Aula 22

Movimento balístico à escala planetária:

integração temporal com Coriolis;

coordenadas esféricas

Movimento em coordenadas esféricas

$$
\begin{cases}
\dfrac{du}{dt} = fv - f'w - \dfrac{uw}{R} + \dfrac{uv\,tan(\phi)}{R} \\
\dfrac{dv}{dt} = -fu - \dfrac{u^2\,tan(\phi)}{R} - \dfrac{vw}{R} \\
\dfrac{dw}{dt} = g + f'u + \dfrac{u^2 + v^2}{R} \\
\dfrac{dx}{dt} = u \\
\dfrac{dy}{dt} = v \\
\dfrac{dz}{dt} = w
\end{cases}
$$

ϕ: latitude

λ: longitude

$R = R_T + z$

$$g = -\frac{GM_T}{R^2}$$

$$dx = R\cos(\phi)\;d\lambda$$

$$dy = R\,d\phi$$

$$dz = dR$$

import

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from math import pi
import cartopy.crs as ccrs
import imageio
import os
```

Projeção cartográfica

```
domap=True; delta=100 #passo entre frames
doplots=True
plt.close('all')
if domap:
    plt.figure(10)
    projection =ccrs.Orthographic(\
        central_longitude=0,\
        central_latitude=-30,globe=None)
    map = plt.axes(projection=projection)
    map.set_global()
    map.stock_img()
    map.coastlines()
    data_crs=ccrs.PlateCarree()
```

constantes

```
g=9.8065
phi=40;
G=6.67259e-11 #gravitação
siderealD=86164.09054 #dia sideral
Omega=2*pi/siderealD
RT=6371.e3;MT=5.9722e24 #raio e massa Terra
phi0=38.;lam0=-9.;z0=0000 #posição inicial
```

preliminares

```
timeInt=35000 #tempo de integração
if doplots:
    fig=plt.figure(1)
    ax=fig.add_subplot(111, projection='3d')
cores=['blue','green','red']

nIMP=1 #Número de iterações
aP=0.5 #aP=0.5: ponto médio; aP=0: Euler
a=1-aP
kc=-1
```

Preliminares (2)

```
for dt in[1]:
    kc=kc+1
    n=round(timeInt/dt)
    tempo=np.arange(0.,timeInt,dt)
    LAM=np.zeros((n),dtype='float')*float('nan')
    PHI=np.copy(LAM) #Alocação das variáveis
    Z=np.copy(LAM)
    U=np.copy(LAM)
    V=np.copy(LAM)
    W=np.copy(LAM)
    EM=np.copy(LAM)

    LAM[0]=lam0 #condições iniciais
    PHI[0]=phi0
    phi=phi0
    Z[0]=z0
    U[0]=u0
    V[0]=v0
    W[0]=w0
```

Integração

$$\frac{du}{dt} = fv - f'w - \frac{uw}{R} + \frac{uv\,tan(\phi)}{R}$$

Euler

```
for kt in range(1,n): #integração
    sinphi=np.sin(PHI[kt-1]*pi/180)
    cosphi=np.cos(PHI[kt-1]*pi/180)
    tanphi=np.tan(PHI[kt-1]*pi/180)
    g=G*MT/(RT+Z[kt-1])**2 #aceleraçao da gravidade
    R=RT+Z[kt-1]
    cfg=Omega**2*(RT+Z[kt-1]) #aceleção centrífuga
    cfgZ=cfg*cosphi
    cfgY=0#cfg*sinphi
    f=2*Omega*sinphi #Coriolis f
    fP=2*Omega*cosphi #Coriolis f'
    U[kt]=U[kt-1]+(f*V[kt-1]-fP*W[kt-1]-U[kt-1]*W[kt-1]\
          /R+U[kt-1]*V[kt-1]*tanphi/R)*dt
    Uh=aP*U[kt]+a*U[kt-1] #ponto médio
    V[kt]=V[kt-1]+(-f*Uh-Uh*Uh*tanphi/R-V[kt-1]*W[kt-1]/R+cfgY)*dt
    Vh=aP*V[kt]+a*V[kt-1]
    W[kt]=W[kt-1]+(fP*Uh-g+(Uh*Uh+Vh*Vh)/R+cfgZ)*dt
    Wh=aP*W[kt]+a*W[kt-1]
    LAM[kt]=LAM[kt-1]+Uh*dt/(R*cosphi)*180/pi
    PHI[kt]=PHI[kt-1]+Vh*dt/R*180/pi
    Z[kt]=Z[kt-1]+Wh*dt
```

$$\frac{dv}{dt} = -fu - \frac{u^2\,tan(\phi)}{R} - \frac{vw}{R}$$

Iteração (ponto médio)

```
for improve in range(nIMP): #Iteração
    sinphiP=np.sin(PHI[kt]*pi/180)
    cosphiP=np.cos(PHI[kt]*pi/180)
    f=2*Omega*(aP*sinphiP+a*sinphi)
    fP=2*Omega*(aP*cosphiP+a*cosphi)
    R=RT+aP*Z[kt]+a*Z[kt-1]

    tanphiH=aP*np.tan(PHI[kt]*pi/180)+a*tanphi
    gP=G*MT/(RT+Z[kt])**2

    cfgP=Omega**2*(RT+Z[kt])
    cfgZ=a*cfg*cosphi+aP*cfgP*cosphiP

    U[kt]=U[kt-1]+(f*Vh+fP*Wh-Uh*Wh/R+Uh*Vh*tanphiH/R)*dt
    Uh=aP*U[kt]+a*U[kt-1]
    V[kt]=V[kt-1]+(-f*Uh-Uh*Uh*tanphiH/R-Vh*Wh/R+cfgY)*dt
    Vh=aP*V[kt]+a*V[kt-1]
    W[kt]=W[kt-1]+(fP*Uh-(a*g+aP*gP)+(Uh*Uh+Vh*Vh)/R+cfgZ)*dt
    Wh=aP*W[kt]+a*W[kt-1]

    LAM[kt]=LAM[kt-1]+Uh*dt/(R*(a*cosphi+aP*cosphiP))*180/pi
    PHI[kt]=PHI[kt-1]+Vh*dt/R*180/pi
    Z[kt]=Z[kt-1]+Wh*dt
```

Cálculos finais

```
    EM=0.5*(U**2+V**2+W**2)-G*MT/(Z+RT)\
            -0.5*Omega**2*(Z+RT)**2*(np.sin(PHI*pi/180.))**2
    if domap:
        for k in range(maxkt%delta,maxkt+1,delta):
            plt.figure(10)
            #Ortografica sobre a localização do projétil (Terra roda)
            projection =ccrs.Orthographic(central_longitude=LAM[k],\
                        central_latitude=PHI[k],globe=None)
            map = plt.axes(projection=projection)
            map.set_global()
            map.stock_img()
            map.coastlines(resolution='50m')
            plt.scatter(0,90,marker='*',transform=data_crs,color='green')
            cs = map.plot(LAM[:k],PHI[:k],transform=data_crs)
            map.scatter(LAM[k],PHI[k],zorder=2,color='red',transform=data_crs)

            plt.title(r'$t=%5.0f s,\lambda=%3.1f,\phi=%4.1f,z=%7.3f km,\Delta
E/E=%4.3f $' %(k*dt,LAM[k],PHI[k],Z[k]/1000.,(EM[k]-EM[0])/EM[0]))
            plt.show()
            plt.savefig('Bal'+str(k)+'.png')
            plt.pause(0.01)
            plt.clf()
            print(k,k*dt,Z[k]/1000)
```

Cálculos finais

```
if doplots:
       plt.figure(4) #Energia mecânica
       plt.suptitle ('Energia (J/kg) aP=%3.1f ' % (aP))
       plt.subplot(3,1,kc+1);
       plt.plot(tempo,EM,color=cores[kc],\
                label=r'$\Delta t=%6.2f s $'%(dt));plt.legend()
       print(kc+1)

       fig=plt.figure(1) #trajetória 3D lam,phi,z
       ax.scatter(lam0,phi0,z0,color='red')
       ax.plot(xs=LAM,ys=PHI,zs=Z,color=cores[kc],\
               label=r'$\Delta t=%6.2f s $'%(dt))
       plt.xlabel(r'$\lambda ^{o}E $');plt.ylabel(r'$\varphi ^{o}N$')
       plt.legend()
       plt.title(r'$Trajetória $')
       Axes3D.view_init(ax,elev=30,azim=-145)
       Axes3D.view_init(ax,elev=30,azim=-145)
       tsubida=w0/g
       zmax=z0+w0*tsubida-0.5*g*tsubida**2
       print(zmax)
```

Cálculos finais

```
if doplots:
       plt.figure(4) #Energia mecânica
       plt.suptitle ('Energia (J/kg) aP=%3.1f ' % (aP))
       plt.subplot(3,1,kc+1);plt.plot(tempo,EM,color=cores[kc],label=r'$\Delta t=%6.2f s $'%(dt));plt.legend()
       print(kc+1)

       fig=plt.figure(1) #trajetória 3D lam,phi,z
       ax.scatter(lam0,phi0,z0,color='red')
       ax.plot(xs=LAM,ys=PHI,zs=Z,color=cores[kc],label=r'$\Delta t=%6.2f s $'%(dt))
       plt.xlabel(r'$\lambda ^{o}E $');plt.ylabel(r'$\varphi ^{o}N$')
       plt.legend()
       plt.title(r'$Trajetória $')
       Axes3D.view_init(ax,elev=30,azim=-145)
       Axes3D.view_init(ax,elev=30,azim=-145)
       tsubida=w0/g
       zmax=z0+w0*tsubida-0.5*g*tsubida**2
       print(zmax)

       plt.figure(2)
       plt.subplot(1,3,1)
       plt.plot(LAM,PHI,color=cores[kc]);plt.xlabel(r'$\lambda$');plt.ylabel(r'$\phi$')
       plt.scatter(lam0,phi0)
       plt.subplot(1,3,2)
       plt.plot(LAM,Z,color=cores[kc]);plt.xlabel(r'$\lambda$');plt.ylabel('z')
       plt.scatter(lam0,z0)
       plt.subplot(1,3,3)
       plt.plot(PHI,Z,color=cores[kc]);plt.xlabel(r'$\phi$');plt.ylabel('z')
       plt.scatter(phi0,z0)

       plt.figure(3)
       plt.subplot(1,3,1)
       plt.plot(tempo,U,color=cores[kc]);plt.xlabel('t');plt.ylabel('u')
       plt.subplot(1,3,2)
       plt.plot(tempo,V,color=cores[kc]);plt.xlabel('t');plt.ylabel('v')
       plt.subplot(1,3,3)
       plt.plot(tempo,W,color=cores[kc]);plt.xlabel('t');plt.ylabel('w')
       plt.suptitle(r"$\vec{v} (t)$")
```