24 We live in a spherical world

10 points

Introduction

Although lack of education, impossibility of access to intellectual works and some Christian expression of ideas affirming the Earth was flat… make it difficult to tell what wider population in early Middle Ages thought (if they even thought about this ;-), the fact is that the Earth is spherical. This idea first appeared in Greek philosophy with Pythagoras in the 6th century BC, and Aristotle provided the first observations supporting this idea, like the round Earth shadow over the Moon during lunar eclipses.

(*) Image not original from Aristotle

Flat trigonometry is not accurate when calculating distances between two points in the surface of the Earth, very important in navigation.
Luckily, the havesine formula allows us to calculate the shortest orthodromic distance or shorter distance between two points in the surface of a sphere.

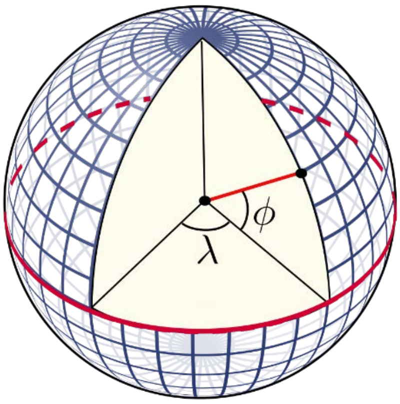
This can be very helpful, for example, if you are trying to win the Barcelona World Race, a non-stop, round-the-world yacht race, and look for the shortest route…
Looking at this map, it would seem that the horizontal distance on the Equator is similar to the one down on the 45ties latitude (in red, clipper route):

but a more realistic view, can help you see that maybe 'souther is shorter':

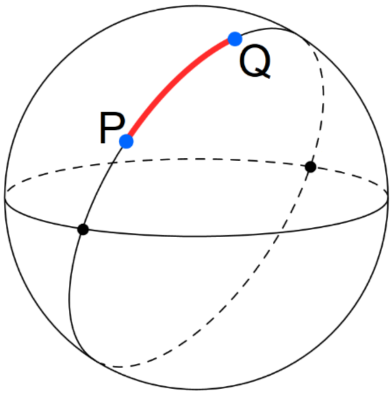
As an example, the circle of latitude 0°, that's the Equator, measures 40030Km (21600nm); while the Antarctic Circle, at latitude 66°33′ S, measures just 17662Km (9536nm)

Haversine formula

Latitude and longitude are basically nothing more than angles. Latitude is measured as your degrees north or south of the equator. Longitude is your degrees east or west of the prime meridian. The combination of these two angles pinpoints an exact location on the surface of the earth.



2- A perspective view of the Earth showing latitude (φ) and longitude (λ)



1- A diagram illustrating great-circle distance (drawn in red) between two points on a sphere, P and Q

As shown in the image *2* above, the quickest route between two points on the surface of the earth is a "great circle path" - in other words, a path that comprises a part of the longest circle you could draw around the globe that intersects the two points.

The shortest distance between two points on the globe can be calculated using the Haversine formula shown below.

$$d = 2r \arcsin\left(\sqrt{\text{haversin}(\phi_2 - \phi_1) + \cos(\phi_1)\cos(\phi_2)\,\text{haversin}(\lambda_2 - \lambda_1)}\right)$$
$$= 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Where:

- *d* is the orthodromic distance between two points
- r is the radius of the sphere
- $\phi 1$, $\phi 2$: latitude of point 1 and latitude of point 2, in radians
- $\lambda 1$, $\lambda 2$: longitude of point 1 and longitude of point 2, in radians

Program specification

Your program should read two coordinates in latitude, longitude (float number in degrees), and calculate the orthodromic distance between them in meters, without decimals (truncated output).
We'll assume that Earth is perfectly spherical, with radius r=6371Km

Latitude will be positive for north hemisphere and negative for south.
Longitude will be positive for East and negative for West
(*) So, no indication for N/S or E/W is needed, and the calculation is direct from the values.

Input

```
latitude1,longitude1
latitude2,longitude2
```

Example

```
41.471485,2.094249
41.43166,2.126039
```

Output

```
distance
```

Example

```
5160
```

Solution

```
from math import radians, cos, sin, asin, sqrt

def haversine(point1, point2):
    """ Calculate the great-circle distance bewteen two points on the Earth surface.

    :input: two 2-tuples, containing the latitude and longitude of each point
    in decimal degrees.

    Example: haversine((45.7597, 4.8422), (48.8567, 2.3508))

    :output: Returns the distance bewteen the two points in kilometers
    """
    AVG_EARTH_RADIUS = 6371000  # in m

    # unpack latitude/longitude
    lat1, lng1 = point1
    lat2, lng2 = point2

    # convert all latitudes/longitudes from decimal degrees to radians
    lat1, lng1, lat2, lng2 = map(radians, (lat1, lng1, lat2, lng2))

    # calculate haversine
    lat = lat2 - lat1
    lng = lng2 - lng1
    d = sin(lat * 0.5) ** 2 + cos(lat1) * cos(lat2) * sin(lng * 0.5) ** 2
    h = 2 * AVG_EARTH_RADIUS * asin(sqrt(d))
    return h  # in kilometers


# Read the input coordinates
# lat,lon
coor1=[float(x) for x in input().split(',')]
coor2=[float(x) for x in input().split(',')]

print(int(haversine(coor1,coor2)))
```