Osnovi mehanike - vežbe 8
19. april 2022.

1. Primenom Ojler-Kromerove metode odrediti putanju kosmičke letelice koja je sa Zemlje lansirana brzinom 10 km/s (u odnosu na Zemlju) normalno na Zemljinu heliocentričnu brzinu. Pretpostaviti da se Zemlja kreće po kružnoj putanji oko Sunca.

   a) Uzeti u obzir samo gravitaciju 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 (helicentricni sistem)
x = [r0s + r0z]
y = [0]
z = [0]
vx = 0
vy = v_zemlja
vz = v0

dt = 1000.
t = 0.
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 + z[-1] ** 2) ** (3 / 2) * x[-1]
    ay = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * y[-1]
    az = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * z[-1]

    vx += ax * dt
    vy += ay * dt
    vz += az * dt

    x.append(x[-1] + vx * dt)
    y.append(y[-1] + vy * dt)
    z.append(z[-1] + vz * dt)

    phi.append(np.mod(np.arctan2(y[-1], x[-1]), 2 * np.pi))

x2 = np.array(x)
y2 = np.array(y)
z2 = np.array(z)

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set_xlim3d(-1.5, 1.5)
ax.set_ylim3d(-1.5, 1.5)
```

```
ax.set_zlim3d(-1.5, 1.5)
plt.plot(0, 0, 0, 'oy', label='Sunce') # polozaj Sunca
plt.plot(1, 0, 0, 'og', label='Zemlja') # 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(x2 / aj, y2 / aj, z2 / aj, label='letelica') # putanja letelice
plt.legend()
plt.show()
```

b) Uzeti u obzir gravitaciju Zemlje i Sunca i posmatrati kretanje u toku jedne julijanske godine.

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

Ms = 1.989e30
Mz = 5.972e24
gama = 6.67e-11
aj = 1.5e11  # astronomska jedinica
r0z = 6.378e6  # poluprecnik Zemlje
r0s = 1.5e11  # rastojanje od Sunca do Zemlje

v_zemlja = 2 * np.pi * r0s / (365.25 * 86400)

v0 = 1e4  # pocetna brzina u odnosu na Zemlju

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

dt = 1000.
t = 0.
phi = list(np.arctan2(y, x))

sk = 2 * np.pi / 365.25 / 86400  # ugaona brzina kretanja Zemlje oko Sunca

# koordinate Zemlje
xz = [r0s]
yz = [0]
zz = [0]

xgc = [x[0] - xz[0]]
ygc = [y[0] - yz[0]]
zgc = [z[0] - zz[0]]

while t < 365.25 * 86400:

    t += dt

    ax = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * x[-1] \
         - gama * Mz / (xgc[-1] ** 2 + ygc[-1] ** 2 + zgc[-1] ** 2) ** (3 / 2) * xgc[-1]

    ay = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * y[-1] \
         - gama * Mz / (xgc[-1] ** 2 + ygc[-1] ** 2 + zgc[-1] ** 2) ** (3 / 2) * ygc[-1]

    az = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * z[-1] \
```

```
            - gama * Mz / (xgc[-1] ** 2 + ygc[-1] ** 2 + zgc[-1] ** 2) ** (3 / 2) * zgc[-1]

        vx += ax * dt
        vy += ay * dt
        vz += az * dt

        x.append(x[-1] + vx * dt)
        y.append(y[-1] + vy * dt)
        z.append(z[-1] + vz * dt)

        lz = t * sk  # longituda Zemlje

        # pravougle koordinate Zemlje
        xz.append(r0s * np.cos(lz))
        yz.append(r0s * np.sin(lz))
        zz.append(0)

        # geocentricne koordinate objekta
        xgc.append(x[-1] - xz[-1])
        ygc.append(y[-1] - yz[-1])
        zgc.append(z[-1] - zz[-1])

    x = np.array(x)
    y = np.array(y)
    z = np.array(z)

    xz = np.array(xz)
    yz = np.array(yz)
    zz = np.array(zz)

    xgc = np.array(xgc)
    ygc = np.array(ygc)
    zgc = np.array(zgc)

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    ax.set_xlim3d(-1.5, 1.5)
    ax.set_ylim3d(-1.5, 1.5)
    ax.set_zlim3d(-1.5, 1.5)

    plt.plot(xz/r0s, yz/r0s, zz/r0s, 'g', label='Zemlja')
    plt.plot(x/r0s, y/r0s, z/r0s, label='letelica')
    plt.plot(0,0,0, 'oy', label='Sunce')

    plt.legend()
    plt.show()
```

2. Primenom Ojler-Kromerove metode odrediti geocentričnu i heliocentričnu putanju kosmičke letelice koja je sa udaljenosti 50 poluprečnika Zemlje (od Zemlje) preusmerila svoje kretanje brzinom $\vec{v} = v_0\vec{e_y} + 0,5v_0\vec{e_z}$, gde je $v_0 = 1$ km/s. Uzeti u obzir i gravitaciju Zemlje i gravitaciju Sunca.

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

Ms = 1.989e30
Mz=5.972e24
gama = 6.67e-11
aj = 1.5e11  # astronomska jedinica
r0z = 6.378e6  # poluprecnik Zemlje
```

```
r0s = 1.5e11  # rastojanje od Sunca do Zemlje

v_zemlja = 2 * np.pi * r0s / (365.25 * 86400)

v0 = 1e3

# pocetni uslovi (helicentricni sistem)
x = [r0s + 50 * r0z]
y = [0]
z = [0]

vx = 0
vy = v_zemlja+v0
vz = v0/2

dt = 1000.
t = 0.

sk = 2 * np.pi / 365.25 / 86400  # ugaona brzina kretanja Zemlje oko Sunca

# koordinate Zemlje
xz = [r0s]
yz = [0]
zz = [0]

# koordinate letelice u odnosu na Zemlju
xgc = [x[0] - xz[0]]
ygc = [y[0] - yz[0]]
zgc = [z[0] - zz[0]]

while t < 365.25 * 86400:

    t += dt

    ax = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * x[-1] \
         - gama * Mz / (xgc[-1] ** 2 + ygc[-1] ** 2 + zgc[-1] ** 2) ** (3 / 2) * xgc[-1]

    ay = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * y[-1] \
         - gama * Mz / (xgc[-1] ** 2 + ygc[-1] ** 2 + zgc[-1] ** 2) ** (3 / 2) * ygc[-1]

    az = -gama * Ms / (x[-1] ** 2 + y[-1] ** 2 + z[-1] ** 2) ** (3 / 2) * z[-1] \
         - gama * Mz / (xgc[-1] ** 2 + ygc[-1] ** 2 + zgc[-1] ** 2) ** (3 / 2) * zgc[-1]

    vx += ax * dt
    vy += ay * dt
    vz += az * dt

    x.append(x[-1] + vx * dt)
    y.append(y[-1] + vy * dt)
    z.append(z[-1] + vz * dt)

    lz = t * sk  # lz - longituda Zemlje

    # pravougle koordinate Zemlje
    xz.append(r0s * np.cos(lz))
    yz.append(r0s * np.sin(lz))
    zz.append(0)

    # geocentricne koordinate objekta
    xgc.append(x[-1] - xz[-1])
    ygc.append(y[-1] - yz[-1])
```

```python
    zgc.append(z[-1] - zz[-1])

x = np.array(x)
y = np.array(y)
z = np.array(z)

xz = np.array(xz)
yz = np.array(yz)
zz = np.array(zz)

xgc = np.array(xgc)
ygc = np.array(ygc)
zgc = np.array(zgc)

########## putanja letelice u odnosu na Zemlju
fig = plt.figure()
fig.add_subplot(111, projection='3d')

plt.plot(xgc/r0z,ygc/r0z,zgc/r0z, label='letelica')
plt.plot(0, 0, 0, 'og', label='Zemlja')
#plt.plot(r0s, 0, 0, 'oy', label='Sunce')
plt.legend()
# plt.savefig('geo_putanja')
plt.show()

########## putanja letelice u odnosu na Sunce
fig = plt.figure()
fig.add_subplot(111, projection='3d')

plt.plot(x/aj, y/aj, z/aj, label='letelica') # letelica
plt.plot(0,0,0, 'oy', label='Sunce')
plt.plot(xz/aj, yz/aj, zz/aj, 'g',label='Zemlja')
plt.legend()
# plt.savefig('helio_putanja')
plt.show()
```