SMA kurso
4 inžinerinio projekto pavyzdys

Paprastosios diferencialinės lygties sprendimas

3000m aukštyje skrendančio lėktuvo pilotas katapultuojasi. Jis drauge su krėslu išmetamas aukštyn katapultos raketinio variklio pagalba, kuris pirmųjų 3 sekundžių laike sukuria vertikaliai į viršų nukreiptą F=3000N jėgą. Piloto su apranga masė mp=80kg, krėslo masė mk=150kg. Šiame uždavinyje nagrinėkime tik vertikalų judesį (aukštyn-žemyn), nors iš tikrųjų pilotas ir krėslas turi dar ir horizontalią judėjimo komponentę.

Išsijungus katapultos varikliui, pilotas su krėslu juda ore, veikiami tik oro pasipriešinimo, kurio jėga porporcinga greičio kvadratui. Oro pasipriešinimo krėslo judėjimui koeficientas kk=3kg/m. Pilotas, būdamas krėsle, oro pasipriešinimo jėgos neįtakoja. Kai pasiekiamas 1000m aukštis, krėslas atsiskiria ir toliau krenta veikiamas oro pasipriešinimo, o piloto parašiutas išsiskleidžia, sukurdamas oro pasipriešinimą su koeficientu kp=25kg/m.

Apskaičiuoti abiejų kūnų aukščio kitimo priklausomybes nuo laiko. Kada ir kokiu greičiu kiekvienas jų pasieks žemę?

h=3000

$$\frac{dh}{dt} = v$$

$$(m_p + m_k)\frac{dv}{dt} = -(m_p + m_k)g - k_k v^2 sign(v) + F$$

$$if \;\; h_k \geq 1000 \;\; :$$

$$\frac{d}{dt}\begin{Bmatrix} h_k \\ h_p \\ v_k \\ v_p \end{Bmatrix} = \begin{Bmatrix} v_k \\ v_k \\ -g - \dfrac{k_k v_k^2 sign(v_k) + F}{m_k + m_p} \\ -g - \dfrac{k_k v_k^2 sign(v_k) + F}{m_k + m_p} \end{Bmatrix}, \qquad \left.\begin{Bmatrix} h_k \\ h_p \\ v_k \\ v_p \end{Bmatrix}\right|_{t=0} = \begin{Bmatrix} 3000 \\ 3000 \\ 0 \\ 0 \end{Bmatrix}$$

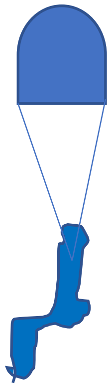
$$if \;\; h_k < 1000 \;\; :$$

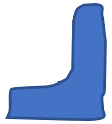
$$\frac{d}{dt}\begin{Bmatrix} h_k \\ h_p \\ v_k \\ v_p \end{Bmatrix} = \begin{Bmatrix} v_k \\ v_p \\ -g - \dfrac{k_k v_k^2 sign(v_k)}{m_k} \\ -g - \dfrac{k_p v_p^2 sign(v_p)}{m_p} \end{Bmatrix}$$

$$if \; h_k <= 0: \;\; h_k = 0; \; v_k = 0$$

$$if \; h_p <= 0: \;\; h_p = 0; \; v_p = 0$$

h=1000





$$\begin{cases} \dfrac{dh}{dt} = v \\ m_p \dfrac{dv}{dt} = -m_p g - k_p v^2 sign(v) \end{cases}$$

$$\begin{cases} \dfrac{dh}{dt} = v \\ m_g \dfrac{dv}{dt} = -m_k g - k_k v^2 sign(v) \end{cases}$$

```python
def funk(X,t):  # PDL desines puses funkcija

  hk=X[0];hp=X[1];vk=X[2];vp=X[3]; # kreslo ir piloto auksciai ir greiciai
  mp=80;mk=150;kk=3;kp=25          # mases ir oro pasipriesinimo koeficientai
  F=8000;    # katapultos jega
  if t > 3: F=0

  rez=np.zeros(4,dtype=float)

  if hk > 1000:   # iki parasiuto issiskleidimo
    rez[0]=vk;rez[1]=vp;
    rez[2]=-9.81+(F-kk*vk**2*np.sign(vk))/(mp+mk);
    rez[3]=rez[2];
  else:  # issikleidus parasiutui
         # kai pasiekta zeme, greitis ir aukstis lygus nuliui,
         # Taciau nulines reiksmes greiciams ir auksciams reikia
         # suteikti, atliekant skaitinio integravimo zingsni,
         # o ne sioje PDL desiniosios puses funkcijoje
    rez[0]=vk;rez[2]=-9.81-kk*vk**2*np.sign(vk)/mk
    rez[1]=vp; rez[3]=-9.81-kp*vp**2*np.sign(vp)/mp;

  return rez
```

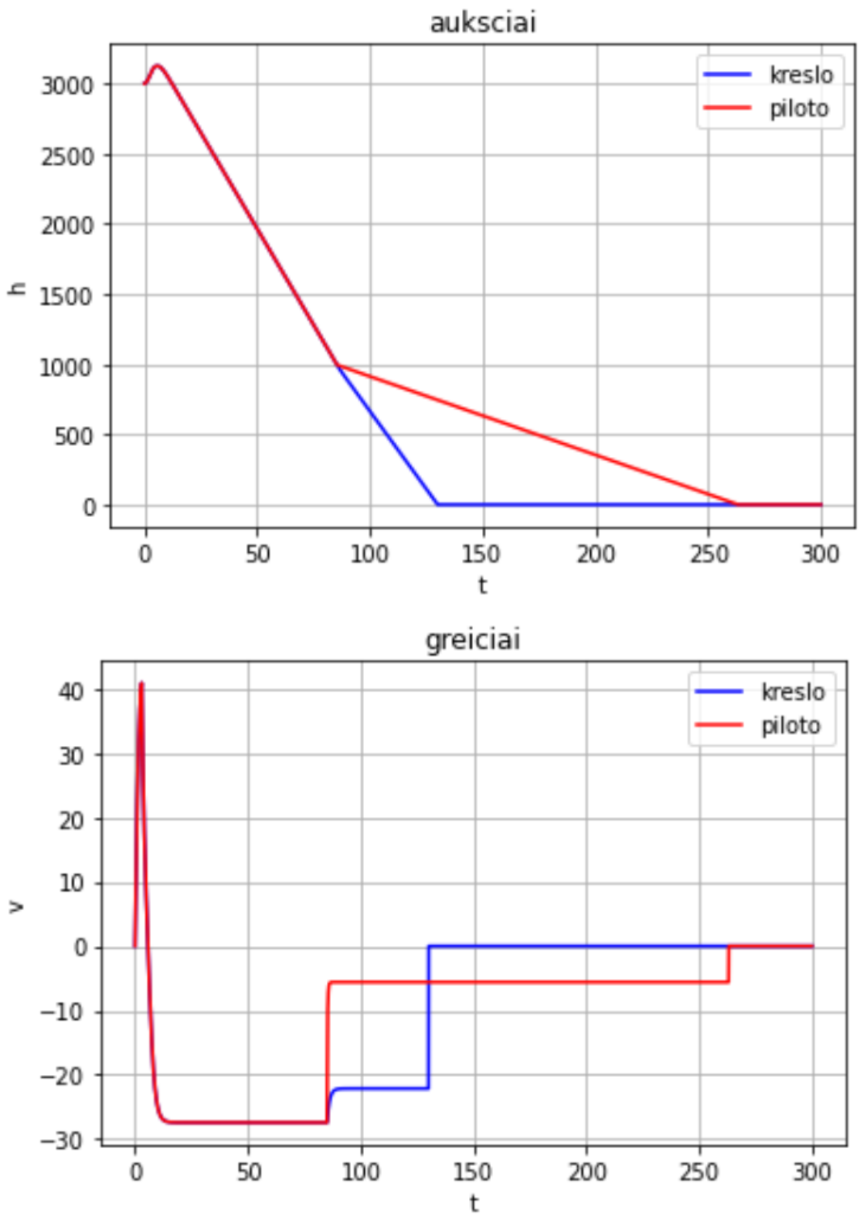
$$if\ \ h_k \geq 1000\ \ :$$

$$\frac{d}{dt}\begin{Bmatrix} h_k \\ h_p \\ v_k \\ v_p \end{Bmatrix} = \begin{Bmatrix} v_k \\ v_k \\ -g - \dfrac{k_k v_k^2 sign(v_k) + F}{m_k + m_p} \\ -g - \dfrac{k_k v_k^2 sign(v_k) + F}{m_k + m_p} \end{Bmatrix}$$

$$if\ \ h_k < 1000\ \ :$$

$$\frac{d}{dt}\begin{Bmatrix} h_k \\ h_p \\ v_k \\ v_p \end{Bmatrix} = \begin{Bmatrix} v_k \\ v_p \\ -g - \dfrac{k_k v_k^2 sign(v_k)}{m_k} \\ -g - \dfrac{k_p v_p^2 sign(v_p)}{m_p} \end{Bmatrix}$$

```python
def funk(X,t):  # PDL desines puses funkcija
  hk=X[0];hp=X[1];vk=X[2];vp=X[3]; # kreslo ir piloto auksciai ir greiciai
  mp=80;mk=150;kk=3;kp=25          # mases ir oro pasipriesinimo koeficientai
  F=8000;    # katapultos jega
  if t > 3: F=0

  rez=np.zeros(4,dtype=float)

  if hk > 1000:   # iki parasiuto issiskleidimo
    rez[0]=vk;rez[1]=vp;
    rez[2]=-9.81+(F-kk*vk**2*np.sign(vk))/(mp+mk);
    rez[3]=rez[2];
  else:           # issikleidus parasiutui
        # kai pasiekta zeme, greitis ir aukstis lygus nuliui,
        # Taciau nulines reiksmes greiciams ir auksciams reikia
        # suteikti, atliekant skaitinio integravimo zingsni,
        # o ne sioje PDL desiniosios puses funkcijoje
    rez[0]=vk;rez[2]=-9.81-kk*vk**2*np.sign(vk)/mk
    rez[1]=vp; rez[3]=-9.81-kp*vp**2*np.sign(vp)/mp;
  return rez

ttt=300  # skaiciavimo laikas  (s)
t=np.linspace(0,ttt,10000);dt=t[1]-t[0]; #laiko momentai
h0=3000  # pradinis aukstis (m)
N=len(t); rez=np.zeros([4,N],dtype=float) # rezultatu masyvas
rez[:,0]=np.array([h0,h0,0,0]);           # pradines salygos

if 0:   # sprendimas Eulerio metodu
  for i in range (N-1):
   rez[:,i+1]=rez[:,i]+funk(rez[:,i],t[i])*dt

else:
  for i in range (N-1) :  # sprendimas IV RK metodu
    fz=rez[:,i]+funk(rez[:,i],t[i])*dt/2
    fzz=rez[:,i]+funk(fz,t[i]+dt/2)*dt/2
    fzzz=rez[:,i]+funk(fzz,t[i]+dt/2)*dt
    rez[:,i+1]=rez[:,i]+dt/6*(funk(rez[:,i],t[i])+2*funk(fz,t[i]+dt/2)+2*funk(fzz,t[i]+dt/2)+funk(fzzz,t[i]+dt))
    # jeigu pasiekta zeme, greiciams ir auksciams priverstinai priskiriami nuliai. Funkcijoje fnk butu galima priskirti nulius tik greiciu isvestinems.
    # Tai reikstu kad jie nebekinta, taciau integruojant PDL reiksmes isliktu tokios pacios, kaip ir smugio i zeme metu:
    if rez[0,i+1] <= 0 : rez[[0,2],i+1]=0
    if rez[1,i+1] <= 0 : rez[[1,3],i+1]=0


# rezultatu pavaizdavimas:
fig1=plt.figure(1); ax1=fig1.add_subplot(1,1,1); ax1.set_xlabel('t');ax1.set_ylabel('h');ax1.grid();plt.title('auksciai')
ax1.plot(t,rez[0,:],'b-');ax1.plot(t,rez[1,:],'r-'); plt.legend(['kreslo','piloto']);plt.show()

fig2=plt.figure(1); ax2=fig2.add_subplot(1,1,1); ax2.set_xlabel('t');ax2.set_ylabel('v');ax2.grid();plt.title('greiciai')
ax2.plot(t,rez[2,:],'b-');ax2.plot(t,rez[3,:],'r-');plt.legend(['kreslo','piloto']);plt.show();
```


auksciai
kreslo
piloto
h
t
greiciai
kreslo
piloto
v
t