

Вычисление постоянного азимута и длины линии румба между двумя точками для геодезических координат

Решения задачи нахождения постоянного азимута (вдоль линии румба), используемый код может применяться в других расширениях.

[Обсудить в форуме](#) Комментариев — 2

Линия румба или локсодром - линия пересекающая меридианы под постоянным углом. Линия румба - прямая линия в проекции Меркатора, на сфере - линия румба представляет из себя сферическую спираль, сходящуюся концентрическими кругами к полюсам. Для того, чтобы вычислить угол линии румба между двумя произвольными точками, можно воспользоваться следующим скриптом.

Все параллели являются локсодромами, так как пересекают меридианы под постоянным углом 90 градусов.

Данный угол не равен углу т.н. начального азимута, т.е. углу, под которым надо выйти из начальной точки, чтобы потом, следуя по линии большого круга (то есть кратчайшим расстоянием) прийти в конечную точку. При следовании по линии большого круга, направление постоянно изменяется.

[Скачать пример расчета в Excel](#)

Вычислениям приведенным выше соответствует следующий код на языке Avenue (координаты двух точек передаются отдельно, [процедуру вызова скрипта](#) расчета см. ниже) ([скачать скрипт расчета азимута](#)):

```
pi = 3.14159265358979
pi2 = Pi / 2
lat1 = pnt1.getY*pi/180
lat2 = pnt2.getY*pi/180
long1 = -1*pnt1.getX*pi/180
long2 = -1*pnt2.getX*pi/180
dlon_W = (long2 - long1) - (2*pi*(((long2 - long1)/(2*pi)).floor))
dlon_E = (long1 - long2) - (2*pi*(((long1 - long2)/(2*pi)).floor))
dphi = (((lat2/2) + (pi/4)).tan)/(((lat1/2) + (pi/4)).tan).ln

if ((lat2 - lat1).abs < 0.00000001) then
    q = lat1.cos
else
    q = (lat2 - lat1)/dphi
end

if (dlon_W < dlon_E) Then
    dlon_W = -1*dlon_W
    'get sign
    if (dlon_W >= 0) then
        sign = 1
    else
        sign = -1
    end
    If ((dlon_W.abs) >= (dphi.abs)) Then
        Atn2 = (sign * pi2) - (dphi / dlon_W).atan
    Else
        If (dphi > 0) Then
            Atn2 = (dlon_W / dphi).atan
        Else
            If ((-1*dlon_W) >= 0) Then
                Atn2 = Pi + (dlon_W / dphi).atan
            Else
                Atn2 = (-1*Pi) + (dlon_W / dphi).atan
            End
        End
    End
```

```

    End
End
else
    'get sign
    if (dlon_W >= 0) then
        sign = 1
    else
        sign = -1
    end
    If ((dlon_E.abs) >= (dphi.abs)) Then
        Atn2 = sign * pi2 - (dphi / (dlon_E)).atan
    Else
        If (dphi > 0) Then
            Atn2 = ((dlon_E) / dphi).atan
        Else
            If ((dlon_E) >= 0) Then
                Atn2 = Pi + ((dlon_E) / dphi).atan
            Else
                Atn2 = (-1*Pi) + ((dlon_E) / dphi).atan
            End
        End
    End
End
dlon = dlon_E
end
tc = atn2 - (2*pi*((atn2)/(2*pi)).floor))
dist = ((q^2)*(dlon^2) + ((lat2-lat1)^2)).sqrt
'результат - угол в градусах
tcdeg = (tc*180)/pi
'результат - расстояние в метрах
dism = dist*6372795
reslist = {tcdeg, dism}
return reslist

```

Для вызова процедуры расчета азимутов приведенной выше, можно воспользоваться следующим скриптом, результатом его работы будет расчет азимутов от точки testpoint на все точки активной темы вида и запись результата в поле Newang атрибутивной таблицы этой темы:

```

atheme = av.getactivedoc.getactivethemes.get(0)
aftab = atheme.getftab
f_shape = aftab.findfield("Shape")
f_ang = aftab.findfield("Loxang")
f_dist = aftab.findfield("Loxdist")

'testpoint - точка отсчета
testpoint = point.make(25.85, 55.15)
aftab.seteditable(true)

'для каждой точки темы на которые считаем азимуты от точки отсчета
for each rec in aftab
    pnts = {}
    apoint = aftab.returnvalue(f_shape, rec)
    pnts.add(apoint.getx)
    pnts.add(testpoint.getx)
    pnts.add(apoint.gety)
    pnts.add(testpoint.gety)

    'Вызов процедуры расчета азимута
    '"Calc-azimuth" - название скрипта с процедурой в проекте
    loxval = av.run("Calc-azimuth", pnts)
    aftab.setvalue(f_ang, rec, loxval.get(0))
    aftab.setvalue(f_dist, rec, loxval.get(1))
end
aftab.seteditable(false)

```

Следует обратить особое внимание на порядок исходной и конечной точки, т.е. точки от которой берется азимут и точки на которую он берется, если получаемые значения представляют собой 180 + угол или 180 -

угол - попробуйте поменять местами входные широты и долготы, порядок в данном случае имеет значение. Например поменяв фрагмент предыдущего скрипта следующим образом:

```
' ...  
    pnts.add(testpoint.getx)  
    pnts.add(apoint.getx)  
    pnts.add(testpoint.gety)  
    pnts.add(apoint.gety)  
' ...
```

[Обсудить в форуме](#) Комментариев — 2

Ссылки по теме

- [Вычисление расстояния и начального азимута между двумя точками на сфере](#)
- [Степанов Н.Н. Сферическая тригонометрия](#)
- [Law of cosines \(spherical\)](#)

Последнее обновление: November 27 2010

Дата создания: 30.12.2006

Автор(ы): [Максим Дубинин](#)