclen( proj(x,-t), x=-2-t..2+t );   #edge to edge
```

$$4.476834890 \tag{3.4}$$

```
> arclen( proj(x,-2-t), x=-2-t..2+t );   # mouth
```

$$1.250596258 \tag{3.5}$$

```
> arclen( proj(x,2+t), x=-2-t..2+t );    # mouth
```
1.250596258 (3.6)

```
> arclen( proj(x,-t), x=-2-t..-2+t );    #thickness of top band
```
0.1039048341 (3.7)

```
> arclen( proj(x,-t), x=-t..t );    #thickness of bottom band
```
0.5085531885 (3.8)

```
> arclen( proj(-t,y), y=-2..0 );
```
2.189156225 (3.9)

Scale

The above models a unit sphere.   We'll scale it up to a size where we can build it:

The model was built to measure 6.5 inches in diameter, but I wanted the measurements in centimeters for ease of measuring.

```
> scale := convert( 6.5, units, inches, cm )/2.0;
```

$$scale := 8.255000000 \quad (4.1)$$

```
> radius := scale;  #radius
```

$$radius := 8.255000000 \tag{4.2}$$

```
> arclen := (L,v) -> local g; scale * int( sqrt(add(diff(g,lhs(v))
  ^2,g=L)),v);
```

$$arclen := (L, v) \rightarrow scale\left(\int \sqrt{add\left((diff(g, lhs(v)))^2, g = L\right)}\, dv\right) \tag{4.3}$$

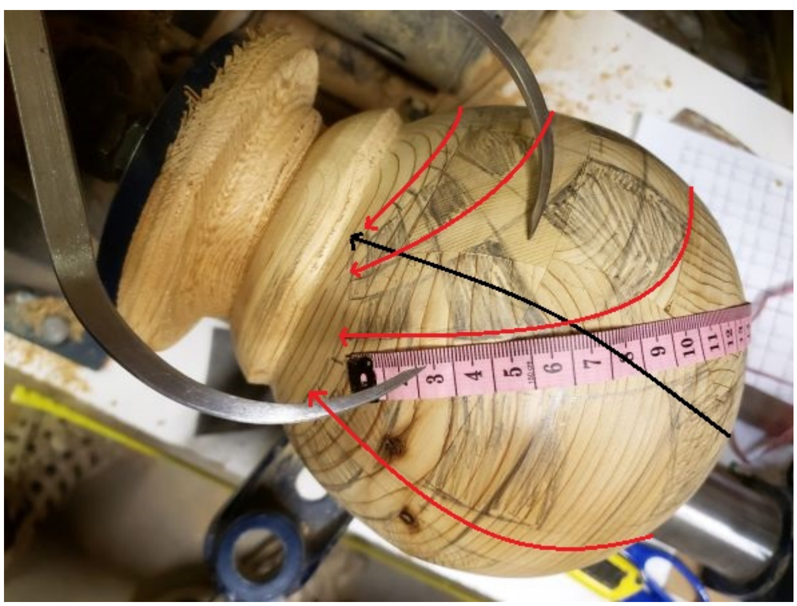
```
> c := 2*Pi*scale;  #circumference
```

$$c := 51.86769472 \tag{4.4}$$

```
> c - arclen( proj(x,-t), x=-2-t..2+t );  # arclength of opening at
  top to the mid-point of each well
```

$$14.91142270 \tag{4.5}$$

Arc Length Between Max Points



The arc lengths we found in the "Build Plan" section are shown in red in the picture above. To check our measurements, it would be good to also know the length of the black line in this picture. This will be the longest length around the globe. It will be one quarter distance to the red line that goes from the bottom of the well all the way around the sphere. The four max points, and the four well points will give a solid starting point for locating the cutouts.

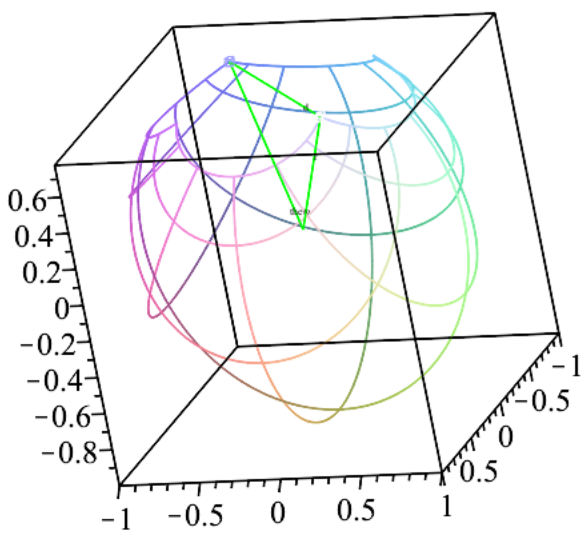
```
> p1 := proj(-2,-2); p2 := proj(2,2);
```

$$p1 := \left[-\frac{4}{9}, -\frac{4}{9}, \frac{7}{9}\right]$$

$$p2 := \left[\frac{4}{9}, \frac{4}{9}, \frac{7}{9}\right] \quad (5.1)$$

```
> plots:-display([
      (for i from -2 to 2 by 2/3 do
           plots:-spacecurve( proj(x,i), x=-2..2 ),
           plots:-spacecurve( proj(i,y), y=-2..2 );
      end do),
      plots:-pointplot3d([ p1, p2 ], symbol=box),
      plottools:-line( p1,p2, color=green ),
      plottools:-line( [0,0,0],p1, color=green ),
      plottools:-line( [0,0,0],p2, color=green ),
      plots:-textplot3d([0,0,.1,"theta"]),
      plots:-textplot3d([0.3,0.3,.5,"r"]),
      plots:-textplot3d([0.3,0.3,p1[3],"d"])

  ]);
```



The distance between these points is:

```
> d := sqrt((p1[1]-p2[1])^2 + (p1[2]-p2[2])^2);
```

$$d := \frac{8\sqrt{2}}{9} \quad (5.2)$$

The angle at the center with lines to these points is:

```
> ap := fsolve( cos(theta) = 1-d^2/(2*1^2), theta);
```

$$ap := 1.359347638 \quad (5.3)$$

The (scaled) arc length of the gap is:

```
> len := scale * ap;
```

$$len := 11.22141475 \quad (5.4)$$

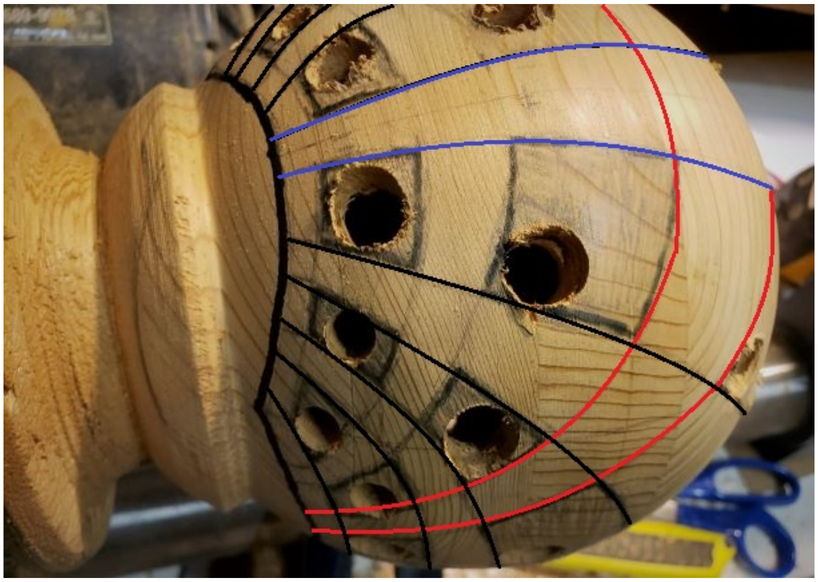
The (scaled) arc length around the globe is:

```
> evalf((2*Pi-ap)*scale);
```

40.64627997 (5.5)

>

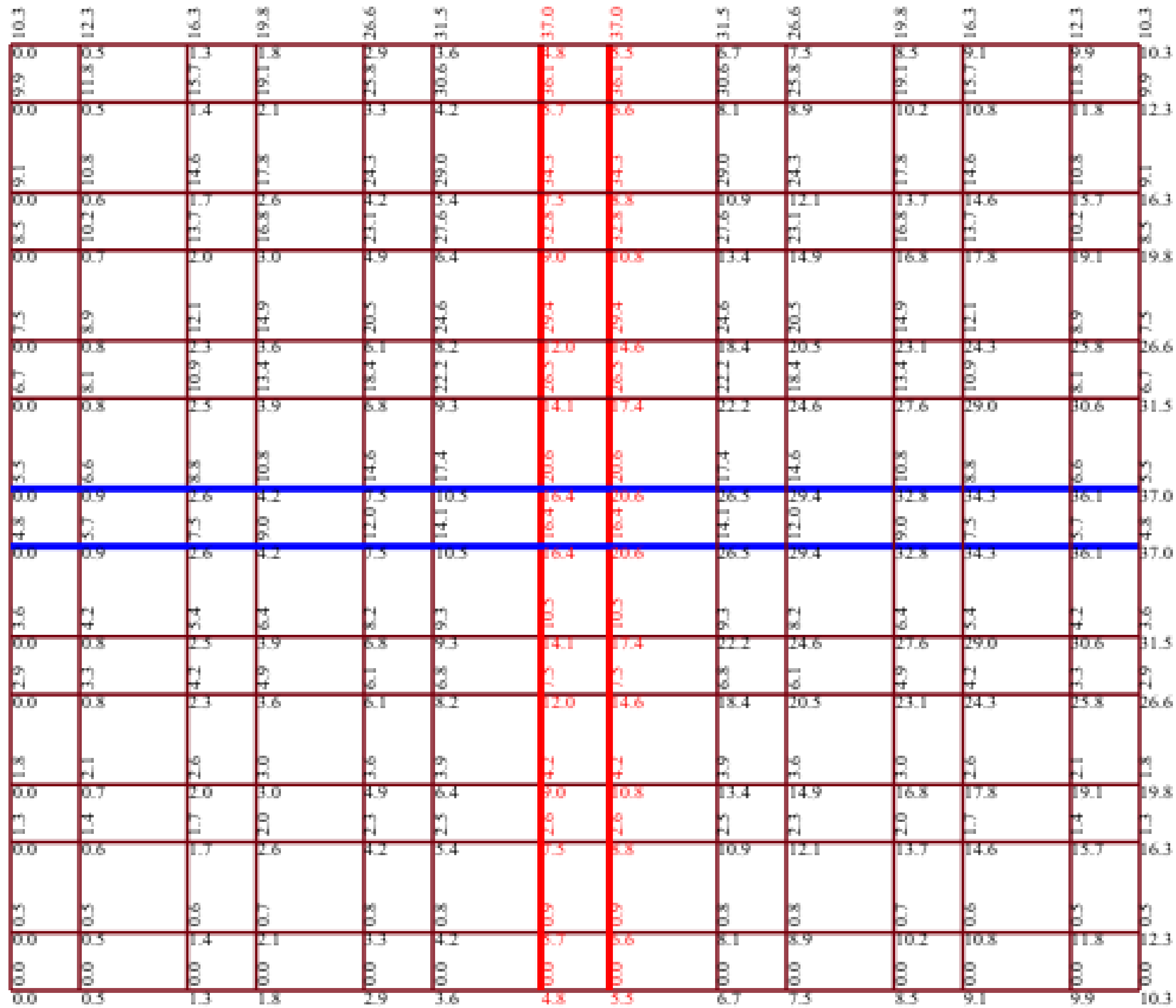
Layout



Using the arclengths of each of the inside and outside edges and their crossing points, we can generate the following map that will let us measure the cut lines on the globe. We use the widths and lengths and distances between arcs to locate the lines on the globe. The cutouts are drilled to establish a hole and the sawed and filed.

```
> plots:-display( [
      (for i from -2 to 2 by 2/3 do
           plots:-implicitplot( x=i-t, x=-2-t..2+t,y=-2-t..2+t,
  `if`(i=0,op([color=red,thickness=3]),NULL)   ),
           plots:-implicitplot( x=i+t, x=-2-t..2+t,y=-2-t..2+t,
  `if`(i=0,op([color=red,thickness=3]),NULL)   ),
           plots:-implicitplot( y=i-t, x=-2-t..2+t,y=-2-t..2+t,
  `if`(i=0,op([color=blue,thickness=3]),NULL) ),
           plots:-implicitplot( y=i+t, x=-2-t..2+t,y=-2-t..2+t,
  `if`(i=0,op([color=blue,thickness=3]),NULL) );
      end do),
      plots:-textplot([
         for i from -2 to 2 by 2/3 do
           (for j from -2 to 2 by 2/3 do
              y1 := arclen( proj(i-t,y), y=-2-t..j+t );
              y2 := arclen( proj(i+t,y), y=-2-t..j-t );
              y3 := arclen( proj(i-t,y), y=-2-t..j-t );
              y4 := arclen( proj(i+t,y), y=-2-t..j+t );
              [i-t,j+t,sprintf("%.1f",y1),`if`(i=0,color=red,NULL)
```

```
],
            [i+t,j-t,sprintf("%.1f",y2),`if`(i=0,color=red,NULL)
],
            [i-t,j-t,sprintf("%.1f",y3),`if`(i=0,color=red,NULL)
],
            [i+t,j+t,sprintf("%.1f",y4),`if`(i=0,color=red,NULL)]
          end do)
      end do],rotation=Pi/2,align={above,right}),
    plots:-textplot([
       for i from -2 to 2 by 2/3 do
          (for j from -2 to 2 by 2/3 do
            x1 := arclen( proj(x,j-t), x=-2-t..i-t );
            x2 := arclen( proj(x,j+t), x=-2-t..i+t );
            x3 := arclen( proj(x,j+t), x=-2-t..i-t );
            x4 := arclen( proj(x,j-t), x=-2-t..i+t );
            [i-t,j-t,sprintf("%.1f",x1),`if`(i=0,color=red,NULL)
],
            [i+t,j+t,sprintf("%.1f",x2),`if`(i=0,color=red,NULL)
],
            [i-t,j+t,sprintf("%.1f",x3),`if`(i=0,color=red,NULL)
],
            [i+t,j-t,sprintf("%.1f",x4),`if`(i=0,color=red,NULL)]
          end do)
      end do],align={below,right})
], axes=none,size=[1000,1000]);
```



Some quick measurements to see if the sizing will work

```
> arclen( proj(x,-2-t), x=-2-t..-2+t );    #thinnest wall
```

$$0.4502841664 \tag{6.1}$$

```
> arclen( proj(x,-t), x=-t..t );    #thickest wall
```

$$4.198106571 \tag{6.2}$$

Alternate Layout

This visualizes the lines as strips, showing the thickness at the middle of the crossing point.   Ultimately the grid above was more helpful for layout, but this better shows the proportions, and may be helpful visualizing the grid above, which is not to scale.

```
> P := Array(1..0):
  anno := Array(1..0):
  for i from -2 to 2 by 2/3 do
    line1 := Array(1..0): line2 := Array(1..0):
    for j from -2 to 2 by 2/3 do
       len1 := arclen( proj(x,i-t), x=-2-t..j );
       len2 := arclen( proj(x,i+t), x=-2-t..j );

       w := arclen( proj(x,j), x=i-t..i+t );
  #     printf("[%f,%f,%f] ",len1,len2,w);

       k := 10*(i+3):
       line1 ,= [len1, k+w/2];
       line2 ,= [len2, k-w/2];
       anno ,= [len1,k,sprintf("%0.2f",w)];
       P ,= plot([line1[-1],line2[-1]],color=yellow);
       anno ,= [len1,k+w/2,sprintf("%0.2f",len1),align=[above,
  right]];
       anno ,= [len2,k-w/2,sprintf("%0.2f",len2),align=[below,
  right]];
    end do:
    P ,= plot(convert(line1,list),color=blue):
    P ,= plot(convert(line2,list),color=green)
  end do:
  P ,= plots:-textplot(convert(anno,list)):
  plots[display]( convert(P,list), size=[1000,800], axes=none );
```