

## Osnovi mehanike - vežbe 7

12. april 2022.

1. Padobranac slobodno pada sa visine  $H = 2\text{km}$ . Pored sile Zemljine teže, na njega deluje i sila otpora vazduha data kao:  $F_o = -mkv$  (važi za male brzine), gde je  $m$  masa padobranca, a  $k = 0.17658$  koeficijent (zavisi od osobina tela i sredine, i geometrije tela). Pomoću Ojler-Kromerove metode odrediti:

- a) na koji način brzina zavisi od vremena i prikazati;

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

g=9.81
k=0.17658 #1/s

V0=0 #pocetna brzina u m/s
t=0 #pocetni trenutak u s
H=2000 #visina u m

dt=1e-3 #s

V=[]
T=[]

v=V0
h=H
while h>0:
    v+=(g-k*v)*dt
    h-=v*dt
    t+=dt
    V.append(v)
    T.append(t)

plt.figure()
plt.plot(T, np.ones_like(T)*g/k, '--r', label='asimptotska brzina')
plt.plot(T, V, '.b', label='trenutna brzina')
plt.legend()
plt.xlabel('vreme padanja')
plt.ylabel('brzina')
plt.show()
```

- b) nakon koliko vremena i na kojoj visini je razlika između asimptotske brzine i one dobijene Ojler-Kromerovom metodom manja od 1%.

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

g=9.81
k=0.17658 #1/s

V0=0 #pocetna brzina u m/s
t=0 #pocetni trenutak u s
H=2000 #visina u m

dt=1e-3 #s

vas=g/k
v=V0
h=H
while h>0:
```

```

v+=(g-k*v)*dt
h-=v*dt
t+=dt
if abs(vas-v)/vas<0.01:
    print(t, h)
    break

```

2. Primenom Ojler-Kromerove metode odrediti kojom brzinom telo mase  $m = 250\text{kg}$  koje slobodno pada sa visine  $h[\text{m}] \in (100, 110, \dots, 9990, 10000)$  udara u Zemlju ako na njega deluje sila otpora atmosfere i homogeno polje Zemljine gravitacije:

- uzimajući u obzir samo homogeno polje Zemljine gravitacije;
- uzimajući u obzir i otpor vazduha koji se može izraziti kao:

$$F_o = -\frac{1}{2}C_oS\rho v^2$$

( $C_o$  je koeficijent otpora,  $S$  je referentna površina tela,  $\rho$  je gustina atmosfere) uz  $C_oS = 0.5$ ;  
Ubrzanje usled aerodinamičkog otpora je:

$$a_o = -\frac{1}{2} \frac{\rho v^2}{BC},$$

gde je:

$$BC = \frac{m}{C_oS}$$

balistički koeficijent. Uzeti i da gustina atmosfere opada sa visinom prema eksponencijalnom zakonu:

$$\rho = \rho_0 e^{-h/H},$$

gde je  $H = 10\text{km}$  skala visina, a  $\rho_0 = 1,23 \frac{\text{kg}}{\text{m}^3}$  gustina vazduha na površini Zemlje;

- prikazati zavisnost brzine udara o tlo od visine za oba slučaja (pod a) i b))
- odrediti rad sile otpora vazduha.

```

import numpy as np
from matplotlib import pyplot as plt

BC=500. # Balisticki koeficijent
m=250. # masa tela
ro0=1.2255 # gustina atmosfere na površini Zemlje
skala=10000. # skala visina
g=9.81

H=np.arange(100.,10000.,10.) # pocetna visina
dt=1e-2

v1=np.zeros_like(H) # samo gravitacija
v2=np.zeros_like(H) # gravitacija i otpor
w1=np.zeros_like(H) # izvrsen rad (sa otporom)
w2=np.zeros_like(H) # razlika u kinetickim energijama (sa/bez otpora)

for i, h1 in enumerate(H):
    #bez otpora
    h=h1
    v=0
    while h>0:
        v+=g*dt
        h+=-v*dt

```

```

v1[i]=v

#sa otporom
h=h1
v=0
hh=[]
w=0
while h>0:
    ro = ro0 * np.exp(-h / skala)
    ao = 0.5*ro*v**2/BC
    v+=(g-ao)*dt
    h+=-v*dt
    hh.append(h)
    #print(i, h)
    if len(hh)>1:
        w=-m*ao*(hh[-2]-hh[-1]) # rad po definiciji

v2[i]=v
w1[i]=w
w2[i]=0.5*m*(v2[i]**2-v1[i]**2)

plt.figure()
linija1=plt.plot(H,v1,'k', label='bez otpora')
linija2=plt.plot(H,v2,'r', label='sa otporom')
plt.legend()
plt.xlabel('pocetna visina [m]')
plt.ylabel('brzina udara [m/s]')
plt.show()

```

3. Primenom Ojler-Kromerove metode odrediti putanju kosmičke letelice koja je sa Zemlje lansirana brzinom  $v_0 = 10 \frac{\text{km}}{\text{s}}$  (u odnosu na Zemlju) u smeru Zemljine heliocentrične brzine. Pretpostaviti da se Zemlja kreće po kružnoj putanji oko Sunca.

```

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

Mz = 5.972e24 # masa Zemlje [kg]
Ms = 1.989e30 # masa Sunca [kg]
gama = 6.67e-11 # [m^3/(kg*s^2)]
aj = 1.5e11 # astronomska jedinica [m]

r0z = 6.378e6 # poluprecnik Zemlje [m]
r0s = 1.5e11 # rastojanje od Sunca do Zemlje [m]
v_zemlja = 2 * np.pi * r0s / (365.25 * 86400) #[m/s]

v0 = 1e4 # pocetna brzina u odnosu na Zemlju [m/s]

# pocetni uslovi (heliocentricni sistem)
x = [r0s + r0z]
y = [0]
z = [0]
vx = 0
vy = v_zemlja + v0
vz = 0

dt = 100000. # [s]
t = 0.

# np.arctan(f) i np.arctan2(f1, f2)

```

```

# phi in (-np.pi, np.pi) !!! (za razliku od np.arctan(f) in (-np.pi/2, np.pi/2))
phi = list(np.arctan2(y, x))

while len(phi) == 1 or (len(phi) > 1 and phi[-1] > phi[-2]):
    t += dt

    ax = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2) ** (3 / 2) * x[-1]
    ay = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2) ** (3 / 2) * y[-1]
    vx += ax * dt
    vy += ay * dt
    x.append(x[-1] + vx * dt)
    y.append(y[-1] + vy * dt)

    #np.mod jer zelimo iz (-pi, pi) u (0, 2*pi)
    # (-pi, 0) --> (pi, 2pi)
    # (0, pi) --> (0, pi)
    phi.append(np.mod(np.arctan2(y[-1], x[-1]), 2*np.pi))

x1 = np.array(x)
y1 = np.array(y)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set_xlim3d(-10, 10)
ax.set_ylim3d(-10, 10)
plt.plot(0, 0, 0, 'oy') # polozaj Sunca
plt.plot(1, 0, 0, 'og') # polozaji Zemlje za neke trenutke
plt.plot(-1, 0, 0, 'og')
plt.plot(0, 1, 0, 'og')
plt.plot(0, -1, 0, 'og')
plt.plot(x1 / aj, y1 / aj, np.zeros(len(x1))) # putanja letelice
plt.show()

```