Osnovi mehanike - vežbe 6
05. april 2022.

1. Primenom Ojler-Kromerove metode odrediti domet i maksimalnu visinu kosog hica koji je ispaljen brzinom $v_0 = 100\frac{\mathrm{m}}{\mathrm{s}}$ i pod uglom od $\alpha = 45^o$:

   a) ako na njega deluje samo homogeno polje Zemljine teže;

   b) ako na njega deluje sila otpora vazduha čiji je intenzitet dat kao:

   $$F_o = -\frac{1}{2}C_o S \rho v^2,$$

   gde je C_o koeficijent otpora (zavisi od mnogo faktora poput oblika tela, njegove veličine, brzine kretanja, Mahovog broja...), S je referentna površina tela, a $\rho = 1.23\frac{\mathrm{kg}}{\mathrm{m}^3}$ je gustina atmosfere. Uzeti da je $C_o S = 0.001$, a masa tela m=1kg.
   Prikazati putanje iz oba (a) i b)) primera.

```
import numpy as np
import matplotlib.pyplot as plt

g=9.81
v0=100
alfa=np.deg2rad(45)

dt=1e-3

# BEZ OTPORA
X=[] #naravno, moze i X=np.array([0])
Y=[]

x=0; y=0
vx=v0*np.cos(alfa); vy=v0*np.sin(alfa)

while y>=0:
    vy+=-g*dt
    x+=vx*dt
    y+=vy*dt
    X.append(x)
    Y.append(y)


# SA OTPOROM
C=0.001
ro=1.23
m=1

Xo=[]
Yo=[]

xo=0; yo=0
vxo=v0*np.cos(alfa); vyo=v0*np.sin(alfa)

while yo>=0:
    vo=np.sqrt(vxo**2+vyo**2)
    axo=-C*ro*vo*vxo/(2*m)
    ayo=-g-C*ro*vo*vyo/(2*m)

    vxo+=axo*dt
    vyo+=ayo*dt
```

```
        xo+=vxo*dt
        yo+=vyo*dt

        Xo.append(xo)
        Yo.append(yo)

X=np.array(X)
Y=np.array(Y)
Xo=np.array(Xo)
Yo=np.array(Yo)

#bez otpora
    #koordinate maksimalne visine
Hmax=Y[np.argwhere(Y==np.amax(Y))] #mo\v ze i samo Hmax=np.amax(Y)
HmaxX=X[np.argwhere(Y==np.amax(Y))]
    #koordinate dometa
D=X[-1]
DY=Y[-1]

#sa otporom
    #koordinate maksimalne visine
Homax=Yo[np.argwhere(Yo==np.amax(Yo))] #mo\v ze i samo Homax=np.amax(Yo)
HomaxX=Xo[np.argwhere(Yo==np.amax(Yo))]
    #koordinate dometa
Do=Xo[-1]
DoY=Yo[-1]

#GRAFIK
plt.figure()
plt.plot(X, Y, 'b')
plt.plot(Xo, Yo, 'r')

plt.plot(HmaxX, Hmax, 'ob')
plt.plot(D, DY, 'og')

plt.plot(HomaxX, Homax, 'or')
plt.plot(Do, DoY, 'oy')

plt.show()
```

c) Kako se menja razlika dometa u slučajevima bez i sa otporom za različite gustine sredina? Uzeti da su vrednosti $\rho \in (0.2, 0.3, ..., 2.4, 2.5)$ - **DOMAĆI**

d) Na koji način domet i maksimalna visina zavise od mase? Prikazati.

```
import numpy as np
import matplotlib.pyplot as plt

g=9.81
v0=100
alfa=np.deg2rad(45)

M=np.arange(100, 10000, 100)
Do=np.zeros(len(M))
Ho=np.zeros(len(M))

dt=1e-3
```

```python
# BEZ OTPORA - ne zavisi od mase ---> const
x=0; y=0
vx=v0*np.cos(alfa); vy=v0*np.sin(alfa)

Y=[]

while y>=0:
    vy+=-g*dt
    x+=vx*dt
    y+=vy*dt
    Y=np.append(Y, y)

D=np.ones_like(M)*x
h=np.amax(Y)
H=np.ones_like(M)*h

# SA OTPOROM - zavisi od mase
C=0.001
ro=1.23

Yo=[[] for column in range(len(M))]

for i, m in enumerate(M):
    print(m)

    xo=0; yo=0
    vxo=v0*np.cos(alfa); vyo=v0*np.sin(alfa)

    while yo>=0:
        vo=np.sqrt(vxo**2+vyo**2)
        axo=-C*ro*vo*vxo/(2*m)
        ayo=-g-C*ro*vo*vyo/(2*m)

        vxo+=axo*dt
        vyo+=ayo*dt

        xo+=vxo*dt
        yo+=vyo*dt
        Yo[i].append(yo)
    Do[i]=xo

for i in range(len(Yo)):
    ho = np.amax(Yo[i])
    Ho[i]=ho

#domet
plt.figure()
plt.title('Zavisnost dometa od mase')
plt.plot(M, D, '--r')
plt.plot(M, Do)
plt.show()

#visine
plt.figure()
plt.title('Zavisnost visine od mase')
plt.plot(M, H, '--r')
plt.plot(M, Ho)
plt.show()
```

e) Na koji način domet i visina zavise od intenziteta početne brzine, ugla nagiba vektora početne brzine i mase? Prikazati.

```
import numpy as np
from matplotlib import pyplot as plt

g=9.81 # ubrzanje homogenog polja
ro=1.2255 # gustina vazduha
C=1e-1 # koeficijent otpora, izmenjen da se bolje vidi rezultat

dt=1e-3 # vremenski korak

M=np.linspace(1, 100, 10) # masa tela
v=np.arange(10,100,10) # pocetna brzina
alfa=np.deg2rad(np.arange(10,80,10)) # pocetni ugao

domet=np.zeros([len(v), len(alfa), len(M)]) # domet

for i in range(len(v)):
    for j in range(len(alfa)):
        for k in range(len(M)):
            print(i, j, k)

            x=0; y=0 # pocetni polozaj
            vx=v[i]*np.cos(alfa[j]); vy=v[i]*np.sin(alfa[j]) # pocetna brzina

            while y>=0:
                vo = np.sqrt(vx**2 + vy**2)

                vx += -vx * vo * C / M[k] * ro / 2 * dt
                vy += -(g + vy * vo * C / M[k] * ro / 2) * dt
                x += vx * dt
                y += vy * dt

            domet[i][j][k] = x

# 3D plot
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
fig = plt.figure()
ax = fig.add_subplot(1,1,1, projection='3d')
ALFA, V = np.meshgrid(alfa, v)

# plotujemo len(M) povrsi u 3D koordinatnom sistemu
# cije su ose: alfa, v i domet,
# a svakoj povrsi odgovara jedna vrednost mase

for i in range(len(M)):
    ax.plot_surface(np.rad2deg(ALFA),V,domet[:,:,i])
ax.set_xlabel('alfa')
ax.set_ylabel('brzina [m/s]')
ax.set_zlabel('domet')
plt.show()
```

2. Primenom Ojler-Kromerove metode odrediti kojom brzinom treba ispaliti telo vertikalno u vis da bi stiglo do Meseca na udaljenosti $H = 3,84 \times 10^8$m:

   a) ako na njega deluje samo homogeno polje Zemljine teže ($g = 9,81 \frac{\mathrm{m}}{\mathrm{s}^2}$);

   ```
   import numpy as np
   from matplotlib import pyplot as plt
   ```

```python
V=0 #pocetna brzina
h=0 # dostignuta visina
korak=100 #korak povecavanja brzine

g=9.81
H=3.84e8 # rastojanje do Meseca

dt=1

while h<H:
    h=0
    V+=korak
    v=V
    while v>0:
        v+=-g*dt
        h+=v*dt

print(V/1000, 'km/s')
```

b) ako na njega deluje samo gravitaciona sila Zemlje obrnuto srazmerna kvadratu rastojanja ($\gamma = 6,67 \times 10^{-11} \frac{\mathrm{m}^3}{\mathrm{kgs}^2}$, $M_z = 5,972 \times 10^{24}$kg, $R_z = 6,378 \times 10^6$m);

```python
import numpy as np
from matplotlib import pyplot as plt

V=0 #pocetna brzina
r=0 # dostignuta visina
korak=10 #korak povecavanja brzine

Rm=3.84e8 # rastojanje do Meseca
gama=6.67e-11 # gravitaciona konstanta

M=5.972e24 #masa Zemlje
R=6378e3 # poluprecnik Zemlje

dt=1

while r<Rm:
    r=R
    V+=korak
    v=V
    while v>0:
        v+=-gama*M/r**2*dt
        r+=v*dt

print(V/1000, 'km/s')
```

c) ako na njega deluju gravitacione sile Zemlje i Meseca obrnuto srazmerne kvadratu rastojanja (pod pretpostavkom da su Zemlja i Mesec stacionarni objekti) ($M_m = 7.348 \times 10^{22}$kg).

```python
import numpy as np
from matplotlib import pyplot as plt

V=0 #pocetna brzina
r=0 # dostignuta visina
korak=10 #korak povecavanja brzine

rm=3.84e8 # rastojanje do Meseca
```

```python
gama=6.67e-11 # gravitaciona konstanta

Mz=5.972e24 # masa Zemlje
Mm=7.348e22 # masa Meseca
R=6378e3 # poluprecnik Zemlje

dt=1e-1

while r<rm:
    r=R
    V+=korak
    v=V
    while v>0 and r<rm:
        v+=(Mm/(rm-r)**2-Mz/r**2)*gama*dt
        r+=v*dt
    #print(r/1000)

print(V/1000, 'km/s')
```