Wes's Haversin Puzzle

Functions and Unit Tests

http://en.wikipedia.org/wiki/Haversine_formula
http://en.wikipedia.org/wiki/Great-circle_distance
http://geospatialmethods.org/spheres/GCIntersect.html
http://www.bing.com/community/site_blogs/b/maps/default.aspx
http://www.bing.com/community/site_blogs/b/maps/archive/2010/05/19/new-bing-map-apps-gas-prices-distance-calculator-and-parking-finder.aspx
http://www.planetmarshall.co.uk/2010/06/drawing-geodesic-curves-using-the-bing-maps-silverlight-control/

In[1]:=
```
deg = π / 180;
```

In[2]:=
```
haversin[θ_] := Sin[θ / 2]^2
```

In[3]:=
```
haversinDistance[R_, {lat1_, lon1_}, {lat2_, lon2_}] :=
 With[{h = haversin[(lat2 - lat1)] + Cos[lat1] Cos[lat2] *
      haversin[(lon2 - lon1)]},
  2 R ArcSin[Sqrt[h]]]
```

A second of arc should be about 100 feet (30 m)

In[4]:=
```
haversinDistance[6356.14 * 1000 meter, {0, 0}, {deg / 3600, 0}]
```

Out[4]=
30.8154 meter

In[5]:=
```
haversinDistance[ll1_List, ll2_List] :=
 haversinDistance[6356.14 * 1000 meter, ll1, ll2]
```

A thousands of a degree (one in the last place in the plots above) is about 364 feet:

In[6]:=
```
haversinDistance[{0, 0}, {0.001 deg, 0}]
```

Out[6]=
110.936 meter

In[7]:=
```
haversinDistance[{0, 0}, {0.001 deg, 0}] * 1250 feet / (381 meter)
```

Out[7]=
363.962 feet

How much is a millionth of a degree (at the equator)?

In[8]:=
```
haversinDistance[{0, 0}, {0.000001 deg, 0}]
```

Out[8]=
0.110936 meter

about 11.0936 centimeters, or about

In[9]:=
```
11.0936 cm / (2.54 cm / inch)
```

Out[9]=
4.36756 inch

For more accuracy, try this: http://en.wikipedia.org/wiki/Vincenty%27s_formulae

The Puzzle

quote:

For some definition of fun and assuming that the earth is a perfect sphere with radius R, we would like to compute what segment of an arc between two points on the surface of a spherical cap described by a point and a distance.
Here are the details. We have a latitude and longitude of a hotspot, C, which is the center of a boundary. The boundary contains all points whose distance to the center, C, is less than or equal to d.
Now given two more points in latitude and longitude, A and B we want to compute a distance on the surface of the sphere between A and B which doesn't overlap with the boundary.
Lunch on me for a correct answer, an explanation, and a derivation of the haversine formula.
- Wes & Bart

end quote

In[10]:=
```
ptA = {latA, lonA};
ptB = {latB, lonB};
```

Must assume that A and B are both on the same side of the boundary, lest the geodesic connecting them cross the boundary.
If they're both on the inside of the boundary, then the distance is simply the haversin formula

In[12]:=
```
haversinDistance[R, ptA, ptB]
```

Out[12]=
$$2\,R\,\sin^{-1}\left(\sqrt{\cos(\text{latA})\cos(\text{latB})\sin^2\left(\frac{\text{lonB}-\text{lonA}}{2}\right)+\sin^2\left(\frac{\text{latB}-\text{latA}}{2}\right)}\right)$$

If they're both on the outside, then the geodesic between them may be required to "ride" the boundary for a while. To find the

WLOG

WLOG (without loss of generality), imagine the unit sphere and that point C is the North Pole, with coordinates {0, 0, 1}.

In[13]:=
```
latLonToXYZ[R_, {lat_, lon_}] :=
 {R Cos[lat] Cos[lon],
  R Cos[lat] Sin[lon],
  R Sin[lat]}
```

The set of points of constant radius 1 that go through A and B define the great circle connecting them:

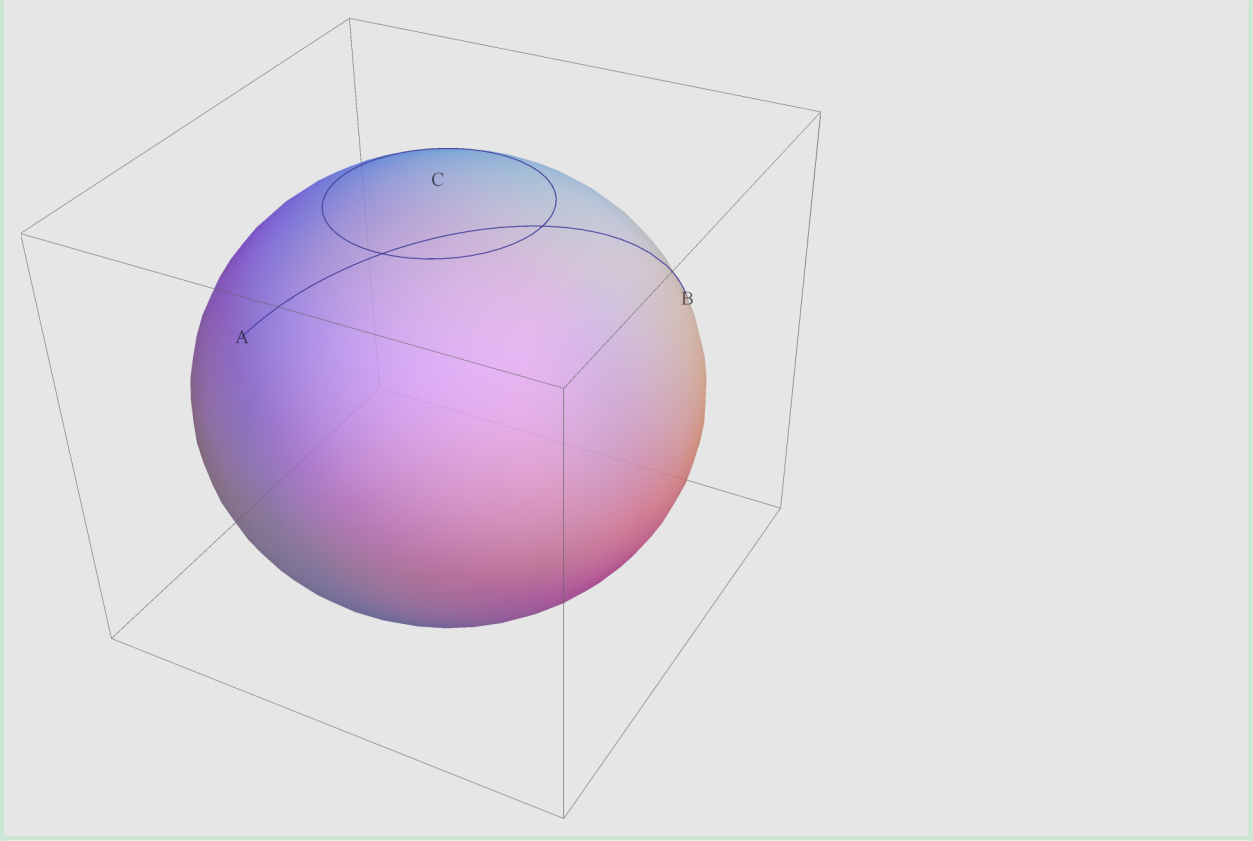
The equation of a Great Circle in the form of latitude as a function of longitude, easy for parametric rendering

In[14]:=
```
latC[lonC_, {lat1_, lon1_}, {lat2_, lon2_}] :=
 With[{
   y = Sin[lat1] Cos[lat2] Sin[lon2 - lonC] - Sin[lat2] Cos[lat1] Sin[lon1 - lonC],
   x = Cos[lat1] Cos[lat2] Sin[lon2 - lon1]},
  ArcTan[x, y]]
```

The circumpolar arc is simply the set of points of constant latitude as longitude varies from 0 through 360.

In[15]:=
```
With[{
  pA = {35, -120} deg,
  pB = {18, 40} deg,
  Radius = 1},
 With[{
   A = latLonToXYZ[1, pA],
   B = latLonToXYZ[1, pB],
   Center = {0, 0, 0},
   Pole = {0, 0, 1}},
  Show[
   Graphics3D[{Opacity[0.6250], Sphere[Center],
     Text["A", A],
     Text["B", B],
     Text["C", Pole]
    },
    AspectRatio → 1],
   (* The circumpolar boundary *)
   ParametricPlot3D[
    latLonToXYZ[Radius, {65, lon} deg],
    {lon, 0, 360},
    AspectRatio → 1],
   (* The Great-Circle semi-arc *)
   ParametricPlot3D[
    latLonToXYZ[Radius, {latC[lon, pA, pB], lon}],
    {lon, pA[[2]], pB[[2]]}]
  ]]]
```

Out[15]=



Finding the Intersection Points

The following is the lat as a function of lon for the great circle:

In[16]:=
```
latC[LON, ptA, ptB]
```

Out[16]=

$$\tan^{-1}(-\cos(\text{latA})\cos(\text{latB})\sin(\text{lonA}-\text{lonB}),\ \cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))$$

Solve for the longitude for a given LAT to find the intersection points:

(Don't run this -- never finishes)

In[17]:=
```
(* Solve[LAT==latC[LON,ptA,ptB],LON] *)
```

In[18]:=
```
LAT == latC[LON, ptA, ptB]
```

Out[18]=

$$\text{LAT}=\tan^{-1}(-\cos(\text{latA})\cos(\text{latB})\sin(\text{lonA}-\text{lonB}),\ \cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))$$

Try a numerical solution

In[19]:=
```
numericalRule = {
  LAT → 65.0 deg,
  latA → 35.0 deg,
  lonA → -120.0 deg,
  latB → 18.0 deg,
  lonB → 40.0 deg}
```

Out[19]=
$$\{\text{LAT} \to 1.13446,\ \text{latA} \to 0.610865,\ \text{lonA} \to -2.0944,\ \text{latB} \to 0.314159,\ \text{lonB} \to 0.698132\}$$

In[20]:=
```
numExact = {
  LAT → 65 deg,
  latA → 35 deg,
  lonA → -120 deg,
  latB → 18 deg,
  lonB → 40 deg}
```

Out[20]=
$$\left\{\text{LAT} \to \frac{13\pi}{36},\ \text{latA} \to \frac{7\pi}{36},\ \text{lonA} \to -\frac{2\pi}{3},\ \text{latB} \to \frac{\pi}{10},\ \text{lonB} \to \frac{2\pi}{9}\right\}$$

In[21]:=
```
LAT == latC[LON, ptA, ptB] /. numExact
```

Out[21]=
$$\frac{13\pi}{36} = \tan^{-1}\left(\sqrt{\frac{5}{8}+\frac{\sqrt{5}}{8}}\ \sin\left(\frac{\pi}{9}\right)\cos\left(\frac{7\pi}{36}\right),\right.$$
$$\left.\frac{1}{4}\left(\sqrt{5}-1\right)\cos\left(\frac{7\pi}{36}\right)\cos\left(\text{LON}+\frac{\pi}{6}\right)-\sqrt{\frac{5}{8}+\frac{\sqrt{5}}{8}}\ \sin\left(\frac{7\pi}{36}\right)\sin\left(\text{LON}-\frac{2\pi}{9}\right)\right)$$

This one takes too long ... I didn't let it finish

In[22]:=
```
(* Solve[LAT==latC[LON,ptA,ptB]/.numExact,LON] *)
```

This was as close as we could get using MMA's tools. This tells us that, in a C# implementation, we must (A) consider the principal branches of the trig functions (B) probably resort to a custom tailored Newton's method. But I'm convinced we can do it :)

In[23]:=
```
NSolve[
 LAT == latC[LON, ptA, ptB] /. numericalRule,
 LON,
 Reals]
```

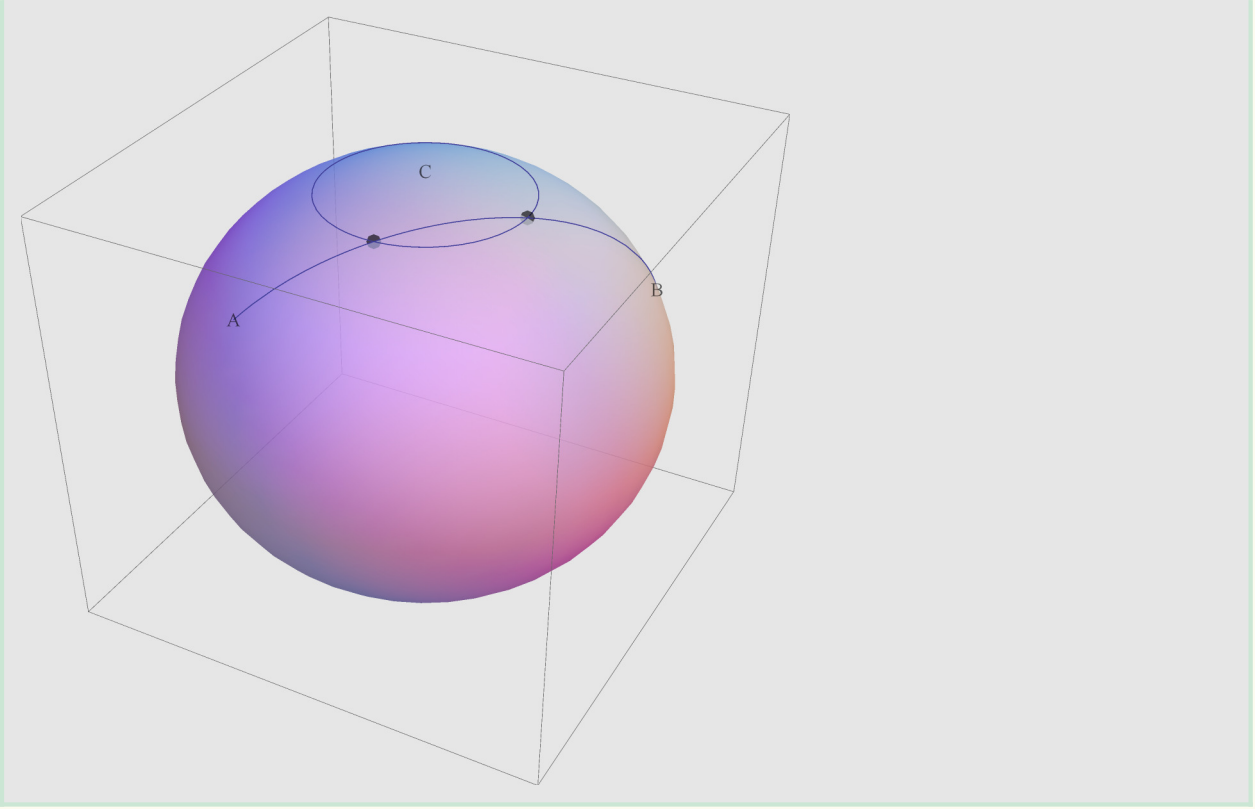
NSolve::ratnz : NSolve was unable to solve the system with inexact coefficients. The answer was obtained by solving a corresponding exact system and numericizing the result. ≫

Out[23]=
$$\{\{\text{LON} \to \text{ConditionalExpression}[1.\,(6.28319\,c_1 - 1.52233),\ c_1 \in \mathbb{Z}]\},\ \{\text{LON} \to \text{ConditionalExpression}[1.\,(6.28319\,c_1 - 0.00285341),\ c_1 \in \mathbb{Z}]\}\}$$

Let's see if their plausible, by copying the solution by hand into the following and plotting

In[24]:=
```
With[{
  lat = 65 deg,
  pA = {35, -120} deg,
  pB = {18, 40} deg,
  Radius = 1},
 With[{
   A = latLonToXYZ[1, pA],
   B = latLonToXYZ[1, pB],
   Center = {0, 0, 0},
   Pole = {0, 0, 1}},
  Show[
   Graphics3D[{Opacity[0.6250], Sphere[Center],
     Text["A", A],
     Text["B", B],
     Text["C", Pole],
     PointSize[Large],
     Point[latLonToXYZ[Radius, {lat, (6.28319 - 1.52233)}]],
     Point[latLonToXYZ[Radius,
       {lat, (6.28318 - 0.00285340)}]]
    },
    AspectRatio → 1],
   (* The circumpolar boundary *)
   ParametricPlot3D[
    latLonToXYZ[Radius, {lat, lon}],
    {lon, 0, 360 deg},
    AspectRatio → 1],
   (* The Great-Circle semi-arc *)
   ParametricPlot3D[
    latLonToXYZ[Radius, {latC[lon, pA, pB], lon}],
    {lon, pA[[2]], pB[[2]]}]
  ]]]
```

Out[24]=



Pop the cork and declare victory

Failed Junk

AMissing Simplification Formulas

Hard to believe, but Mathematica apparently doesn't know the following:

In[25]:=

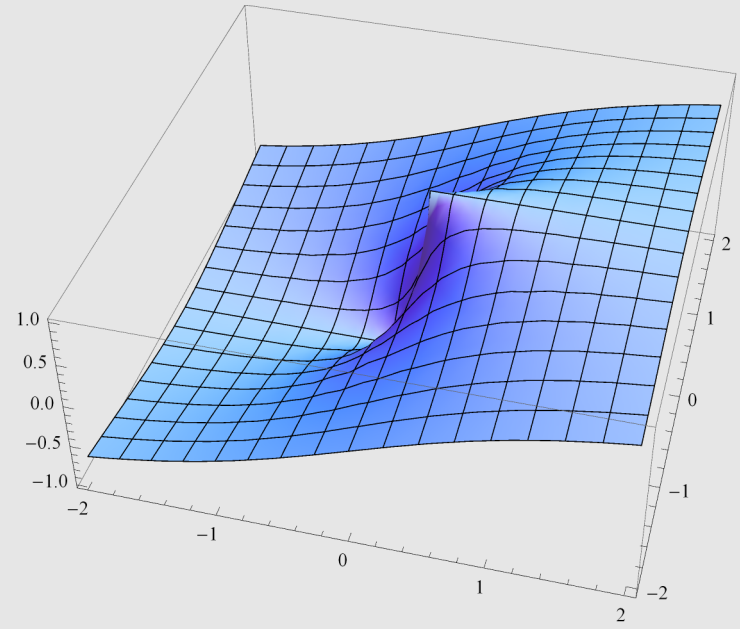
```
Cos[ArcTan[x, y]] /. {Cos[ArcTan[x_, y_]] → x / Sqrt[x^2 + y^2]}
```

Out[25]=

$$\frac{x}{\sqrt{x^2 + y^2}}$$

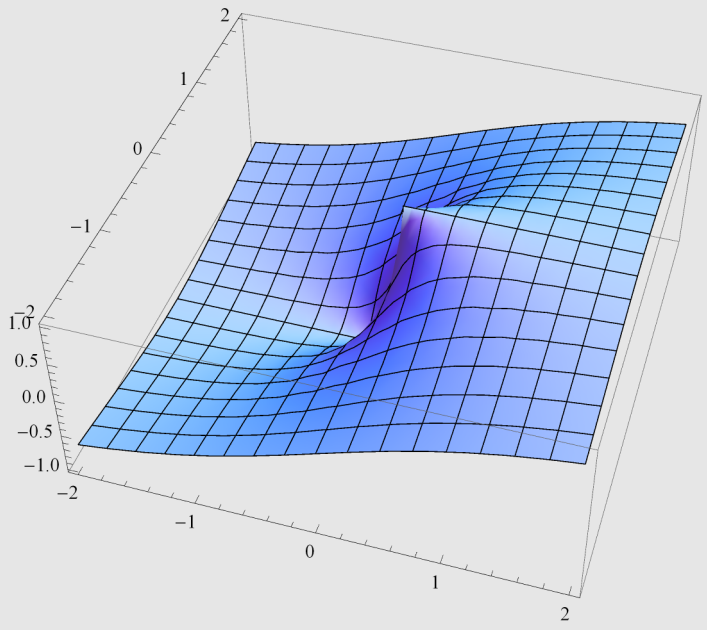
In[26]:=
```
Plot3D[Cos[ArcTan[x, y]], {x, -2, 2}, {y, -2, 2}]
```

Out[26]=



In[27]:=
```
Plot3D[x / Sqrt[x^2 + y^2], {x, -2, 2}, {y, -2, 2}]
```

Out[27]=



Break it up

In[28]:=
```
expr1 = latLonToXYZ[1, {LAT, LON}]
```

Out[28]=
$$\{\cos(\text{LAT})\cos(\text{LON}),\ \cos(\text{LAT})\sin(\text{LON}),\ \sin(\text{LAT})\}$$

In[29]:=
```
(expr2 = latLonToXYZ[1, {latC[LON, ptA, ptB], LON}]) // FullSimplify
```

Out[29]=
$$\{\cos(\text{LON})\cos\left(\tan^{-1}(-\cos(\text{latA})\cos(\text{latB})\sin(\text{lonA}-\text{lonB}),\ \cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))\right),$$
$$\sin(\text{LON})\cos\left(\tan^{-1}(-\cos(\text{latA})\cos(\text{latB})\sin(\text{lonA}-\text{lonB}),\ \cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))\right),$$
$$\sin\left(\tan^{-1}(-\cos(\text{latA})\cos(\text{latB})\sin(\text{lonA}-\text{lonB}),\ \cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))\right)\}$$

In[30]:=
```
(expr3 = (expr2 /. {Cos[ArcTan[x_, y_]] → x/Sqrt[x^2 + y^2],
    Sin[ArcTan[x_, y_]] → y/Sqrt[x^2 + y^2]})) // FullSimplify
```

Out[30]=
$$\{-(\cos(\text{latA})\cos(\text{latB})\cos(\text{LON})\sin(\text{lonA}-\text{lonB}))\Big/\left(\sqrt{((\cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))^2+\cos^2(\text{latA})\cos^2(\text{latB})\sin^2(\text{lonA}-\text{lonB})}\right),$$
$$-(\cos(\text{latA})\cos(\text{latB})\sin(\text{LON})\sin(\text{lonA}-\text{lonB}))\Big/\left(\sqrt{((\cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))^2+\cos^2(\text{latA})\cos^2(\text{latB})\sin^2(\text{lonA}-\text{lonB})}\right),$$
$$(\cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))\Big/\left(\sqrt{((\cos(\text{latA})\sin(\text{latB})\sin(\text{LON}-\text{lonA})-\sin(\text{latA})\cos(\text{latB})\sin(\text{LON}-\text{lonB}))^2+\cos^2(\text{latA})\cos^2(\text{latB})\sin^2(\text{lonA}-\text{lonB})}\right)\}$$

Let's try a numerical solution

In[31]:=
```
numericalRule = {
  LAT → 65.0 deg,
  latA → 35.0 deg,
  lonA → -120.0 deg,
  latB → 18.0 deg,
  lonB → 40.0 deg}
```

Out[31]=
$$\{\text{LAT} \to 1.13446,\ \text{latA} \to 0.610865,\ \text{lonA} \to -2.0944,\ \text{latB} \to 0.314159,\ \text{lonB} \to 0.698132\}$$

In[32]:=
```
expr1 /. numericalRule
```

Out[32]=
$$\{0.422618\cos(\text{LON}),\ 0.422618\sin(\text{LON}),\ 0.906308\}$$

In[33]:=
```
expr1 == expr3 /. numericalRule
```

Out[33]=
$$\{0.422618\cos(\text{LON}),\ 0.422618\sin(\text{LON}),\ 0.906308\} = \left\{\frac{0.266454\cos(\text{LON})}{\sqrt{(0.545504\sin(0.698132-\text{LON})+0.253132\sin(\text{LON}+2.0944))^2+0.0709978}},\ \frac{0.266454\sin(\text{LON})}{\sqrt{(0.545504\sin(0.698132-\text{LON})+0.253132\sin(\text{LON}+2.0944))^2+0.0709978}},\ \frac{0.545504\sin(0.698132-\text{LON})+0.253132\sin(\text{LON}+2.0944)}{\sqrt{(0.545504\sin(0.698132-\text{LON})+0.253132\sin(\text{LON}+2.0944))^2+0.0709978}}\right\}$$

Numerical Solve not able to find an answer :(

In[34]:=
```
NSolve[expr1 == expr3 /. numericalRule, LON]
```

NSolve::ifun : Inverse functions are being used by NSolve, so
some solutions may not be found; use Reduce for complete solution information. >>

Out[34]=
$$\{\}$$

Hopeless for regular Solve

In[35]:=
```
Solve[expr1 == expr3 /. numericalRule, LON]
```

Solve::ifun : Inverse functions are being used by Solve, so
some solutions may not be found; use Reduce for complete solution information. >>

Out[35]=
$$\{\}$$

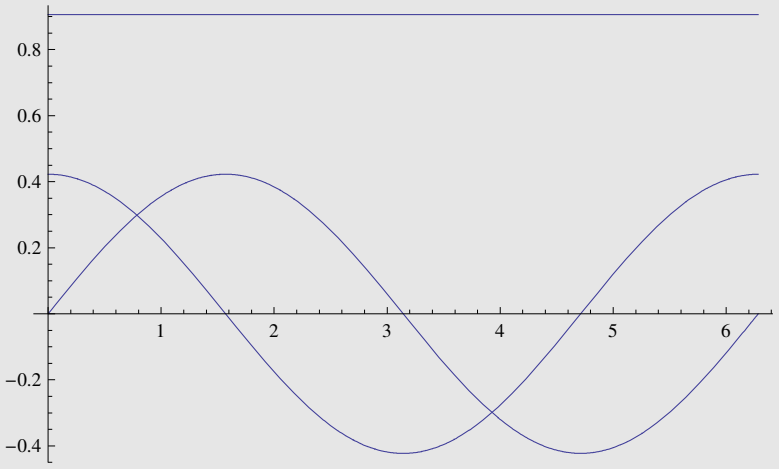
Try by hand

In[36]:=
```
expr1
```

Out[36]=
$$\{\cos(\text{LAT})\cos(\text{LON}),\ \cos(\text{LAT})\sin(\text{LON}),\ \sin(\text{LAT})\}$$

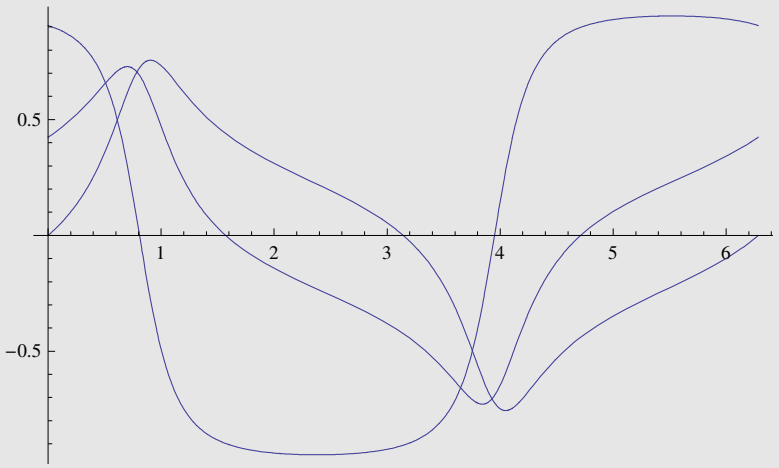
In[37]:=
```
Plot[expr1 /. numericalRule, {LON, 0, 360 deg}]
```

Out[37]=



In[38]:=
```
Plot[expr3 /. numericalRule, {LON, 0, 360 deg}]
```

Out[38]=



Find intersections by component: