xyztp[(θ, ϕ}] gives (x, y, z) in terms of longitude θ and colatitude ϕ; this is the mathematics convention for θ and ϕ, the reverse of the physics convention

In[1]:=
```
sind[t_] := Sin[t Degree];
cosd[t_] := Cos[t Degree];
xyztp[{θ_, ϕ_}] := {cosd[θ] sind[ϕ], sind[θ] sind[ϕ], cosd[ϕ]};
```
$$\texttt{unit[v\_] :=} \frac{\texttt{v}}{\sqrt{\texttt{v.v}}};$$

TestTriples consists of 5000 normalized triples, each sorted by decreasing size.

In[5]:=
```
TestTriples = Table[Reverse[Sort[
      unit[{RandomReal[{-1, 1}], RandomReal[{-1, 1}], RandomReal[{-1, 1}]}]]], {i, 5000}];
```

The first two items in TestTriples :

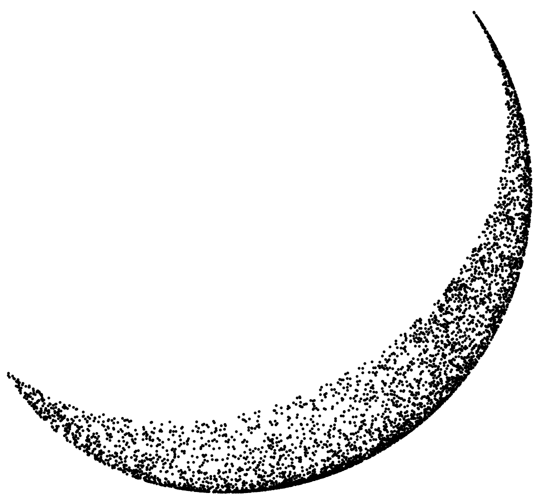
In[6]:=
```
Take[TestTriples, 2]
```

Out[6]=
```
{{0.789812, 0.435182, 0.432219}, {0.723483, 0.690263, -0.0104308}}
```

TestTriples plotted in R^3.   Moving the result with the mouse makes it clear that this is a lune of the sphere.

In[7]:=
```
Graphics3D[{PointSize[.004], Point /@ TestTriples}, Boxed → False]
```

Out[7]=



So you can always plot a sorted and normalized eigenvalue triple as is. But the result is awkward, since the isotropic axis is not vertical.

In[8]:=
$$\texttt{uG} = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{3} & 0 & -\sqrt{3} \\ -1 & 2 & -1 \\ \sqrt{2} & \sqrt{2} & \sqrt{2} \end{pmatrix};$$

The rotation matrix uG makes the isotropic axis vertical and takes the $(1, 0, -1)$ direction to the $(1, 0, 0)$ direction :

In[9]:=
```
uG.{1, 1, 1}
uG.{1, 0, -1}
```

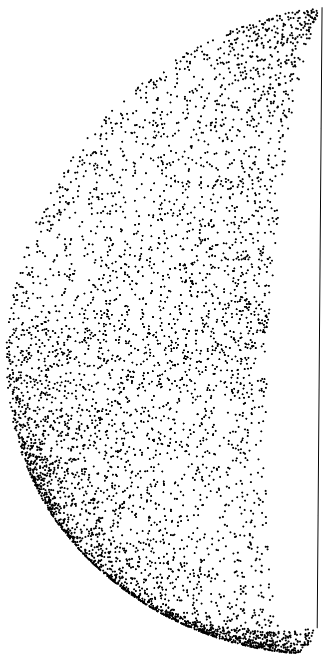
Out[9]= $\{0, 0, \sqrt{3}\}$

Out[10]= $\{\sqrt{2}, 0, 0\}$

So if you apply uG to your eigenvalue triples you get a more palatable picture :

In[11]:=
```
Graphics3D[{Line[{{0, 0, 1}, {0, 0, -1}}],
   {PointSize[.004], Point[uG.#] & /@ TestTriples}},
 Boxed → False, ViewPoint → 7 xyztp[{40, 70}]]
```

Out[11]=



Once the lune has been rotated, gamma and beta coordinate curves coincide with contours for ordinary longitude θ and colatitude ϕ (again, math convention)

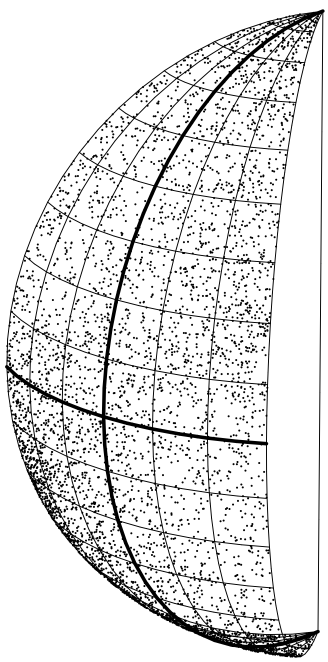
In[12]:=
```
LuneColatLineRotated[ϕ_] := Table[xyztp[{θ, ϕ}], {θ, -30, 30, 5}];
LuneLongLineRotated[θ_] := Table[xyztp[{θ, ϕ}], {ϕ, 0, 180, 5}];
```

In[14]:=
```
LuneColatLinesRotated = Table[LuneColatLineRotated[ϕ], {ϕ, 10, 170, 10}];
LuneLongLinesRotated = Table[LuneLongLineRotated[θ], {θ, -30, 30, 10}];
```

In[16]:=
```
Graphics3D[{Line /@ LuneColatLinesRotated}]
Graphics3D[{Line /@ LuneLongLinesRotated}]
```

In[18]:=
```
Graphics3D[{Line[{{0, 0, 1}, {0, 0, -1}}],
  Line /@ LuneColatLinesRotated,
  Line /@ LuneLongLinesRotated,
  {Thickness[.007], Line[LuneColatLineRotated[90]], Line[LuneLongLineRotated[0]]},
  {PointSize[.004], Point[uG.#] & /@ TestTriples}},
 Boxed → False, ViewPoint → 7 xyztp[{40, 70}]]
```

Out[18]=



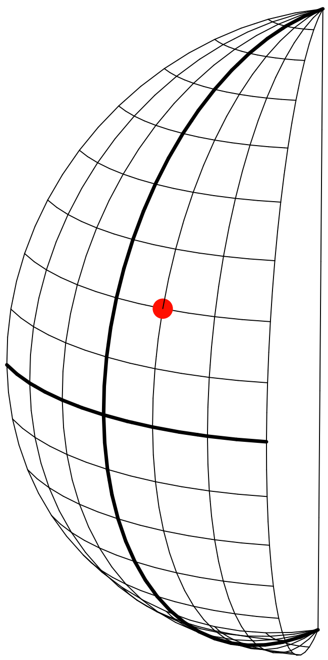
Suppose you want to plot the following Λ, which is already sorted and normalized.

In[19]:=
```
Λ = Transpose[uG].xyztp[{10., 70.}]
```

Out[19]=
```
{0.785217, 0.330698, -0.523519}
```

If you want the lune to be vertical, then you plot uG·Λ instead of Λ

In[20]:=
```
Graphics3D[{Line[{{0, 0, 1}, {0, 0, -1}}],
  Line /@ LuneColatLinesRotated,
  Line /@ LuneLongLinesRotated,
  {Thickness[.007], Line[LuneColatLineRotated[90]], Line[LuneLongLineRotated[0]]},
  {Hue[.01], PointSize[.04], Point[uG.Λ]}}, Boxed → False, ViewPoint → 7 xyztp[{40, 70}]]
```

Out[20]=


From our Eq (20) you can see that Λ above was chosen so that γ = 10 and β =70. And indeed that is where it plots on the lune.