Subject: Re: What is the problem ?
Posted by penteado on Wed, 28 Oct 2009 11:20:21 GMT
View Forum Message <> Reply to Message

On Oct 28, 4:28 am, Ruby <wuqiao...@gmail.com> wrote:
> I wrote a simple program to caculate the distance between two location
> (lon1,lat1), (lon2,lat2)
>
> function earth_dis ,lon1,lon2,lat1,lat2
>
> b1 = !pi*lat1/180.0
> b2 = !pi*lat2/180.0
> a1 = !pi*lon1/180.0
> a2 = !pi*lon2/180.0
>
> dis = 6378.1*acos(cos(b1)*cos(b2)*cos(a1-a2)+sin(b1)*sin(b2))
>
> RETURN,dis
> END
>
> But When I tried to test the program, the results turned to be like
> IDL> print,earth_dis(4.0,4.0,4.0,4.0)
> -NaN
> IDL> print,earth_dis(8.0,8.0,8.0,8.0)
> 2.20215
>
> In both cases, the result should be straightforwardly equal 0. Then
> what is the problem with my program or IDL ?

First, this is unnecessary. There is the built-in function map_2points
that does it (and even comes with a default radius).

Second, the problem is precision. To begin with, do not ever do this
kind of calculation with floats, as you did. Use doubles.

When you get an NaN, it happens because the roundoff errors make cos
(b1)*cos(b2)*cos(a1-a2)+sin(b1)*sin(b2) slighly larger than 1, so
there is no acos of it. The other small number is still the result of
these errors.

If you use doubles, the problem will be smaller, but this expression
will always have problems when the two points are near 0 or 180
degrees. I suppose map_2points already deals with this in some way,
and it is probably better to use instead of writing this yourself.

If you must deal with this kind of problem yourself, one possible
solution is to change your function to

```
function earth_dis ,lon1,lon2,lat1,lat2
b1 = !dpi*lat1/180d0
b2 = !dpi*lat2/180d0
a1 = !dpi*lon1/180d0
a2 = !dpi*lon2/180d0
dis =  6378.1d0*acos(complex(cos(b1)*cos(b2)*cos(a1-a2)+sin(b1)*sin (b2),
0d0))
RETURN,double(dis)
END
```