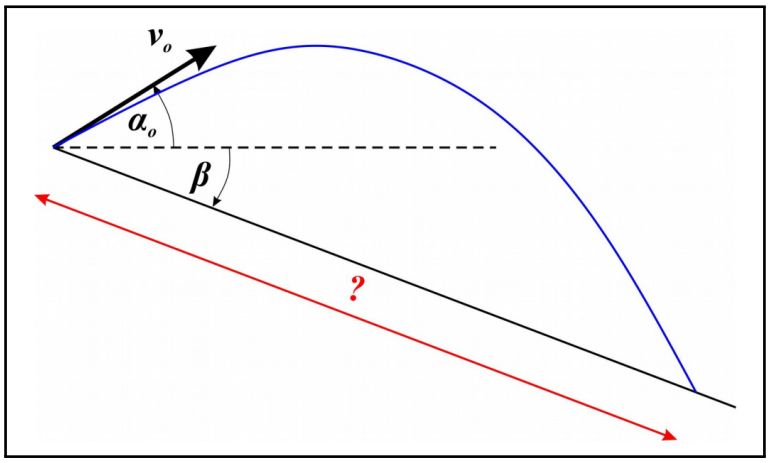
Osnovi mehanike - vežbe 3
15. mart 2022.

1. Primenom Ojler-Kromerove metode:



a) odrediti domet hica koji se nad padinom pod uglom β (u odnosu na horizontalnu ravan) lansira početnom brzinom v_0 i pod uglom α (u odnosu na horizontalnu ravan) i prikazati putanju i padinu;

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

g=-9.81

beta=np.deg2rad(-36) #nagib padine
dt=1e-3 #vremenski korak

# pocetni uslovi
v=386
alfa=np.deg2rad(49)

x=[0]; y=[0]; vx=v*np.cos(alfa); vy=v*np.sin(alfa)


while x[-1]==0 or y[-1]/x[-1]>np.tan(beta):

    vy+=g*dt
    x.append(x[-1]+vx*dt)
    y.append(y[-1]+vy*dt)

dolet=x[-1]/np.cos(beta)

x=np.array(x)
y=np.array(y)
padina=x*np.tan(beta)

plt.figure()
plt.plot(x,y,'r', linewidth=3) #putanja
plt.plot(x,padina,'k') #padina
plt.show()
```

b) odrediti na koji način domet i vreme leta zavisi od uglova α i β i prikazati datu zavisnost;

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

```
# konstante

g=-9.81

beta=np.deg2rad(np.arange(-45,0)) #nagib padine
alfa=np.deg2rad(np.arange(0,80)) #nagib padine
dolet=np.zeros([len(beta), len(alfa)])
v=250.
dt=1 #vremenski korak

# pocetni uslovi
for i in range(len(beta)):
    for j in range(len(alfa)):


        x=0; y=0;
        vx=v*np.cos(alfa[j]); vy=v*np.sin(alfa[j])


        while x==0 or y/x>np.tan(beta[i]):

            vy+=g*dt
            x+=vx*dt
            y+=vy*dt

        dolet[i][j]=x/np.cos(beta[i])

# plotovanje 3D

from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

# konture
plt.figure()
plt.contourf(np.rad2deg(alfa), np.rad2deg(beta), dolet,20, cmap=cm.jet)
plt.xlabel('alfa')
plt.ylabel('beta')
plt.colorbar()
#plt.show()

# 3D plot
fig = plt.figure()
ax = fig.add_subplot(1,1,1, projection='3d')

alfa, beta = np.meshgrid(alfa, beta)

ax.plot_surface(np.rad2deg(alfa),np.rad2deg(beta),dolet,cmap=cm.jet,
                linewidth=0, antialiased=False)
ax.set_xlabel('alfa')
ax.set_ylabel('beta')
ax.set_zlabel('dolet [m]')
plt.show()
```

c) odrediti pod kojim uglom će hitac udariti u Zemlju i kako ovaj ugao zavisi od uglova α i β i prikazati datu zavisnost.

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

# konstante

g=9.81
```

```
beta=np.deg2rad(np.arange(-45,0,2)) #nagib padine
alfa=np.deg2rad(np.arange(0,80,2)) #nagib padine
ugao=np.zeros([len(beta), len(alfa)])
v=250.
dt=1 #vremenski korak

for i in range(len(beta)):
    for j in range(len(alfa)):

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

        while x==0 or y/x>np.tan(beta[i]):

            vy-=g*dt
            x+=vx*dt
            y+=vy*dt

        ugao[i][j]=np.abs(np.arctan(vy/vx)-beta[i])

# plotovanje 3D

from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

# konture
plt.figure()
plt.contourf(np.rad2deg(alfa), np.rad2deg(beta), np.rad2deg(ugao),20, cmap=cm.jet)
plt.xlabel('alfa')
plt.ylabel('beta')
plt.colorbar()

# 3D plot
fig = plt.figure()
ax = fig.add_subplot(1,1,1, projection='3d')

alfa, beta = np.meshgrid(alfa, beta)

ax.plot_surface(np.rad2deg(alfa),np.rad2deg(beta),np.rad2deg(ugao),cmap=cm.jet,
                linewidth=0, antialiased=False)
ax.set_xlabel('alfa')
ax.set_ylabel('beta')
ax.set_zlabel('ugao')
plt.show()
```