

Задание 1-6.

Расчет областей захваченных, локально-захваченных и высыпающих частиц для прогноза радиационной опасности.

Выполнил Лапин Ярослав. 14/01/2011.

Введение

Частицы захваченные магнитным полем Земли будут колебаться между точками отражения и дрейфовать вдоль долготы. При этом движении будут выполняться условия сохранения первого и второго адиабатических инвариантов. Дрейфовыми траекториями будут линии которые соответствуют одинаковым значениям инвариантов (таким образом мы “как бы” считаем, что это движение одной и той же частицы). Чтобы построить такие линии воспользуемся следующим алгоритмом: мы будем брать фиксированные значения поля B_m (таким образом мы используем закон сохранения первого инварианта, сохранение магнитного момента $\mu = -\frac{mV^2}{2B}$). Далее для каждой точки на поверхности Земли мы будем считать значение второго инварианта. Таким образом если мы построим изоконтурные карты поля значений J_2 , то линии одинакового значения инварианта и будут линиями дрейфовой траекторий.

Для определения зоны А фактически необходимо проделать процедуру контуризации самостоятельно. Для одной и той же частицы (одинаковые значения адиабатических инвариантов) нам нужно посчитать высоты отражения и отметить опускается ли она ниже 100 км. Для этого при обработке каждой точки (широта, долгота, B_m) мы будем отмечать в каком промежутке значений J_2 находится эта точка и опускается ли высота ниже 100 км. Причём отмечать нужно отдельно зоны в северном и южном магнитном полушарии. Тогда алгоритм определения того, какой зоне принадлежит зона следующий: для данной точки смотрим на какой высоте лежит точка отражения, если ниже 100 км, то это зона С, дальше считаем J_2 и смотрим есть ли при данном значении J_2 в противоположном полюсе точки отражения ниже 100 км. Если есть, то зона В, если нету, то зона А.

Код

```
program task6
external igrf_gsm, t89c
real :: xx(1000), yy(1000), zz(1000), len
      open (unit=1, file='data.dat')

R0=1.+300./6371.
```

```

R11=1.+100./6371.
RLIM=60.0
IOPT=1

call recalc(2000,90,1,1,1)
do long = -179, 180
do lat = -63,63
  print *, long, lat
  THETA = (90-lat)*3.14/180.0
  PHI = long*3.14/180.0
  call sphcar(R0,THETA,PHI,XGEO,YGEO,ZGEO,1)
  call geogsm(XGEO,YGEO,ZGEO,XGSM,YGSM,ZGSM,1)
  if (lat.lt.0) then
    dir = -1.
  else
    dir = 1.
  end if

    call igrf_gsm(XGSM,YGSM,ZGSM,HX,HY,HZ)
call trace(XGSM,YGSM,ZGSM,dir,RLIM,R11,IOPT,PARMOD,T89C,
  _ IGRF_GSM,XF,YF,ZF,XX,YY,ZZ,L)
if (sqrt(xf**2+yf**2+zf**2).gt.2.) then
  print *, 'Error: line from ', lat, long, ' isn''t closed'
  print *, sqrt(xf**2+yf**2+zf**2)
  stop 1
end if
call igrf_gsm(xf,yf,zf,HXf,HYf,HZf)
field = sqrt(HX**2+HY**2+HZ**2)
field_delta = field - sqrt(HXf**2+HYf**2+HZf**2)

!! len = int (1-B/Bm)*ds
len = 0.
Bm = sqrt(HX**2+HY**2+HZ**2) ! or on R11?
do i=1, L-1
  ds = sqrt((xx(i)-xx(i+1))**2+(yy(i)-yy(i+1))**2+
    _ (zz(i)-zz(i+1))**2)
  call igrf_gsm(xx(i),yy(i),zz(i),Hxx,Hyy,Hzz)
  B = sqrt(Hxx**2+Hyy**2+Hzz**2)

```

```

        if (B.lt.Bm) then
len = len + sqrt(1-B/Bm)*ds
        end if
    end do

    write (1, '(2i8,4f10.2)') long, lat, field, field_delta, len
end do
c  write (1, *) ''
end do
end program

```

Программа для визуализации

```

#!/usr/bin/python

from matplotlib.mlab import griddata
import numpy as np
import matplotlib.pyplot as plt

def plotone(x,y,z):
    #z = np.maximum( z, 0.001); z = np.minimum( z, 10 );

    xi = np.linspace(-180,180, 200)
    yi = np.linspace(-90,90, 100)
    zi = griddata(x, y, z, xi, yi)

    CS = plt.contour(xi,yi,zi,15,linewidths=0.5,colors='k')
    CS = plt.contourf(xi,yi,zi,15,cmap=plt.cm.jet)
    plt.xlim(np.min(xi),np.max(xi))
    plt.ylim(np.min(yi),np.max(yi))
    plt.colorbar() # draw colorbar

plt.figure(num=None, figsize=(12, 10), dpi=90, facecolor='w', edgecolor='k')

data = np.genfromtxt("data.dat")

x = data[:,0]

```

```

y = data[:,1]

print x

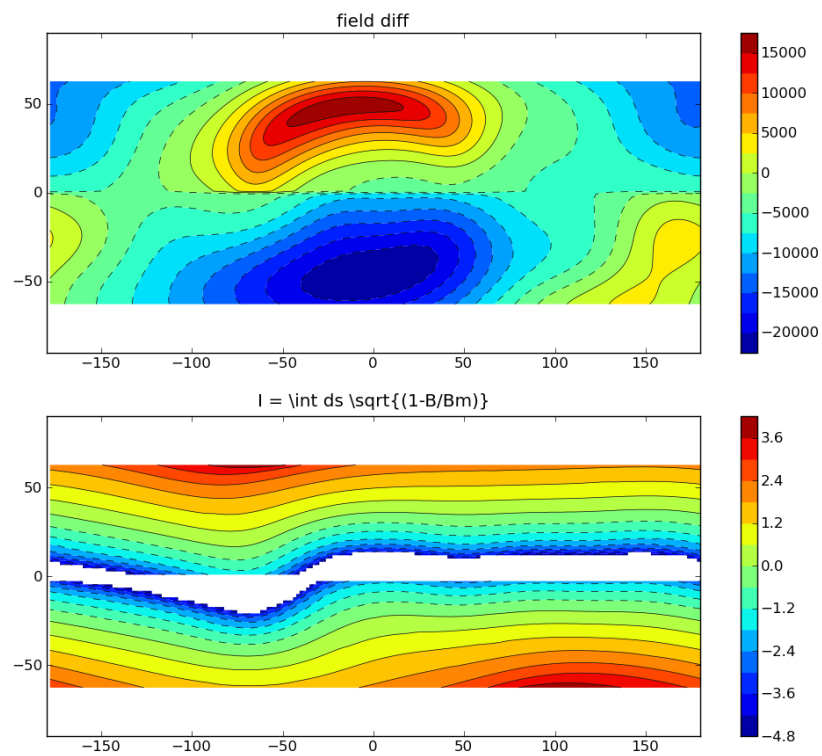
plt.subplot(2, 1, 1);plotone(x,y,data[:,3])
plt.title('field diff')

plt.subplot(2, 1, 2);plotone(x,y,np.log(data[:,4]))
plt.title('I = \int ds \sqrt{(1-B/B_m)}')

plt.savefig("lines.png")

```

Результат



Вывод