

[x Dismiss](#)

Join the Stack Overflow Community

Stack Overflow is a community of 7.1 million programmers, just like you, helping each other.

Join them; it only takes a minute:

[Sign up](#)

Calculating angle between two Great Circles on a sphere

I expected this to be simple, but I'm getting some strange results. I'd appreciate if someone could point out what I'm doing wrong.

I have 3 points (A, B, C) defined on the surface of the Earth (assumed to be a sphere) with [lat, long] coordinates for each point. I need to calculate the angle between the two Great Arcs formed by AC and AB.

I already have a function that calculates the Great Circle Distance (GCD), so I decided to solve this problem by getting the GCD for AC, AB and CA, reducing them to a unit sphere and then applying the Spherical Law of Cosines to get at the angle BAC.

This appeared to work and gave me reasonable angles. However, I then tried to put all three points on the same Great Circle and a strange thing started happening. If B and C were within 1 degree, the results were reasonable, but as I started moving B and C further apart along the same Great Circle, the angle started growing!

For example:

```
A = 49, 1
B = 49, 10    => Angle: 0.0378
C = 49, 10.1
```

```
A = 49, 1
B = 49, 10    => Angle: 0.2270
C = 49, 10.6
```

```
A = 49, 1
B = 49, 10    => Angle: 3.7988
C = 49, 20
```

```
A = 49, 1
B = 49, 10    => Angle: 99.1027
C = 49, 200
```

Is this some sort of precision error, or is my formula wrong?

Here is the code (`getDistance()` is known to work):

```
public static BigDecimal getAngle(
    final BigDecimal commonLat, final BigDecimal commonLong,
    final BigDecimal p1Lat, final BigDecimal p1Long,
    final BigDecimal p2Lat, final BigDecimal p2Long) {

    // Convert real distances to unit sphere distances
    //
    double a = getDistance(p1Lat, p1Long, commonLat, commonLong).doubleValue() /
RADIUS_EARTH;
    double b = getDistance(p2Lat, p2Long, commonLat, commonLong).doubleValue() /
RADIUS_EARTH;
    double c = getDistance(p1Lat, p1Long, p2Lat, p2Long).doubleValue() /
RADIUS_EARTH;

    // Use the Spherical law of cosines to get at the angle between a and b
    //
    double numerator = Math.cos(c) - Math.cos(a) * Math.cos(b);
    double denominator = Math.sin(a) * Math.sin(b);
    double theta = Math.acos(numerator / denominator);

    // Back to degrees
    //
    double angleInDegrees = Math.toDegrees(theta);

    return new BigDecimal(angleInDegrees);
}
```

```
}
```

Unfortunately for me, my application will often have points nearly on a line, so accuracy in this situation is important. What is going wrong here?

EDIT: As requested, here is the code for `getDistance()` :

```
public static BigDecimal getDistance(final BigDecimal endLat, final BigDecimal
endLong,
    final BigDecimal startLat, final BigDecimal startLong) {

    final double latDiff = Math.toRadians(endLat.doubleValue() -
startLat.doubleValue());
    final double longDiff = Math.toRadians(endLong.doubleValue() -
startLong.doubleValue());

    final double lat1 = Math.toRadians(startLat.doubleValue());
    final double lat2 = Math.toRadians(endLat.doubleValue());

    double a =
        Math.sin(latDiff / 2) * Math.sin(latDiff / 2) +
        Math.sin(longDiff / 2) * Math.sin(longDiff / 2) * Math.cos(lat1) *
Math.cos(lat2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
    double d = RADIUS_EARTH * c;

    return new BigDecimal(d);
}
```

The declaration of `RADIUS_EARTH` is irrelevant, b/c we multiply by it in distance calculation and then divide by it in angle calculation, so it is cancelled out.

java geometry geospatial precision

edited Nov 10 '11 at 6:35

asked Nov 10 '11 at 6:08

 Val Blant
658 2 9 23


You should add code for `getDistance` code and `RADIUS_EARTH` declaration – RC. Nov 10 '11 at 6:21

Okay so for one you are taking a big decimal and then treating it as a double ... you may be losing accuracy there. Have a look at stackoverflow.com/questions/2173512/... it points out a math library that does arbitrary precision trigonometry – Ahmed Masud Nov 10 '11 at 6:26

1 Answer

A quick look on your coordinates say, that the latitude ist the same and longitude varies. But all circles made by latitude (except equator) aren't great circles. Did you try your program, if longitude is constant and latitude is varied?

answered Nov 10 '11 at 7:28

 Mnemenenth
27.3k 34 118 185

Thanks, Mnemenenth. I can't believe I missed that :). – Val Blant Nov 10 '11 at 18:51