Questions Jobs Documentation BETA Tags Users Log In Sign Up

x Dismiss

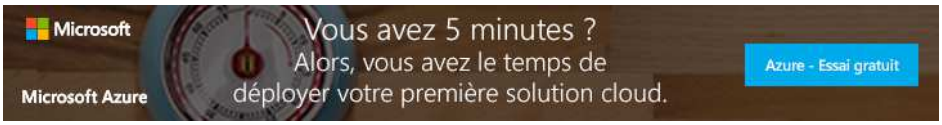
Join the Stack Overflow Community

Stack Overflow is a community of 6.7 million programmers, just like you, helping each other.
Join them; it only takes a minute:

Sign up

Haversine Formula in Python (Bearing and Distance between two GPS points)

Ask Question



Problem

I would like to know how to get the distance and bearing between 2 GPS points. I have researched on the haversine formula. Someone told me that I could also find the bearing using the same data.

Edit

Everything is working fine but the bearing doesn't quite work right yet. The bearing outputs negative but should be between 0 - 360 degrees. The set data should make the horizontal bearing `96.02166666666666` and is:

```
Start point: 53.32055555555556 , -1.7297222222222221

Bearing:  96.02166666666666

Distance: 2 km

Destination point: 53.31861111111111, -1.6997222222222223

Final bearing: 96.04555555555555
```

Here is my new code:

```
from math import *

Aaltitude = 2000
Oppsite  = 20000

lat1 = 53.32055555555556
lat2 = 53.31861111111111
lon1 = -1.7297222222222221
lon2 = -1.6997222222222223

lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

dlon = lon2 - lon1
dlat = lat2 - lat1
a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
c = 2 * atan2(sqrt(a), sqrt(1-a))
Base = 6371 * c


Bearing =atan2(cos(lat1)*sin(lat2)-sin(lat1)*cos(lat2)*cos(lon2-lon1), sin(lon2-
lon1)*cos(lat2))

Bearing = degrees(Bearing)
print ""
print ""
print "--------------------"
print "Horizontal Distance:"
print Base
print "--------------------"
print "Bearing:"
print Bearing
print "--------------------"


Base2 = Base * 1000
distance = Base * 2 + Oppsite * 2 / 2
Caltitude = Oppsite - Aaltitude

a = Oppsite/Base
b = atan(a)
```

```
print "The degree of vertical angle is:"
print c
print "--------------------"
print "The distance between the Balloon GPS and the Antenna GPS is:"
print distance
print "--------------------"
```

python gps distance haversine bearing

edited Jul 28 '13 at 13:15

asked Feb 6 '11 at 12:41

avitex
814 2 10 21

Python haversine implementation can be found codecodex.com/wiki/…. However for short distance calculations very simple ways exists. Now, what is your maximum distance expected? Are you able to get your co-ordinates in some local cartesian co-ordinate system? – eat Feb 6 '11 at 13:15

1 @James Dyson: with distances like 15km, creat circle doesen't count anything. My suggestion: figure out first the solution with euclidean distances! That will give you a working solution and then later if your distances will be much much longer, then adjust your application. Thanks – eat Feb 6 '11 at 22:30

Can I also find the Bearing from this equation? – avitex Feb 6 '11 at 22:42

1 @James Dyson: If your above comment was aimed to me (and at to my earlier suggestion), the answer is surely (and quite 'trivially' as well). I may be able to give some example code, but it won't utilize trigonometry, rather geometry (so I'm unsure if it will help you at all. Are you familiar at all with the concept of vector? In your case positions and directions could be handled most straightforward manner with vectors). – eat Feb 6 '11 at 23:04

1 `atan2(sqrt(a), sqrt(1-a))` is the same as `asin(sqrt(a))` – user102008 Dec 7 '11 at 1:44

7 Answers

Here's a Python version:

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

def haversine(lon1, lat1, lon2, lat2):
    """
    Calculate the great circle distance between two points
    on the earth (specified in decimal degrees)
    """
    # convert decimal degrees to radians
    lon1, lat1, lon2, lat2 = map(radians, [lon1, lat1, lon2, lat2])

    # haversine formula
    dlon = lon2 - lon1
    dlat = lat2 - lat1
    a = sin(dlat/2)**2 + cos(lat1) * cos(lat2) * sin(dlon/2)**2
    c = 2 * asin(sqrt(a))
    r = 6371 # Radius of earth in kilometers. Use 3956 for miles
    return c * r
```

edited Feb 9 '15 at 13:31

answered Feb 6 '11 at 13:47

Michael Dunn
4,312 1 26 46

6 Could use math.radians() function instead of multiplying by pi/180 - same effect, but a bit more self-documenting. – Hugh Bothwell Feb 6 '11 at 15:10

Thanks soooo much, My distance can vary (horizontaly) between 15-1km – avitex Feb 6 '11 at 20:19

3 You can, but if you say `import math` then you have to specify `math.pi`, `math.sin` etc. With `from math import *` you get direct access to all the module contents. Check out "namespaces" in a python tutorial (such as docs.python.org/tutorial/modules.html) – Michael Dunn Feb 6 '11 at 21:20

2 How come you use atan2(sqrt(a), sqrt(1-a)) instead of just asin(sqrt(a))? Is atan2 more accurate in this case? – Eyal Jul 25 '11 at 16:34

1 should be float division to cover really rare corner case of dlat|dlon being integers: a = sin(dlat/2.)**2 + cos(lat1) * cos(lat2) * sin(dlon/2.)**2 – Dmitriy Jul 23 '14 at 18:21

The bearing calculation is incorrect, you need to swap the inputs to atan2.

```
    bearing = atan2(sin(long2-long1)*cos(lat2), cos(lat1)*sin(lat2)-
sin(lat1)*cos(lat2)*cos(long2-long1))
    bearing = degrees(bearing)
    bearing = (bearing + 360) % 360
```

This will give you the correct bearing.

answered Apr 30 '15 at 3:03

Some implementations in python:

- http://code.activestate.com/recipes/576779-calculating-distance-between-two-geographic-points/
- http://www.platoscave.net/blog/2009/oct/5/calculate-distance-latitude-longitude-python/

answered Feb 6 '11 at 13:40

Fábio Diniz
5,892 1 22 35

You can solve the negative bearing problem by adding 360°. Unfortunately, this might result in bearings larger than 360° for positive bearings. This is a good candidate for the modulo operator, so all in all you should add the line

```
Bearing = (Bearing + 360) % 360
```

at the end of your method.

answered Aug 30 '13 at 20:42

OBu
2,062 1 8 26

I think it's just: Bearing = Bearing % 360 – Holger Bille Dec 21 '15 at 8:59

I don't have the python translation, but her are the formulas you need.

http://www.movable-type.co.uk/scripts/latlong.html

edited Aug 30 '13 at 20:46

corsiKa
54.3k 19 109 164

answered Feb 6 '11 at 13:05

rcravens
6,857 2 19 22

Thanks alot, I found this one before but with the code above can understand it better. – avitex Feb 6 '11 at 22:21

You can try the following:

```
from haversine import haversine
haversine((45.7597, 4.8422),(48.8567, 2.3508),miles = True)
243.71209416020253
```

edited Jun 23 '15 at 20:09

Leo Correa
9,103 1 23 54

answered Jun 23 '15 at 18:58

Vamshi G
97 1 3

How can this be used in a Django's ORM query? – Gocht Aug 21 '15 at 23:00

The Y in atan2 is, by default, the first parameter. Here is the documentation. You will need to switch your inputs to get the correct bearing angle.

```
bearing = atan2(sin(lon2-lon1)*cos(lat2), cos(lat1)*sin(lat2)in(lat1)*cos(lat2)*cos(lon2-
lon1))
bearing = degrees(bearing)
bearing = (bearing + 360) % 360
```

answered Jan 7 '16 at 20:04

gisdude
11 2
