David Roman Garcia
UROP 2022
July 1, 2022

Cross Track Distance

Algorithm

Consider an arbitrary point B and a given great circle arc both on the surface of the same unit sphere centered at the origin. To calculate the shortest distance from each other along the sphere, let $\vec{P}$ be the vector going from the origin to B, and $\vec{Q}$ and $\vec{R}$ be the vectors from the origin to endpoints of the arc. We can then use the following algorithm:

$$\vec{S} = \hat{Q} \times \hat{R} \qquad \text{Find the normal to the great circle and call it } \vec{S}.$$

$$\hat{P} \cdot \vec{S} = |\vec{S}| \cos\beta \qquad \text{Project } \vec{P} \text{ onto } \vec{S}.$$

$$\cos\beta = \frac{\hat{P} \cdot \vec{S}}{|\vec{S}|} \qquad \text{Solve for } \cos\beta.$$

$$\cos\beta = \sin\alpha \qquad \text{Use right angle trigonometry.}$$

$$\alpha = \arcsin\left(\frac{\hat{P} \cdot \vec{S}}{|\vec{S}|}\right) \qquad \text{Solve for } \alpha.$$

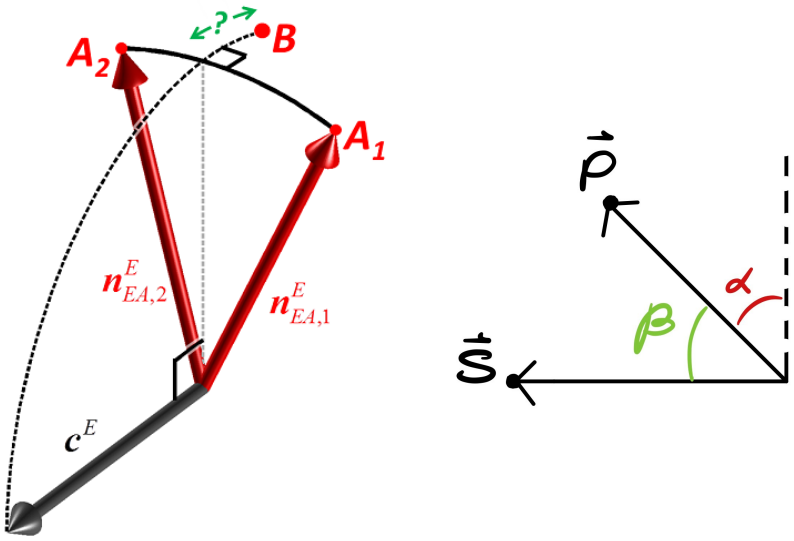

FIGURE 1. Great circle distance to a point.

Figure 1 shows two viewpoints. The left one considers the full 3D representation of the sphere and the points that lie on its surface. The right hand side shows its projection on the plane spanned by the normal of the arc segment joining $\vec{Q}$ and $\vec{R}$, and an arbitrary point on the surface whose coordinates are given by the vector $\vec{P}$.

See the last example on [1] for a sample code and [2] for an explanation of the method.

Implementation

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (
    AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

import numpy as np
import math
```

```
import astropy.units as u
from astropy.coordinates import SkyCoord


def cross_track_distance(lat1, long1, lat2, long2, lat3, long3):
    c1 = SkyCoord(ra=long1*u.radian, dec=lat1*u.radian, frame='icrs')
    c2 = SkyCoord(ra=long2*u.radian, dec=lat2*u.radian, frame='icrs')
    c3 = SkyCoord(ra=long3*u.radian, dec=lat3*u.radian, frame='icrs')

    # radec2icrf function is provided at a previous section of satelllites.py
    P1 = radec2icrf(c1.ra.radian, c1.dec.radian, deg=False)
    P2 = radec2icrf(c2.ra.radian, c2.dec.radian, deg=False)
    P3 = radec2icrf(c3.ra.radian, c3.dec.radian, deg=False)

    S = np.cross(P1,P2)
    dot = np.dot(S,P3)

    dist = np.arcsin(dot/np.linalg.norm(S))

    return dist
```

LISTING 1. Python 3 implementation of figure 1.

REFERENCES

[1] Norwegian Defence Research Establishment. The n-vector page. https://www.ffi.no/en/research/n-vector/, 2010.

[2] L. Strous. Compute minimum distance between point and great arc on sphere. https://math.stackexchange.com/questions/337055/compute-minimum-distance-between-point-and-great-arc-on-sphere, 2018.